

RAPPORT PROJET FIL ROUGE

Réalisé par : Adil El hakouni-Marc Perrin-
Youssef Lamzaouak

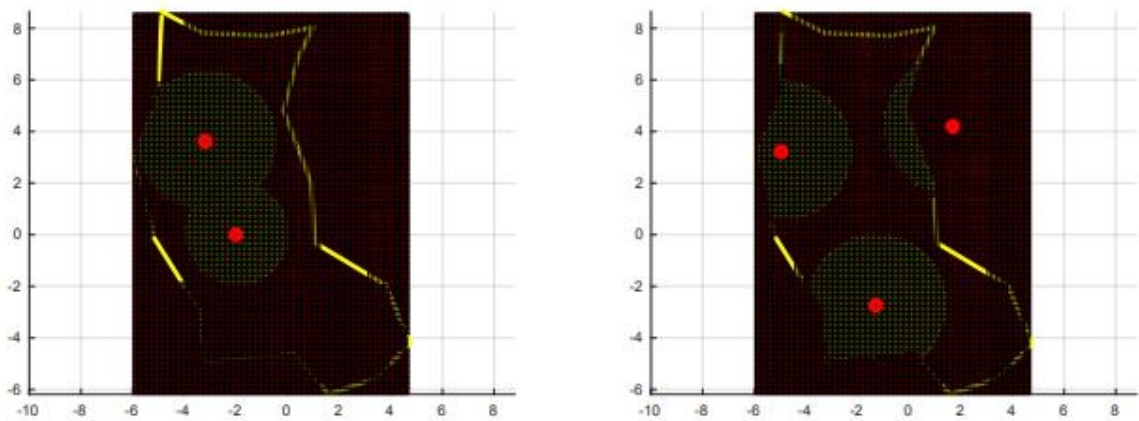
2019-2020

SOMMAIRE :

1. Présentation générale du problème	
2. SVM/SVR.....	
3. Metamodelling et krigeage.....	
4. Planification d'expériences.....	
5. Analyse de sensibilité.....	
6. Optimisation globale	
• En utilisant l'algorithme CMA-ES.....	
• En utilisant l'algorithme EGO.....	
7. Conclusion	

I. Présentation générale du problème :

Nous voulons étudier l'influence de la localisation des antennes pour la couverture d'un territoire. Chaque antenne se caractérise par son emplacement (longitude et latitude) et son rayon d'émission (déterminé et non modifiable).



Noter territoire est sous forme polytope et c'est le département de la Loire situé en France. Nous supposons que mesurer ou calculer la surface couverte par un ensemble d'antennes est une expérience coûteuse, dans le sens où plus nous souhaitons un damage précis du territoire, plus le nombre de calcul de surface associée à chacun des points de ce damage sera important, et donc in fine, plus la puissance de calcul nécessaire sera importante. Ainsi nous ne considérerons, comme c'est le cas dans les analyses actuelles que seules les expériences nexpl et à partir desquelles nous pourrions créer un modèle de prédiction pour toutes les autres valeurs possibles de localisation d'antennes.

Bien que notre objectif final soit de maximiser la surface du territoire couvert par les antennes en faisant varier leur localisation, nous allons rassembler des

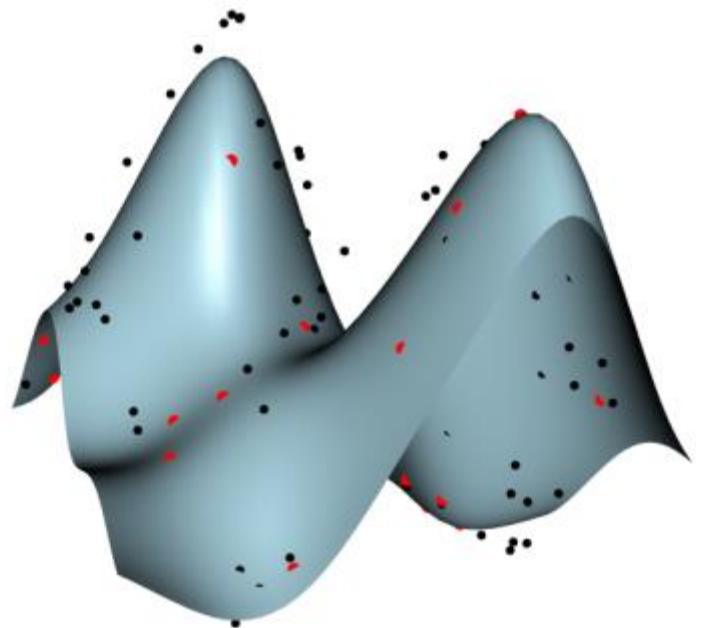
analyses complémentaires à partir de méthodes diverses : au lieu de fournir une seule réponse, nous progresserons par étapes régulières et appliquerons des méthodes classiques afin de répondre à cette question.

Pour ce faire nous parcourons les domaines de : conception d'expériences, méta modélisation, analyse de sensibilité et finalement optimisation globale et locale.

Nous allons nous intéresser à deux problèmes de complexité croissante. Le premier est un problème a deux dimensions, constitué de deux antennes dont l'une est fixe dans le centre et la deuxième dont la position est libre dans le département de la Loire. Le second problème, de même nature que le premier, avec un problème de dimension passant de deux à six à travers la possibilité de mouvoir 3 antennes et donc : trois longitudes et trois latitudes.

Le but est donc de construire un modèle de prédiction à partir de données d'entraînement et de pouvoir en tirer une approximation la plus exacte possible.

(ici on peut dissocier les points d'apprentissages : rouge, et ceux en bleu, produit de la prédiction qui diffèrent de la valeur réelle (surface dorée)



Nous disposons des données suivantes :

- **antennes_2d_train.Rdata, antennes_6d_train.Rdata** contenant les données d'entraînement (en 2 dimensions et en 6 dimensions) :

C est une matrice 20×2 et 300×6 matrice quand $d = 2$ et 6 .

S est la matrice de surface associée aux points de coordonnées de C

(C de taille $n_{exp} \times dimension$, et S de taille $n_{exp} \times 1$)

- **antennes_2d_test.Rdata, antennes_6d_test.Rdata** contenant les données de test avec lequel nous pouvons juger de la qualité du prédicteur :

C est une matrice 100×2 et 300×6 respectivement pour $d = 2$ et 6 .

Il n'y a logiquement pas de donnée S puisqu'il s'agit des données

recherchées (la donnée rendu sous forme de fichier txt et qui atteste de la qualité ou non de notre prédicteur).

Les mêmes données sont accessibles depuis

Matlab avec les mêmes caractéristiques.

(Les données se composent ainsi de la manière suivante (idem en 6d) sous forme vectorielle avec des valeurs prises pour le cas 2d :

S est compris entre 12 et 33 avec

C[1] compris entre -6 et 5 et C[2] entre -6.5 et 8.7)

	S
1	14.56342
2	18.77720
3	29.73057
4	22.55975
5	12.56742
6	17.69296
7	31.52944
8	16.30068
9	19.51646
10	21.75889
11	20.87177
12	17.16315
13	22.92938
14	14.66199
15	16.49782
16	33.32831
17	19.73824
18	24.48183
19	12.56742
20	29.39791

	C
1	2.7748256 1.8972772
2	-4.4270497 -2.4462527
3	0.2365979 -4.0732366
4	-0.3340344 0.3768325
5	3.4120126 4.8243961
6	0.5512095 7.8398642
7	-2.0351642 6.0860091
8	-2.6407512 -5.9391810
9	-4.1037863 -0.8681943
10	4.2572768 -3.0626376
11	-3.3175870 -1.5218766
12	-5.1829300 8.2974310
13	-1.2857373 0.7060805
14	2.3650053 3.4875971
15	1.7187010 5.3062923
16	1.4377355 -3.4444437
17	-5.7555188 6.5789084
18	-0.7728201 -4.8767850
19	3.7607500 3.2827259
20	-2.8165880 2.3702229

2. SVM/SVR

Dans un premier temps nous nous intéressons au modèle SVM et SVR (pour Support Vector Machine et Support Vector Regression)

Dans notre projet, seule la partie de régression (SVR) nous intéresse, puisque notre but est d'approcher au mieux les valeurs prises par la fonction surface à partir de la localisation de l'antenne, cependant, la partie SVM nous sert de base de compréhension du modèle de régression.

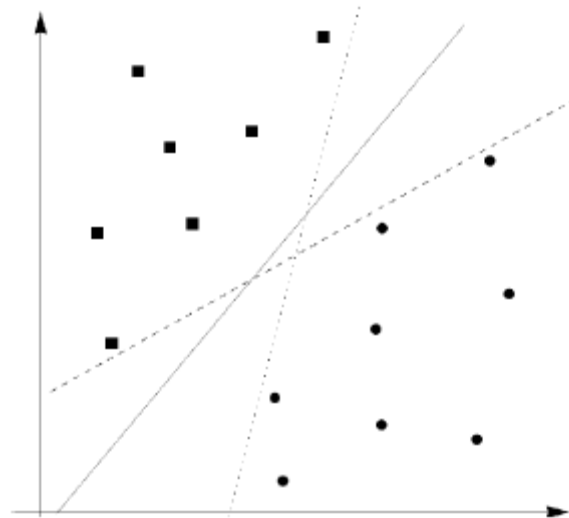
La base de la SVM se situe dans la classification : il s'agit de séparer deux populations à l'aide d'un hyperplan d'équation $y = w.x + b$ d'inconnu w et b .

La détermination d'un tel hyperplan n'est pas complexe en soit, le problème se situe dans l'infini des solutions :

Pour un jeu de données, une infinité d'hyperplan existe afin de délimiter les deux classes.

L'intérêt de la SVM se situe donc dans la recherche « du meilleur hyperplan.

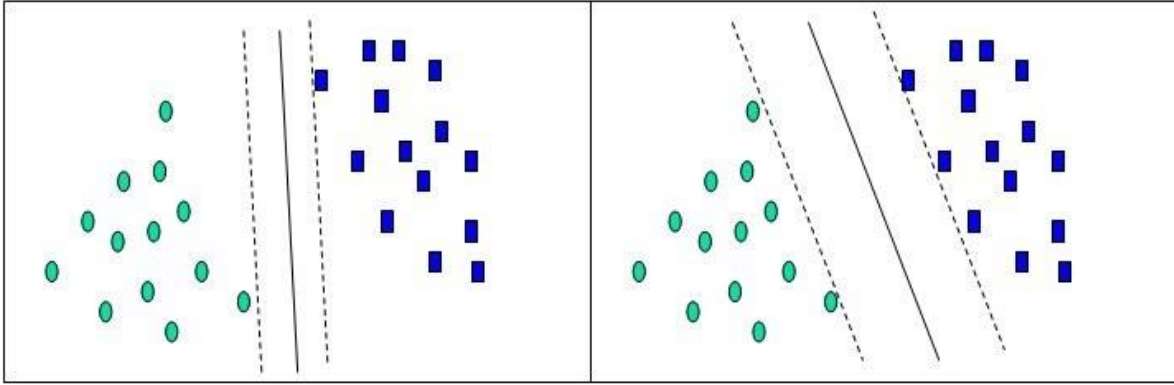
Ce qui nous amène à nous demander la nature du critère qui nous permet de juger si notre hyperplan est judicieux ou non.



Ce critère correspond à la marge entre les deux classes : un hyperplan sera considéré comme meilleur dès lors qu'il sépare au mieux les deux classes.

Cette marge se définit de la manière suivante : $M = 1 / ||w||$

D'où la nécessité de minimiser $||w||$ afin de maximiser la marge.



Dans le premier cas la marge sépare de manière correcte les deux classes, mais semble faible et améliorable, dans le second cas, les deux classes sont séparées par un hyperplan qui maximise la distance entre les points (de support) de chaque classe. Ces points, qui constituent la frontière entre appartenance à la classe et marge sont les points clés qui définissent l'hyperplan H.

La théorie du modèle de SVM s'appuie sur le Lagrangien lié à cette maximisation, sous certaines conditions :

On définit le lagrangien :

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^l \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1).$$

Sous la condition :

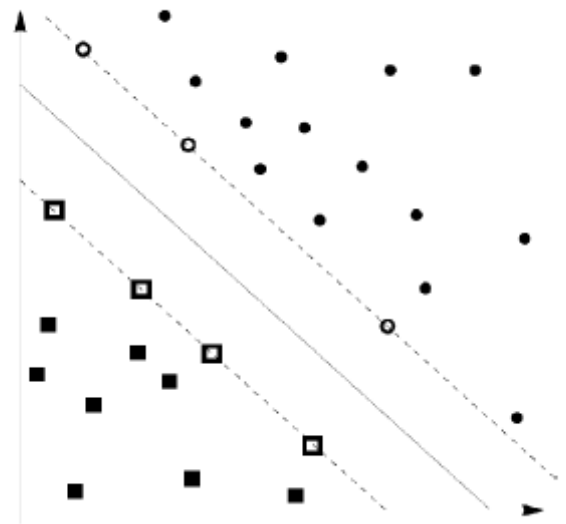
$$-(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) \leq 0, \quad i = 1, \dots, l.$$

(Qui consiste ici à classer les points test entre 2 classes 1 et -1.)

La résolution d'un tel problème est permise en passant par la définition d'un problème dual (à opposer au problème primal exposé plus haut).

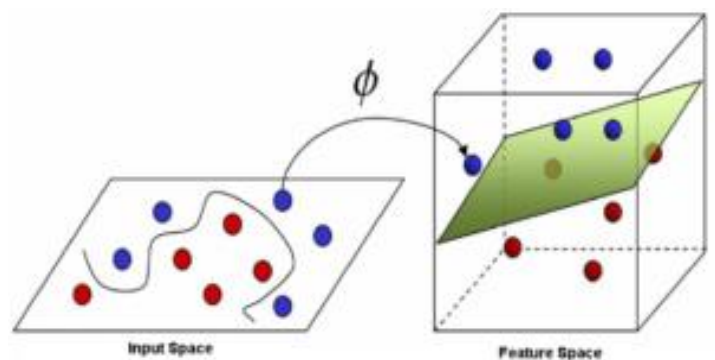
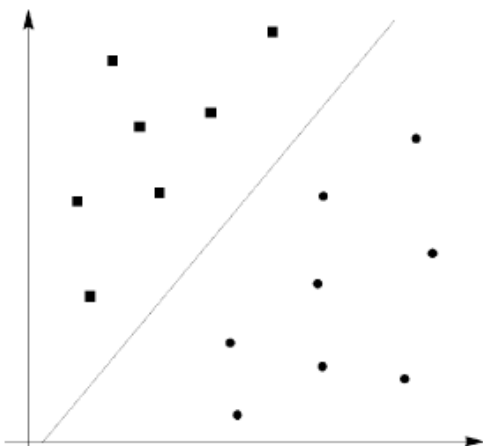
En finalité on peut estimer les inconnues primales w et b (après avoir estimé les inconnues duales introduites) et donc définir l'hyperplan H .

Cet hyperplan est ainsi défini à l'aide des vecteurs de support (d'où le nom de la théorie), en effet, ils définissent la marge et donc l'hyperplan en question.



Enfin, et pour expliquer la présence de noyau dans la suite de la démarche de recherche il faut introduire les cas où les données ne sont pas séparables telles quelles.

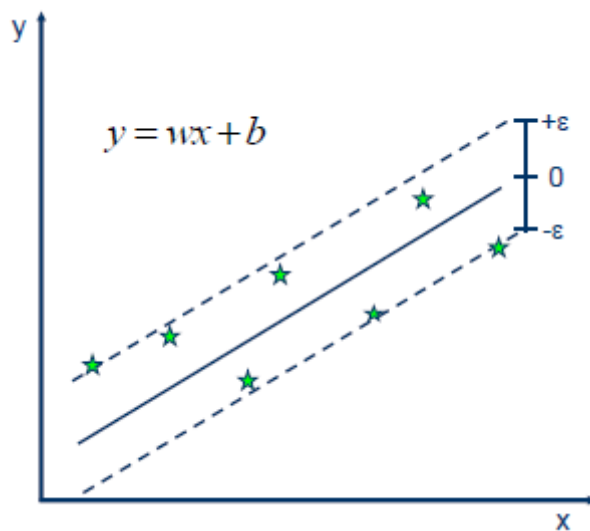
Voyons cela à travers les figures suivantes :



Dans le premier cas un hyperplan existe afin de séparer les données, dans le second cas on passe par une fonction auxiliaire, le noyau, afin de pouvoir trouver un espace (de dimension plus grande souvent) dans lequel les données sont séparables.

Le reste de la théorie reste inchangé et nous permet de dissocier nos deux classes de manière optimale.

Revenons alors au problème initial de régression de nos données à l'aide du modèle SVR. Tout ce qui a été abordé lors de la SVM va nous être utile, seule la conclusion va être différente et les contraintes que l'on fixe lors de la maximisation :



• **Solution:**

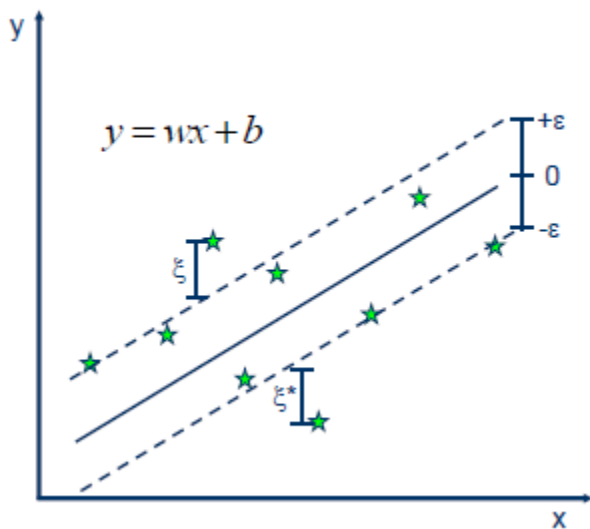
$$\min \frac{1}{2} \|w\|^2$$

• **Constraints:**

$$y_i - wx_i - b \leq \epsilon$$

$$wx_i + b - y_i \leq \epsilon$$

Cependant, dans une volonté de trouver le meilleur prédicteur, nous allons également considérer les cas où l'on aurait des données « absurdes » ou très éloignées du reste du jeu, et cela en introduisant un coefficient d'erreur :



• **Minimize:**

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

• **Constraints:**

$$y_i - wx_i - b \leq \epsilon + \xi_i$$

$$wx_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

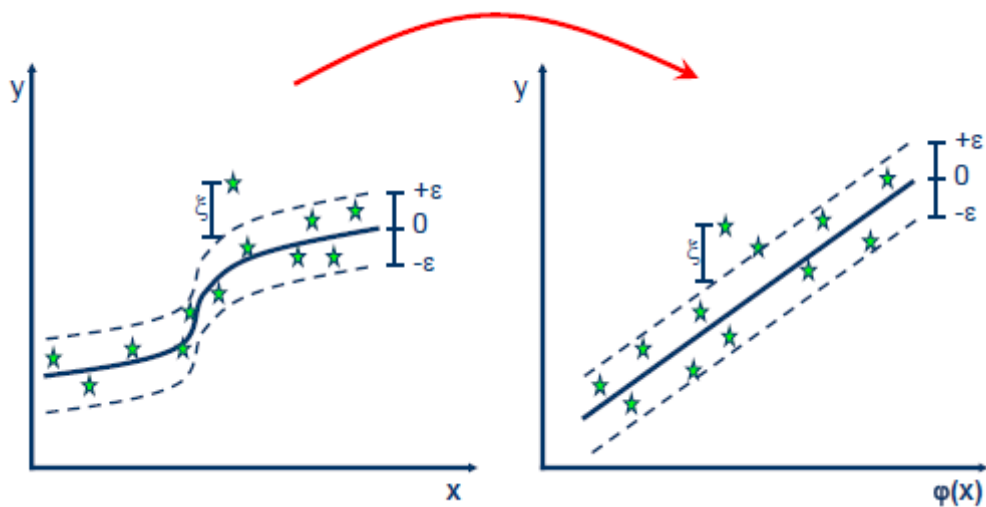
Ainsi nous créons un modèle de régression défini grâce à l'hyperplan H.

La suite de la résolution (détermination des constantes w et b) est similaire à celle de la partie SVM et constitue une partie mathématique.

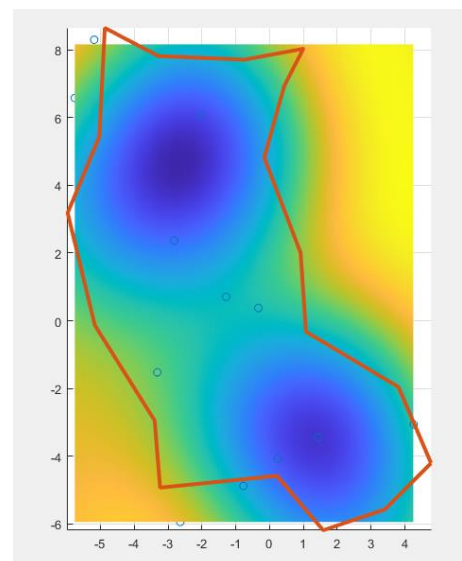
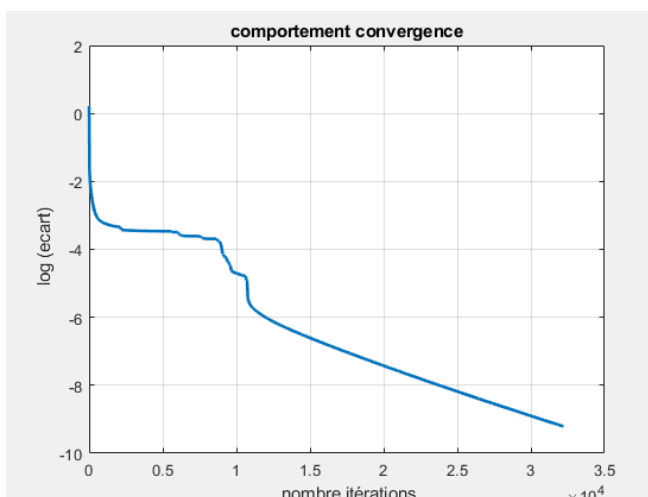
Le cas de données non séparables linéairement est également abordable avec un modèle SVR avec l'introduction des fonctions noyaux (Kernel)

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b$$

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b$$



Ce modèle de régression SVR, nous permet donc de créer un fichier de données correspondant aux surfaces prédites **Couv_2/6d_svr** à partir de nos données test.



3. Metamodelling et krigage

Dans cette seconde partie, nous nous intéressons à une méthode de régression par Processus Gaussien : le krigage, il s'agit d'un apprentissage statistique dans des espaces de grande dimension par rapport au nombre de données.

L'idée de base du krigage est de prévoir la valeur de la variable régionalisée étudiée en un site non échantillonné par une combinaison linéaire de données ponctuelles adjacentes.

Le modèle de base du krigage a la même forme que le modèle de régression classique ou locale, mais les erreurs sont maintenant supposées dépendantes spatialement.

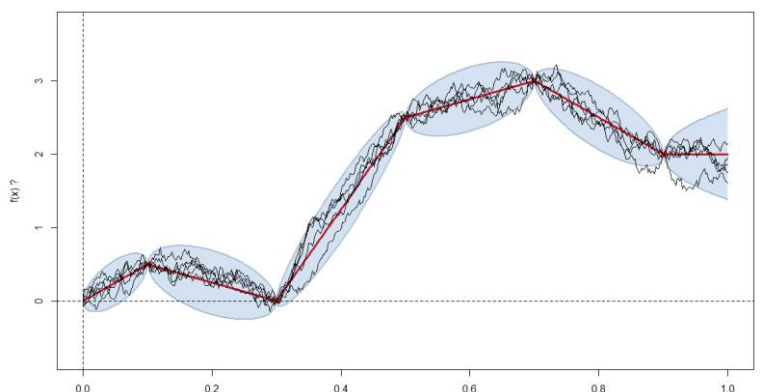
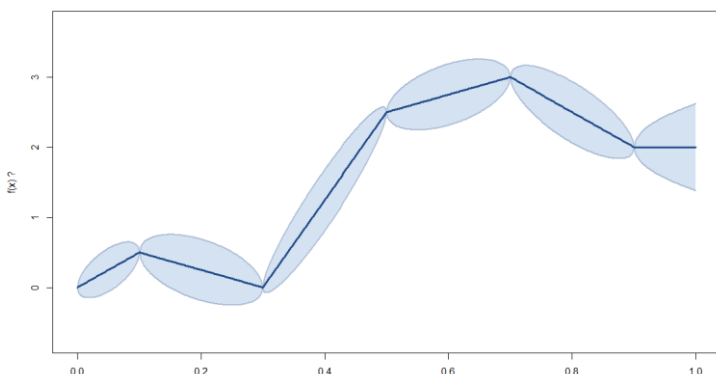
On définit le modèle avec la formule suivante :

$$Z(s) = \mu(s) + \delta(s), \quad s \in D$$

avec μ la structure déterministe pour l'espérance de Z , et δ la fonction aléatoire stationnaire, d'espérance nulle et de structure de dépendance connue.

Dans ce modèle, il est également question du choix d'un noyau : brownien, gaussien, exponentiel, etc.

Dans ce modèle, à partir de nos points d'entraînement on peut construire un ensemble de vraisemblance des valeurs (bornes définies en bleu clair)



La moyenne est représentée en bleu foncé à gauche, et on peut voir sur la figure de droite les différentes trajectoires entre chaque points connu (5 points qui regroupent les intervalles de confiance) lors de simulation.

Cet exemple est fondé sur un noyau brownien, construit tel que :

$$k(x, x') = \min(x, x') = x \wedge x'$$

Le processus est construit de la manière suivante :

On considère un processus Gaussien Y de tel sorte que l'on est une décomposition orthogonale suivante avec le vecteur Béta solution des équations de Krigeage

$$Y(x) = \beta_1(x)Y(x_1) + \dots + \beta_n(x)Y(x_n) + \varepsilon(x) \quad k(X, X)\beta(x) = k(x, X)$$

Dans cette écriture, le vecteur x correspond aux données de test, dont l'image à travers la fonction sont inconnues, X , qui correspond aux données d'apprentissage, et k qui correspond au noyau choisit.

A partir de ses équations on peut définir la moyenne de krigeage $m(x)$.

$$m(x) = y^T k(X, X)^{-1} k(x, X) = k(x, X)^T k(X, X)^{-1} y$$

Où y correspond à l'image des données d'apprentissage.

Ainsi que la loi conditionnelle de Y (loi a posteriori).

$$k_\varepsilon(x, x') = k(x, x') - k(x, X)^T k(X, X)^{-1} k(x', X)$$

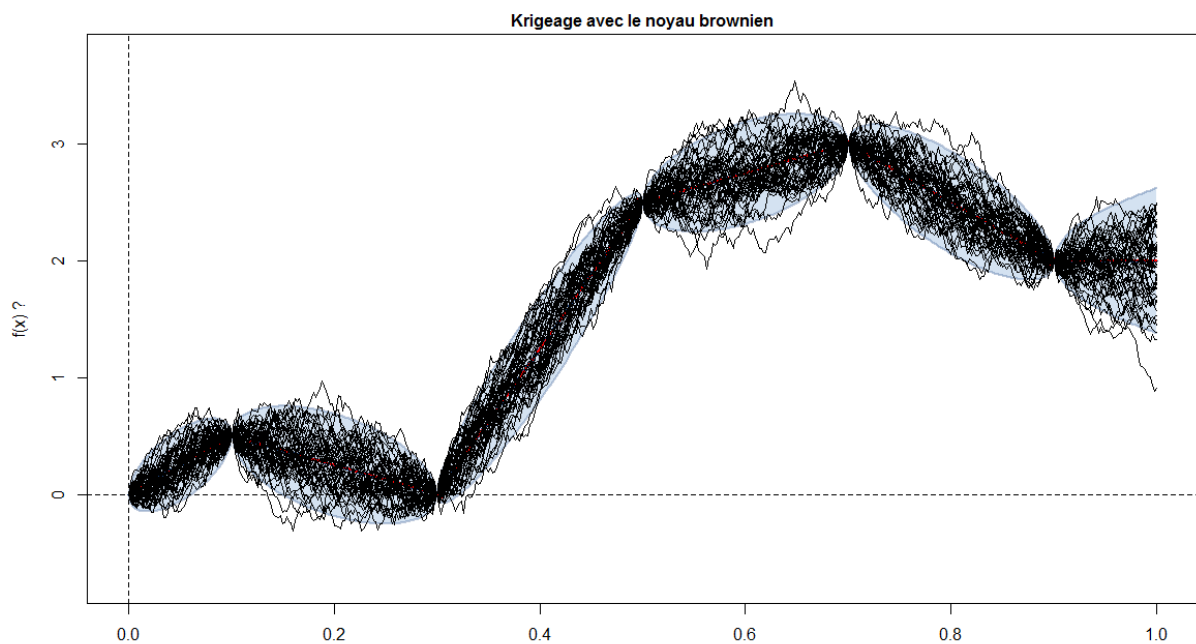
A partir de ces données théoriques sur le krigage, on peut construire nos propres outils, on choisit notre noyau, les données d'apprentissages, et on crée nos matrices à travers le noyau en question :

On définit ici les outils nécessaires et on peut calculer les valeurs de moyenne et de variance (défini plus haut) afin de pouvoir considérer un intervalle de confiance.

```
kxx <- kBrown(x,x)
kxx <- kBrown(x,x)
kxx <- kBrown(X,X)
invkxx <- solve(kxx)
mu.krig <- kxx%%invkxx%%yx
K.krig <- kxx - kxx%%invkxx%%t(kxx)
```

On remarque en réitérant l'expérimentation un grand nombre de fois (ici 50), que les valeur prises

```
param <- 1
var.krig <- param*diag(K.krig)
upp95 <- mu.krig + 1.96*sqrt(pmax(0,var.krig))
low95 <- mu.krig - 1.96*sqrt(pmax(0,var.krig))
```



par la variable y est bien comprise dans l'intervalle ainsi défini.

Ceci fait, on peut appliquer ce modèle de régression à nos données de localisation d'antennes.

Pour ce faire :

On détermine dans un premier temps le noyau le plus adapté en calculant l'erreur associée au modèle avec les données de la phase d'apprentissage.

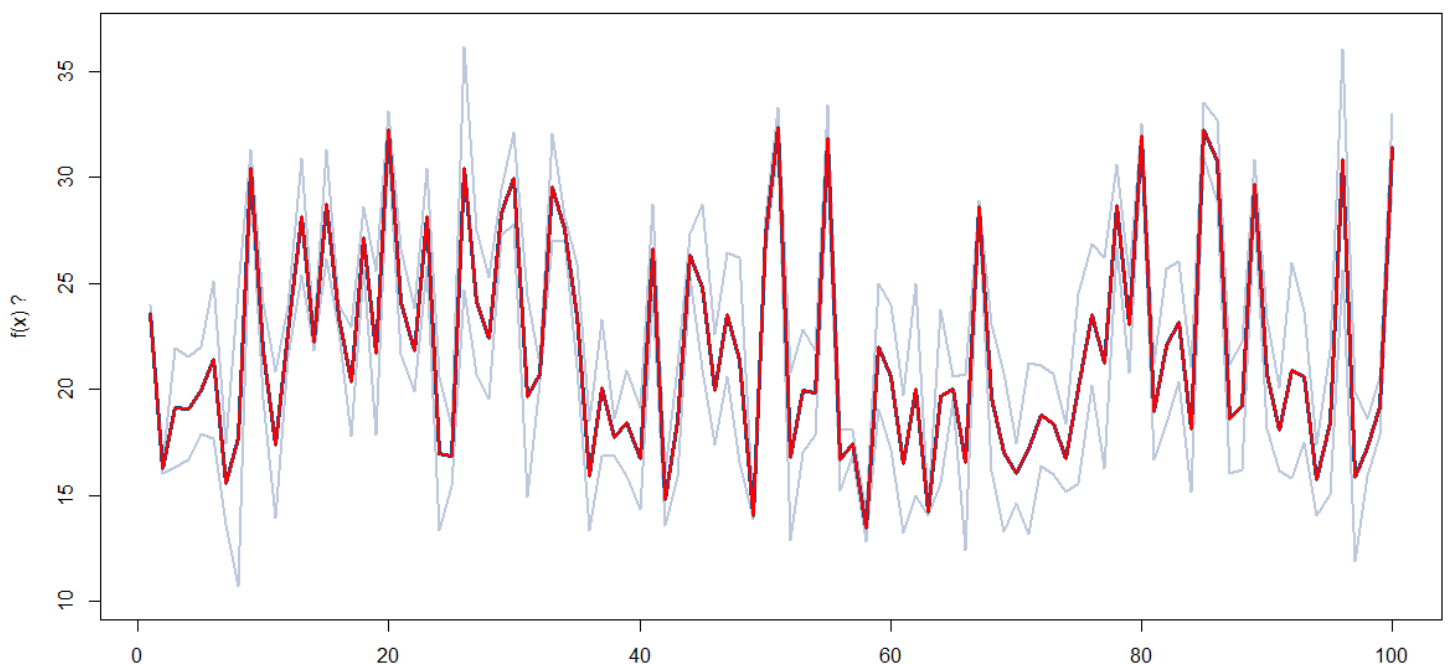
"The best kernel for the kriging is : gauss ,with an RMSE of : 0.09"

Parmi les différents noyaux : gaussien, exponentiel, matern5.2 et matern3.2, le plus performant en termes d'erreur correspond au noyau gaussien.

Afin d'attester de la validité de nos valeurs, on peut mettre en place un système de **cross-validation**, ou validation croisée lors de laquelle, nous allons prendre comme données d'apprentissage les 19 des 20 premières valeurs de C, et appliquer la modèle prédictif à la dernière donnée.

Ce système admet tout de même certaines limites puisque, plus on veut considérer de données d'apprentissages : pour le parfaire et l'améliorer, moins on a de données issues du modèle pour le valider.

Ainsi, on se retrouve ici avec seulement 1 donnée d'erreur (écart en la vraie valeur et la valeur prédite), il y a donc un équilibre entre le nombre de données d'apprentissage et la validation du modèle à partir du reste des données connues.



A partir du modèle de krigeage, on peut afficher les valeurs prédites ainsi que l'intervalle de confiance lié à la variance du modèle :

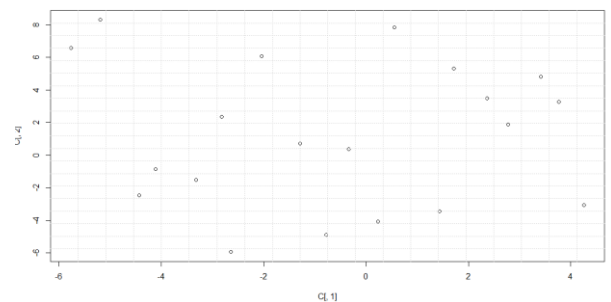
On peut alors encadrer chaque surface associée à la localisation des antennes entre deux valeurs de vraisemblance.

Ce modèle de régression par krigeage, nous permet donc de créer un fichier de données correspondant aux surfaces prédites **Couv_2/6d_krigeage**.

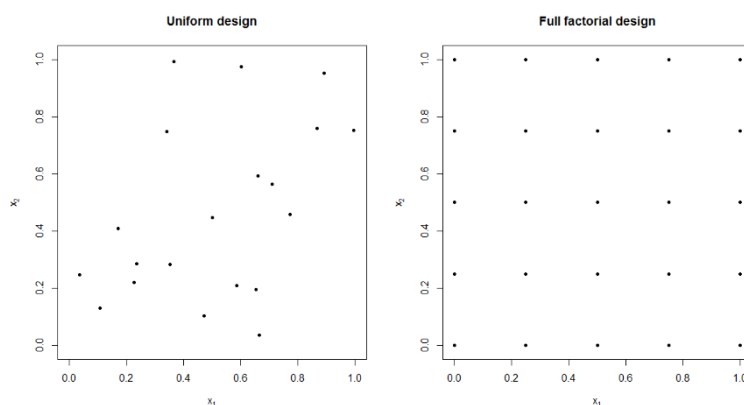
4. Planification d'expériences

Il s'agit dans cette partie de proposer 4 nouveaux modèles afin d'améliorer l'actuel planification d'expérience.

Le modèle ci-dessous, qui nous sert de données d'apprentissages correspond à un design uniforme

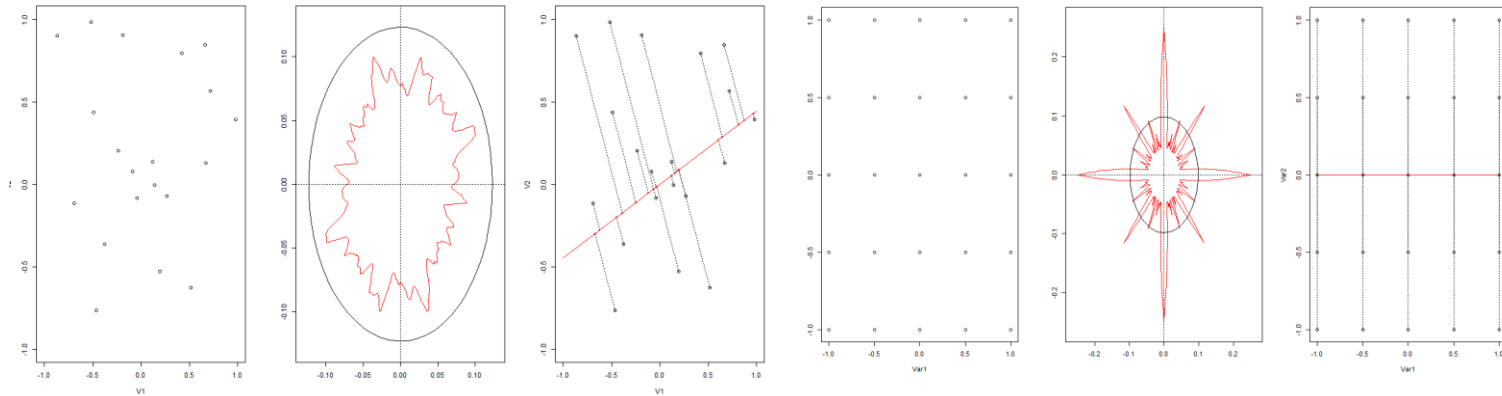


Les deux modèles équivalents qui sont de type uniforme et full factoriel et qui sont de même dimension sont représentés ci-dessous :



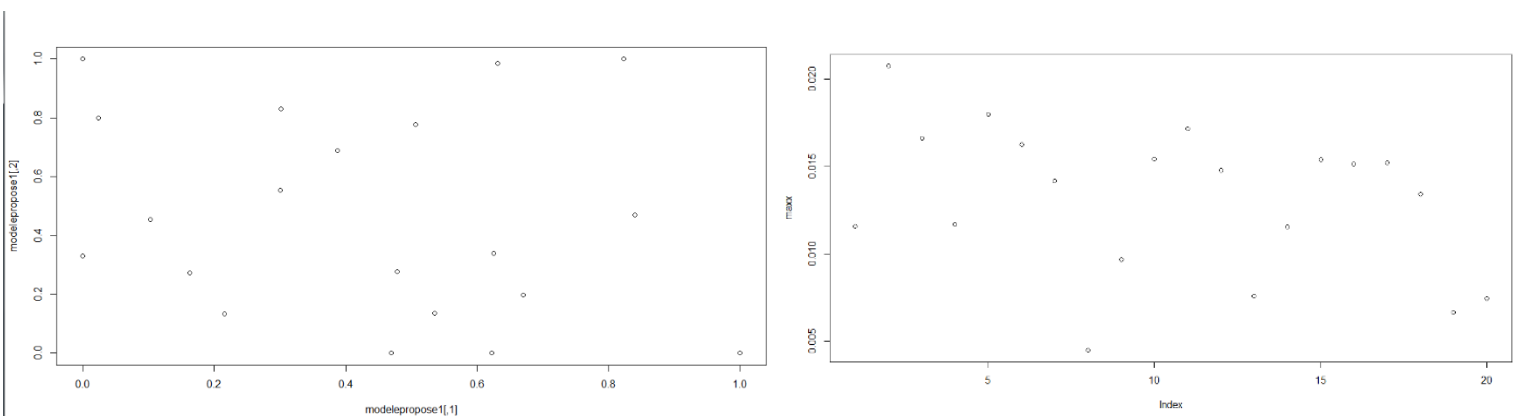
A l'aide de la fonction `rss2d`, on peut tracer les deux figures ci-contre, sur lesquelles on peut voir d'une part :

La surface n'est pas totalement recouverte pour le cas uniforme, mais sans présence de redondance, dans le second cas, full factoriel, le recouvrement est plus important, mais on peut observer la présence de redondance.

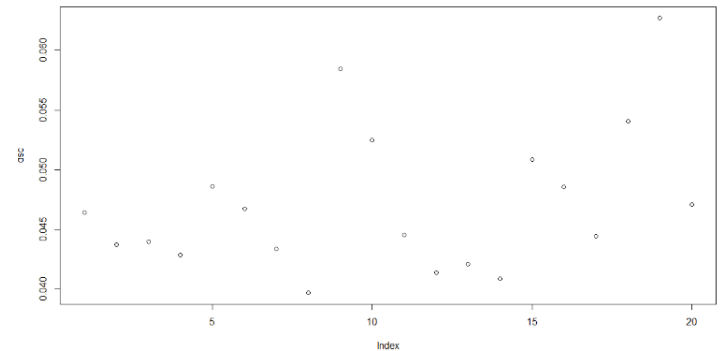
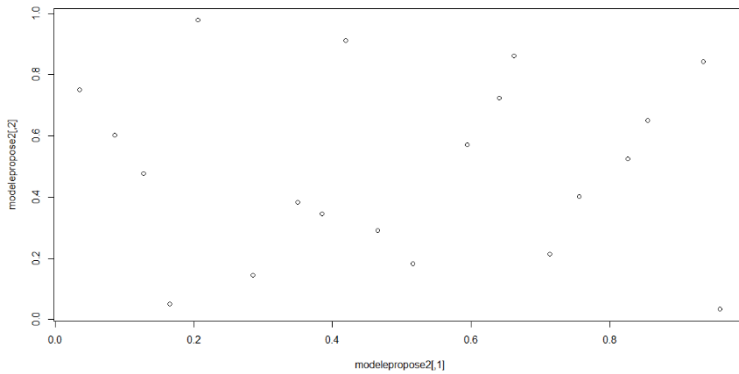


On peut également à partir d'une partition aléatoire (à gauche) , déterminer le max issu de la fonction maximin, qui génère une conception de remplissage de l'espace sous le critère de la distance maximale ; des recherches discrètes et continues sont fournies

On peut voir sur les figures suivantes, la distance minimale à chaque itération, ainsi que le modèle proposé qui maximise ce critère de distance.



De plus, à l'aide du critère « discrepancy criteria » on peut définir un nouveau modèle qui se base la minimisation de degré de non-uniformité.



5. Analyse de sensibilité

Dans cette partie, nous nous intéressons à la décomposition de la réponse de notre prédicteur, c'est-à-dire la surface S , noté $S(X_1, X_2)$ pour mettre en avant l'implication des deux antennes X_1 et X_2 lors du calcul de la surface (respectivement $S(X_1, X_2, X_3, X_4, X_5, X_6)$ pour les six antennes).

L'analyse de sensibilité nous permet d'écrire cette surface comme une somme de termes :

$$S_0 + S_1(X_1) + S_2(X_2) + S_{12}(X_1, X_2)$$

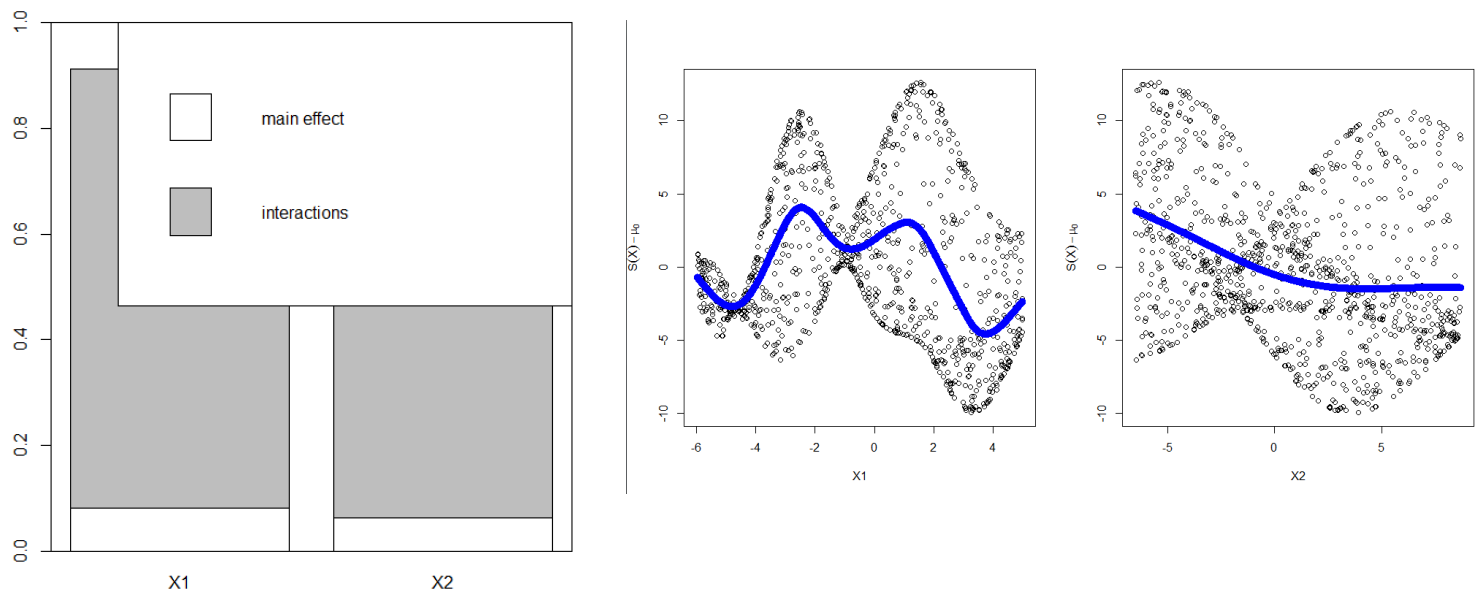
On met alors en avant une partie commune à chaque antenne, S_0 ainsi que 3 autres membres : $S_1(X_1)$ propre à l'influence de la première antenne et de même $S_2(X_2)$ ainsi qu'un terme d'influence conjointe des antennes X_1 et X_2 : $S_{12}(X_1, X_2)$.

Nous allons nous intéresser à la constante S_0 ainsi qu'à l'impact des antennes X_1 et X_2 à travers deux valeurs que sont la variance de S_1 et celle de S_2 .

Pour cela, nous allons décomposer la surface selon les conditions de Sobol afin de rendre cette décomposition unique.

A partir des résultats fournis par les fonctions fast99, Morris on peut établir l'influence de chacun des membres qui composent la surface finale, via la part de la variance de S1, S2 et S12 par rapport a la variance de S.

On a ainsi un pourcentage de variance expliquée pour chacun des termes, nous permettant de répondre à la question initiale : quelle est la part d'influence de X1 et X2, et donc en d'autres termes, l'importance de la longitude par rapport à la longitude dans l'emplacement de l'antenne.



D'après les résultats, les interactions des variables X1 et X2 sont plus importantes combinées que celle de chacune séparément.

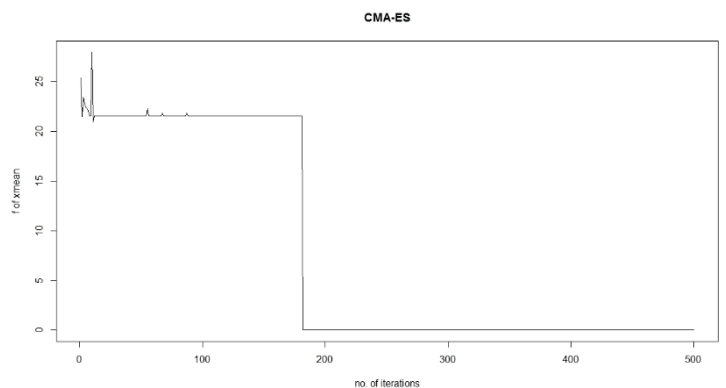
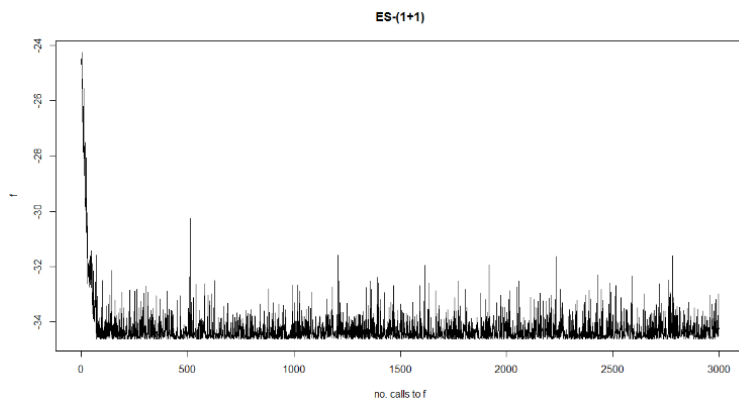
On peut parallèlement à cela faire de même dans le cas à 6 dimensions, c'est-à-dire l'influence des 3 longitudes et 3 latitudes, avec une expression de la surface qui dépend cette fois au minimum de 6 termes : S1,S2,S3,S4,S5 et S6, avec d'autres termes eux potentiellement nul S12,S23 etc.

6. Optimisation globale

Dans cette partie on s'intéresse à l'optimisation globale de notre modèle, pour ce faire, nous remplaçons le problème initial de maximisation de la surface $S(X)$ par la maximisation de la valeur $m(x)$, moyenne du krigeage à travers la prédiction.

Pour cela, nous développons un modèle d'optimisation avec une recherche de type « normal » et utilisons l'algorithme CMA-ES, qui repose sur l'adaptation, au cours des itérations, de la matrice de variance-covariance de la distribution multi-normale utilisée pour la mutation.

A partir de ce modèle nous pouvons déterminer à partir de ES-(1+1) et CMA-ES



En 2d on peut trouver les résultats suivants pour CMA

```
xbest=> optresCMA$x_best  
[1] -5.942156 -3.516663  
> cat("fbest=")  
fbest=> -optresCMA$f_best  
[1] 22.69062
```

Après une itération de EGO on obtient les résultats pour CMA :

```
> x_solution_EGO_2d  
[1] -5.233751 -5.490104
```

L'intérêt de cette méthode s'appuie sur la complémentarité entre modèle de krigeage, peut gourmand en ressource numérique, et adapté au grande dimension, couplé à l'algorithme CMA-ES.

Dans un second temps, nous utilisons l'algorithme EGO, afin d'améliorer encore les résultats : et ce, à partir de nos valeurs de « Expected improvement » qui nous

permettent, à chaque itération, de placer un nouveau point pour notre prédiction de la fonction.

Dans ce cas, on maximise l'amélioration attendu $EI(x)$ avec l'algorithme de CMA-ES (car notre fonction est multimodale) afin une de fois de plus, d'optimiser la recherche du nouveau point après une itération de EGO.

7. Conclusion

Le fil rouge de cette étude se décompose en plusieurs parties à partir du problème suivant : nous faisons face à un problème de maximisation de surface dans le cadre du déploiement d'antennes sur le territoire de la Loire.

Notre démarche s'articule autour de plusieurs axes :

Un premier, d'expérimentation des méthodes pré existantes telles que le krigeage ou le modèle SVR (issu de la théorie du SVM) afin d'estimer au mieux une régression à partir de données d'apprentissage et d'en construire un modèle de prédiction pour les données test.

Un second, d'optimisation local des modèles, afin d'agir au mieux sur les paramètres internes à ces méthodes et d'en tirer la meilleure approximation, le meilleur modèle possible.

Nous avons également pu nous intéresser à la décomposition de notre problème, en définissant notre paramètre à déterminer : la surface, comme une entité composée de l'influence de plusieurs paramètres et ainsi mettre en avant les paramètres prédominants.

Enfin, dans une démarche d'optimisation globale du processus, nous nous sommes intéressés à des algorithmes tel que CMA-ES et EGO afin d'affiner le modèle que nous avons développé, en redéfinissant le problème avec de nouveaux paramètres à optimiser.