

Société de Calcul Mathématique SA

Outils d'aide à la décision



Amélioration d'outils

pour la Sûreté Nucléaire :

Comparaison entre l'EPH et le Krigeage

Rapport adressé à

l'IRSN/ Pôle Sûreté Nucléaire
(à l'attention de M. Yann Richet)

par la

Société de Calcul Mathématique SA

Rédaction Frédéric Godan et Olga Zeydina

Résumé opérationnel

L'IRSN utilise actuellement un algorithme appelé EGO, qui fait appel à une technique standard appelée "Krigage", pour reconstituer l'information et évaluer les situations où aucune mesure n'a été faite. L'IRSN souhaite maintenant savoir si le concept d'Hypersurface Probabiliste (EPH), introduit par la SCM (voir référence [PIT]) donne de meilleurs résultats dans une situation test : celle où les valeurs sont générées par une fonction particulière, dite de Branin-Hoo.

Les caractéristiques de l'EPH sont les suivantes :

- L'EPH est utilisée pour reconstruire ou prédire des informations en se basant sur l'existant, sans aucune hypothèse de modèle ;
- La méthode peut être appliquée quel que soit le nombre de points de mesure ;
- La construction des points recherchés n'est pas basée sur des lois factices ;
- La méthode étant probabiliste, l'EPH est très robuste en termes d'incertitudes.

1. Implémentation

Pour répondre à la question posée par l'IRSN, la SCM a tout d'abord implémenté la méthode EPH en langage S pour que les fonctions soient exploitables avec le logiciel R. Cette implémentation est intéressante en soi, car elle permettra dans l'avenir toutes sortes d'expérimentations, outre celle qui est menée ici.

Dans un premier temps, et pour respecter le formalisme utilisé dans le package DiceKriging, la SCM a implémenté une fonction *eph* qui crée un objet de type *eph*, représentant les informations à disposition de l'utilisateur avant reconstruction, à savoir :

- La valeur du min et du max pour chacun des paramètres d'entrée et de sortie (indiqués à dire d'expert) ;
- La valeur des mesures en chaque point d'observation ;
- Le pas de discrétisation des différents paramètres d'entrée et de sortie.

La SCM a ensuite implémenté une fonction *predict.eph* qui prend en entrée un objet de type *eph* et une matrice comportant les coordonnées des points dont on souhaite faire une reconstruction.

Cette fonction estime la médiane, les quantiles 5 et 95 de la loi de probabilité obtenue par l'EPH ainsi que son espérance en chaque point dont on souhaite une reconstruction.

Enfin, pour obtenir un résultat analogue à l'optimiseur global EGO, la SCM a mis en place la méthode de recherche des zones dangereuses pour trouver les points où il manque de l'information.

Cette fonction prend en entrée un objet de type eph, le seuil souhaité par l'utilisateur et le type d'extremum recherché : il est possible de rechercher les extrema hauts ou bas de la fonction objectif.

La fonction renvoie une ou plusieurs possibilités de points à explorer, chacun appartenant à une zone dangereuse de l'espace des solutions.

2. Comparaison EPH et Krigeage

Ensuite, conformément à la demande formulée par l'IRSN, la SCM a confronté les résultats obtenus par Krigeage et par l'EPH dans différentes configurations sur 400 points de la fonction Branin-Hoo.

Nous avons mis en place trois outils de mesure pour procéder à ces comparaisons, un test et deux mesures :

Soit N le nombre de points reconstruits.

Soit X le vecteur de N termes des points estimés par EPH.

Soit Y le vecteur de N termes des points estimés par Krigeage.

Soit R le vecteur de N termes des valeurs réelles de la fonction.

On note respectivement X_n , Y_n et R_n le n ème coefficient des vecteurs X , Y et R .

Test 1 :

Nombre de points pour lesquels la valeur réelle est en dehors de les bornes à 95% déterminés par les quantiles 2,5 et 97,5 de la loi de probabilité. C'est un test très grossier.

Deux mesures :

Dist_1 :

Elle représente l'écart entre la valeur moyenne (valeur assignée par la reconstruction) et la valeur réelle :

- $Dist_1 = \frac{1}{N} \sum_{n=1}^N |X_n - R_n|$ pour l'EPH.
- $Dist_1 = \frac{1}{N} \sum_{n=1}^N |Y_n - R_n|$ pour le Krigeage.

EQ_2 :

Elle représente la moyenne des erreurs quadratiques, aussi appelé moyenne des risques quadratiques :

- $EQ_2 = \frac{1}{N} \sum_{n=1}^N (X_n - R_n)^2$ pour l'EPH.
- $EQ_2 = \frac{1}{N} \sum_{n=1}^N (Y_n - R_n)^2$ pour le Krigage.

3. Les résultats de la comparaison

Les points étant générés par la fonction de Branin-Hoo, les résultats de la comparaison sont très clairs :

- Dans tous les cas, les deux procédés de reconstruction passent le test qualitatif ;
- Lorsque l'information disponible est très importante (beaucoup de mesures disponibles), le Krigage donne de meilleurs résultats que l'EPH ;
- A l'inverse, lorsque l'information disponible est pauvre (peu de points de mesure), l'EPH donne de bien meilleurs résultats que le Krigage.

Lorsqu'on dispose de 16 points de mesure, le Krigage donne de meilleurs résultats ; avec 9 points et a fortiori avec 3 points, l'EPH l'emporte nettement.

Voici des résultats extraits de la comparaison :

Avec 16 points de mesure, le Krigage l'emporte (plus le nombre est faible, meilleur est le résultat) :

	Dist_1	EQ_2
Krigage	11	605
EPH	24	4 779

Tableau 1 : Comparaison Krigage et EPH par rapport à la valeur réelle, Branin à 16 points

Avec 3 points de mesure, l'EPH l'emporte de beaucoup :

	Dist_1	EQ_2
Krigeage	119	32 296
EPH	100	3 998

Tableau 2 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 3 points

Ces résultats confortent ceux obtenus dans l'article [ICAPP] Bernard Beauzamy, Hélène Bickert, Olga Zeydina (SCM), Giovanni Bruna (IRSN) : Probabilistic Safety Assessment and Reliability Engineering: Reactor Safety and Incomplete Information : l'EPH est tout à fait adaptée à la reconstruction de l'information dans les cas où celle-ci est très pauvre et où on ne peut faire d'hypothèse de modèle.

4. Recherche des zones dangereuses

L'optimiseur global EGO, utilisant le Krigeage, fonctionne selon un principe différent de l'EPH. Pour celle-ci, la SCM a mis en place une méthode de recherche des zones "dangereuses", qui se substitue à la recherche des extrema.

L'optimiseur global EGO et la méthode développée par la SCM ont donc été appliqués aux mêmes jeux de données.

Les deux méthodes renvoient des points à extrapoler différents, mais vont toutes les deux dans le bon sens.

Sommaire

I.	Introduction.....	8
II.	Présentation du besoin	8
III.	EPH en langage R.....	8
A.	Estimation par Hypersurface Probabiliste	8
1.	Méthodologie	8
2.	La méthode de comparaison.....	9
B.	Implémentation	10
C.	Fonction classe eph	11
D.	Fonction predict.eph.....	11
IV.	Exemple d'utilisation – Fonction de Branin-Hoo	12
A.	Estimation au point de coordonnées (0.5 ; 0.5) dans R^2 :	13
3.	Résultats avec Krigeage :	13
4.	Résultats avec EPH :	13
5.	Valeur réelle :	13
B.	Estimation surfacique dans R^2 :	14
1.	Reconstruction de 400 points à partir de 16 points de mesure	14
2.	Reconstruction de 400 points à partir de 9 points de mesure	17
3.	Reconstruction de 400 points à partir de 3 points de mesure	20
C.	Reconstructions à partir de points choisis de manière aléatoire	23
V.	Limites	26
VI.	Recherche des points à explorer	27
A.	Optimiseur Global EGO utilisé par l'IRSN	27
B.	Recherche des zones dangereuses, une méthode développée par la SCM	27
C.	Méthodologie	28
1.	Recherche des zones dangereuses.....	28
2.	Recherche du point le plus éloigné de H_1	29
3.	Calcul de la probabilité d'être au delà du seuil au point M_1	32
D.	Etape suivante de la méthode.....	33
1.	Tirage aléatoire dans un hyperplan.....	33
E.	Implémentation	34
VII.	Exemple d'utilisation	35
A.	Recherche d'optima	35
1.	Recherche d'un maximum	35
2.	Recherche d'un minimum	36
3.	Conclusion	38
B.	EGO n-steps sur le même jeu de données.....	38
VIII.	Références.....	39

IX.	Annexe :.....	40
A.	Estimation au point de coordonnées (0.5 ; 0.5) dans R^2	40
B.	Reconstruction de 400 points à partir de 16 points de mesure	40
1.	Krigeage Ordinaire :.....	40
2.	Hypersurface Probabiliste Experimentale :	41
C.	Reconstruction de 400 points à partir de 9 points de mesure	42
1.	Krigeage Ordinaire :.....	42
2.	Hypersurface Probabiliste Experimentale :	42
D.	Reconstruction de 400 points à partir de 3 points de mesure	43
1.	Krigeage Ordinaire :.....	43
2.	Hypersurface Probabiliste Experimentale :	44
E.	Reconstructions à partir de points choisis de manière aléatoire	45
1.	Fonction de Branin.....	45
2.	Autre fonction de test.....	45

I. Introduction

Le présent document représente la réponse apportée par la Société de Calcul Mathématique SA à la question posée par l'IRSN : Commande EX10/32001814 du 30.05.2014.

II. Présentation du besoin

L'IRSN effectue des recherches destinées à améliorer la Sûreté Nucléaire. Le présent projet concerne les règles fondamentales de sûreté relatives aux installations nucléaires de base autres que les réacteurs. La question se pose de savoir si le "design" de certaines configurations est suffisamment sûr. La difficulté tient au fait que, de manière générale, tous les paramètres ne sont pas connus, et ceux qui le sont apparaissent entachés d'une incertitude mal quantifiée. Comme l'IRSN l'a observé, le fait de "mailler" l'espace des paramètres ne conduit pas nécessairement à de bonnes estimations, car les situations "à risque" peuvent se trouver en dehors du maillage.

Pour reconstituer les données manquantes, et évaluer les situations où aucune mesure n'a été faite, l'IRSN utilise actuellement un algorithme appelé EGO, qui fait appel à une technique standard appelée "Krigage". L'IRSN souhaite maintenant savoir si le concept d'Hypersurface Probabiliste (EPH), introduit par la SCM (voir référence [PIT]) donne de meilleurs résultats.

III. EPH en langage R

A. Estimation par Hypersurface Probabiliste

1. Méthodologie

La méthode utilisée, l'Hypersurface Probabiliste (EPH), est basée sur un modèle mathématique qui introduit des méthodes probabilistes pour reconstruire ou prédire des valeurs en se basant sur les données disponibles. Le point-clé de cette construction est la propagation de l'information de points de mesure (données d'entrée) aux points recherchés. Le résultat obtenu pour chaque point recherché est une densité de probabilité. L'importance accordée aux données d'entrée dépendra de leur distance avec les points recherchés. Plus un point est proche du point recherché, plus il influera sur la densité de probabilité de ce dernier. Ici, la proximité se comprend au sens spatial.

La méthode est décrite dans le livre [PIT] (ref. ci-dessous).

Les caractéristiques de cette méthode sont les suivantes :

- L'EPH est utilisée pour reconstruire ou prédire des informations en se basant sur l'existant ;
- La méthode peut être appliquée quel que soit le nombre de points de mesure ;

- La construction des points recherchés n'est pas basée sur des lois factices ;
- Cette méthode étant probabiliste, l'EPH est très robuste en termes d'incertitudes.

2. La méthode de comparaison

Afin de comparer les deux méthodes, on va travailler sur un exemple-test très particulier. Les valeurs sont générées par une fonction test connue, à savoir la fonction de Branin,

La fonction de Branin est définie sur $[0,1] \times [0,1]$ par :

$$u_1 = 15x_1 - 5$$

$$u_2 = 15x_2$$

$$Branin(x) = (u_2 - \frac{5}{4\pi^2}u_1 + \frac{5}{\pi}u_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(u_1) + 10$$

On va se donner un certain nombre de points de base et reconstituer d'autres valeurs, par l'une et l'autre méthode ; le résultat de la reconstruction sera ensuite comparé à la valeur réelle indiquée par la fonction.

Les valeurs de la fonction de Branin sont connues en 16 points de l'espace.

x1	x2	Branin(x1,x2)
0,00	0,00	306
0,33	0,00	56
0,67	0,00	14
1,00	0,00	10
0,00	0,33	160
0,33	0,33	21
0,67	0,33	27
1,00	0,33	7
0,00	0,67	63
0,33	0,67	36
0,67	0,67	90
1,00	0,67	55
0,00	1,00	17
0,33	1,00	101
0,67	1,00	203
1,00	1,00	152

Tableau 1 : Valeur de la fonction de Branin en différents points

L'EPH sera utilisée ici pour extrapoler les valeurs de la fonction de Branin en des points inconnus du domaine. Ainsi, pour chaque point recherché, la densité de probabilité qui lui est associée est calculée.

Pour le point de coordonnées (0.8 ; 0.6) nous obtenons la densité de probabilité donnée ci-dessous.

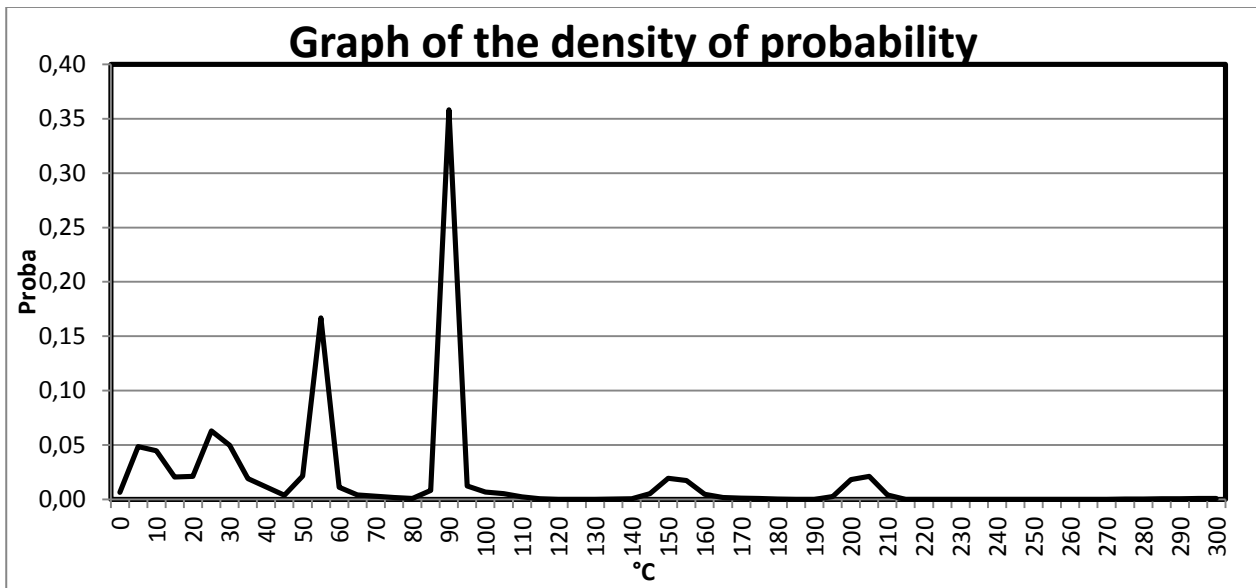


Figure 2 : Densité de probabilité de la valeur de la fonction de Branin extrapolée au point (0.8 ; 0.6)

L'espérance pour le point de coordonnées (0.8 ; 0.6) est 73.12 et la médiane vaut 90.

La valeur choisie pour l'extrapolation est l'espérance, car c'est la plus robuste : c'est celle qui minimise l'erreur quadratique sur le long terme.

B. Implémentation

La SCM a implémenté en langage R l'EPH sous sa forme la plus générale : la dimension de l'espace est choisie par l'utilisateur, le nombre de points de mesure est également arbitraire. L'utilisateur fournit en entrée :

- La valeur du min et du max pour chacun des paramètres d'entrées et de sorties (indiqués à dire d'expert) ;
- La valeur des mesures en chaque point d'observation ;
- Les points sur lesquels l'utilisateur souhaite appliquer l'EPH ;
- Le pas de discrétisation des différents paramètres d'entrées et de sorties.

Pour ce faire, la SCM a implémenté plusieurs fonctions :

- la fonction class eph qui crée un objet de classe eph.

- la fonction `predict.eph` qui estime la valeur de l'espérance, la médiane et des quantiles 5 et 95 de la loi de probabilité obtenue par l'EPH en n'importe quels points du domaine.

C. Fonction classe *eph*

La fonction *eph* est une fonction qui crée un objet de classe *eph*. Elle prend en argument :

- Design : La valeur des mesures en chaque point d'observation ;
- Response : La valeur des mesures en chaque point d'observation ;
- Boundaries : La valeur du min et du max pour chacun des paramètres d'entrée ;
- BoundariesP : La valeur du min et du max pour chacun des paramètres de sorties (indiqués à dire d'expert) ;
- Step : Le pas de discrétisation des différents paramètres d'entrées et de sorties.

La fonction renvoie un modèle *eph* dont les composantes sont :

- x : Les coordonnées de chaque point d'observation ;
- y : La valeur des mesures en chaque point d'observation ;
- NofParam : Le nombre de paramètres d'entrée ;
- NofMeas : Le nombre de points d'observation ;
- NofInk2 : Le nombre de paramètres de sortie ;
- Boundariesin : La valeur du min et du max pour chacun des paramètres d'entrée ;
- Boundariesout : La valeur du min et du max pour chacun des paramètres de sortie ;
- Step : Le pas de discrétisation des différents paramètres d'entrée et de sortie.

D. Fonction *predict.eph*

La fonction *predict.eph* est la fonction qui estime la valeur la plus probable, pour la médiane et les quantiles 5 et 95, de la loi de probabilité obtenue par l'EPH en n'importe quel point du domaine.

Elle prend en argument un objet de classe *eph* et une matrice *newdata* de coordonnées sur lesquelles effectuer l'estimation.

La fonction n'effectue pas d'estimation en des points déjà connus de l'espace. Si des points sont communs aux matrices *newdata* et *x* la valeur de sortie sera la valeur exacte de la mesure.

IV. Exemple d'utilisation – Fonction de Branin-Hoo

La SCM a confronté les résultats obtenus par Krigeage et par l'EPH dans différentes configurations sur la fonction Branin-Hoo.

Nous avons mis en place trois outils de mesure pour procéder à ces comparaisons, un test et deux mesures :

Soit N le nombre de points reconstruits.

Soit X le vecteur de N termes des points estimés par EPH.

Soit Y le vecteur de N termes des points estimés par Krigeage.

Soit R le vecteur de N termes des valeurs réelles de la fonction.

On note respectivement X_n , Y_n et R_n le n ème coefficient des vecteurs X , Y et R .

Test 1 :

Nombre de points pour lesquels la valeur réelle est en dehors de les bornes à 95% déterminés par les quantiles 2,5 et 97,5 de la loi de probabilité. C'est un test très grossier.

Deux mesures :

Dist_1 :

Elle représente l'écart entre la valeur moyenne (valeur assignée par la reconstruction) et la valeur réelle :

- $Dist_1 = \frac{1}{N} \sum_{n=1}^N |X_n - R_n|$ pour l'EPH.
- $Dist_1 = \frac{1}{N} \sum_{n=1}^N |Y_n - R_n|$ pour le Krigeage.

EQ_2 :

Elle représente la moyenne des erreurs quadratiques, aussi appelé moyenne des risques quadratiques :

- $EQ_2 = \frac{1}{N} \sum_{n=1}^N (X_n - R_n)^2$ pour l'EPH.

$$- EQ_2 = \frac{1}{N} \sum_{n=1}^N (Y_n - R_n)^2 \text{ pour le Krigage.}$$

A. Estimation au point de coordonnées (0.5 ; 0.5) dans R^2 :

L'exemple ci-après cherche à reconstruire la valeur de la fonction de Branin-Hoo au point de coordonnées (0.5,0.5) à partir de 400 points de mesures choisis régulièrement dans le carré $[0,1] \times [0,1]$.

3. Résultats avec Krigage :

On a testé dans un premier temps la prédiction par Krigage, les résultats sont les suivants :

```
- $lower95 : -5.549907
- $mean : 32.79877
- $upper95 : 71.14745
```

Les valeur lower95 et upper95 correspondent respectivement aux bornes inférieure et supérieure de l'intervalle à 95%, soient les quantiles 2.5% et 97,5% de la loi de probabilité.

4. Résultats avec EPH :

Nous avons appliqué l'EPH au même jeu de mesures en souhaitant prédire le même point inconnu. Les résultats sont les suivants :

```
- $lower95 : -9.6
- $mean : 27
- $upper95 : 98.4
- $esperance :35.31372
```

Les valeur lower95 et upper95 correspondent respectivement aux bornes inférieure et supérieure de l'intervalle à 95%, soient les quantiles 2.5% et 97,5% de la loi de probabilité. Nous avons conservé le formalisme du package {DiceKrigin}.

5. Valeur réelle :

Nous avons utilisé la fonction de Branin implémentée en R pour calculer la valeur exacte que nous cherchions précédemment à prédire.

```
x=c(0.5,0.5)
branin(x)
[1] 24.27813
```

Les deux méthodes d'estimation passent le test qualitatif.

Nous avons calculé les deux mesures préalablement définies :

	Dist_1	EQ_2
Krigeage	9	73
EPH	11	122

Tableau 4 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple sur un point

On constate que l'EPH est plus éloignée de la valeur attendue. Qui plus est, son intervalle à 95% est également plus large. La distance moyenne par rapport à la valeur réelle est plus faible pour l'estimation par Krigeage en ce point de coordonnées. L'erreur quadratique moyenne est également plus faible pour l'estimation par Krigeage.

B. Estimation surfacique dans R^2 :

Dans chacun des trois exemples, nous reconstruisons 400 points du domaine à partir des points de mesure dont nous disposons : 3, 9 ou 16 points.

1. Reconstruction de 400 points à partir de 16 points de mesure

- Krigeage Ordinaire :

Nous avons reconstruit 400 points choisis de manière régulière dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir de 16 points de mesure choisis eux aussi régulièrement dans $[0,1] \times [0,1]$ en utilisant la méthode de Krigeage.

La figure ci-après représente la surface attendue (les valeurs réelles) et la surface obtenue pour l'estimation par Krigeage :

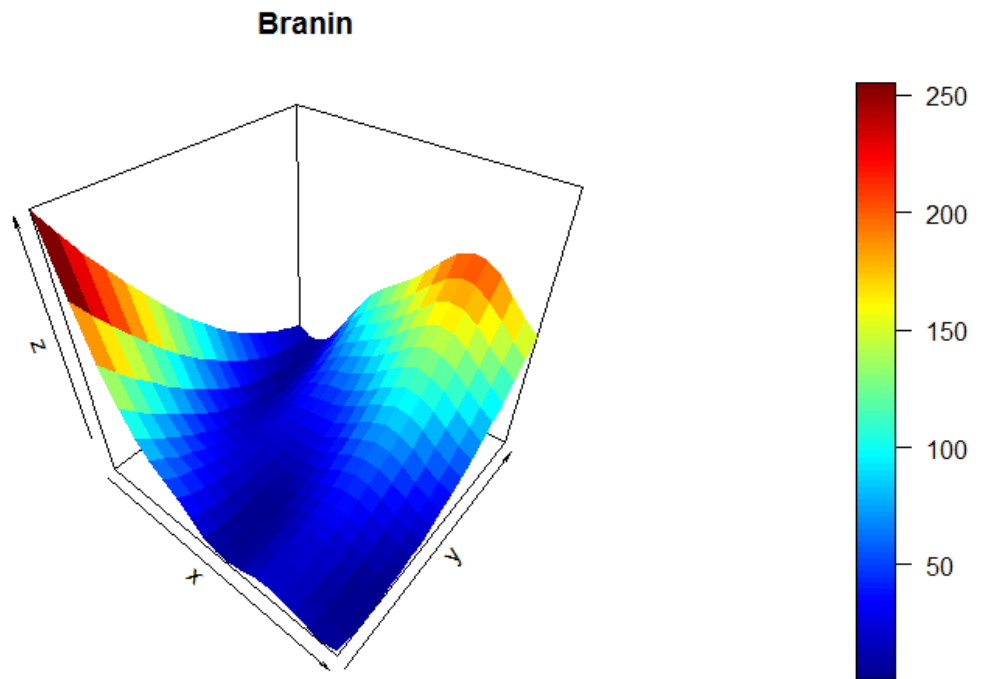


Figure 1 : Surface à reconstruire, fonction de Branin

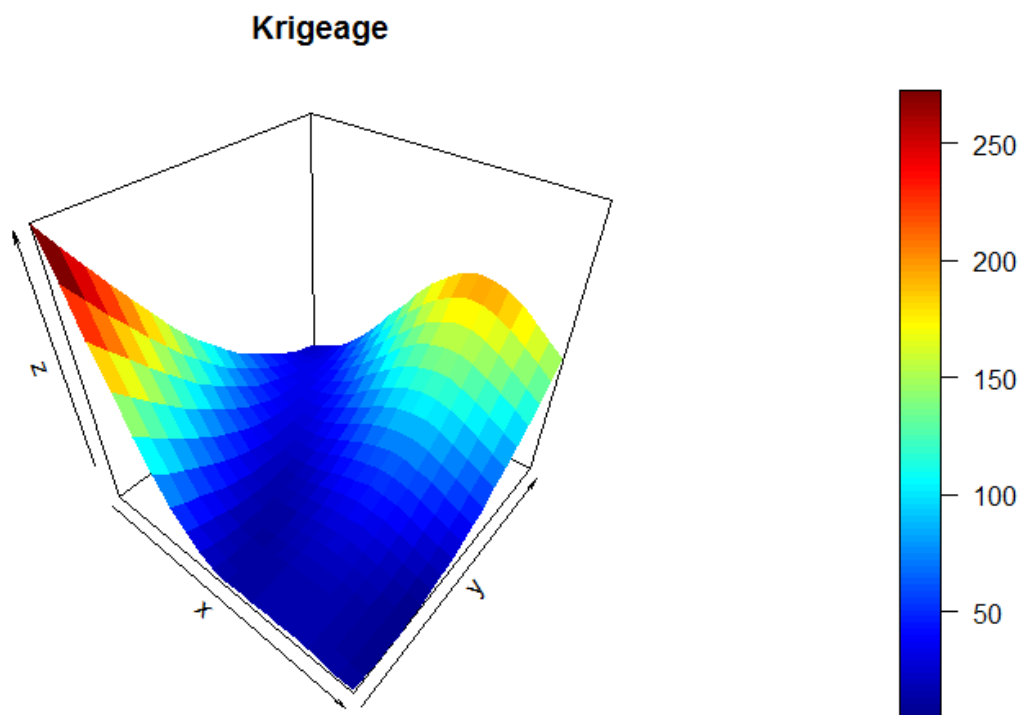


Figure 7 : Krigage ordinaire à 16 points

– Utilisation de l'EPH

De la même façon, nous avons reconstruit les 400 même points que dans le cas du Krigage dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir des 16 points de mesure qui ont servis pour le Krigage en utilisant cette fois la méthode EPH.

La figure ci-après représente la surface obtenue pour l'estimation par EPH :

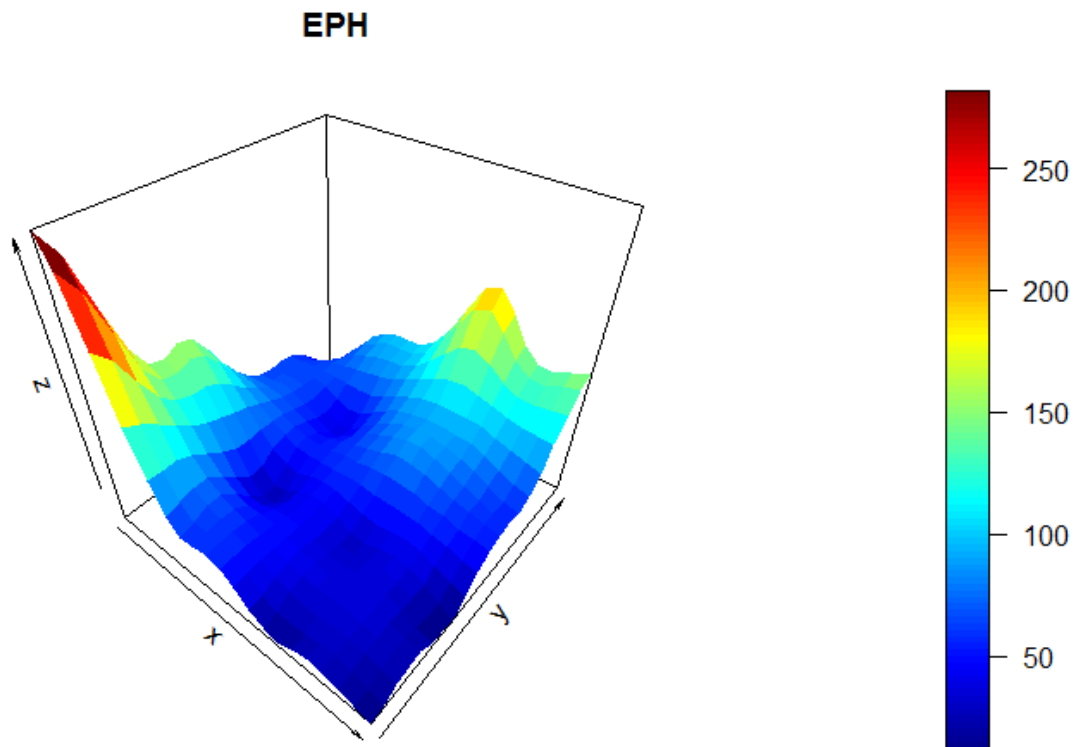


Figure 8 : EPH à 16 points

Après calcul des distances entre les estimations et les valeurs réelles nous obtenons les résultats suivants :

	Dist_1	EQ_2
Krigeage	11	605
EPH	24	4 778

Tableau 9 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 16 points

Aucun des 400 points reconstruits n'est en dehors des bornes inférieures et supérieures de l'intervalle à 95% pour le cas du Krigeage à partir de 16 points de mesure. Il en est de même pour l'estimation par EPH. Le test qualitatif est réussi pour les deux méthodes.

La distance moyenne de la valeur réelle à l'espérance sur ces 400 points est de 11 pour le Krigeage. Cette distance est de 24 pour l'estimation par EPH, ce qui est plus du double de la reconstruction par Krigeage.

L'erreur quadratique moyenne est de 605 pour le Krigeage. Cette erreur est de 4 778 pour l'estimation par EPH, ce qui est nettement moins bon.

2. Reconstruction de 400 points à partir de 9 points de mesure

– Krigeage Ordinaire :

Nous avons reconstruit 400 points choisis de manière régulière dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir de 9 points de mesure choisis eux aussi régulièrement dans $[0,1] \times [0,1]$ en utilisant la méthode de Krigeage.

La figure ci-après représente la surface attendue (les valeurs réelles) et la surface obtenue pour l'estimation par Krigeage :

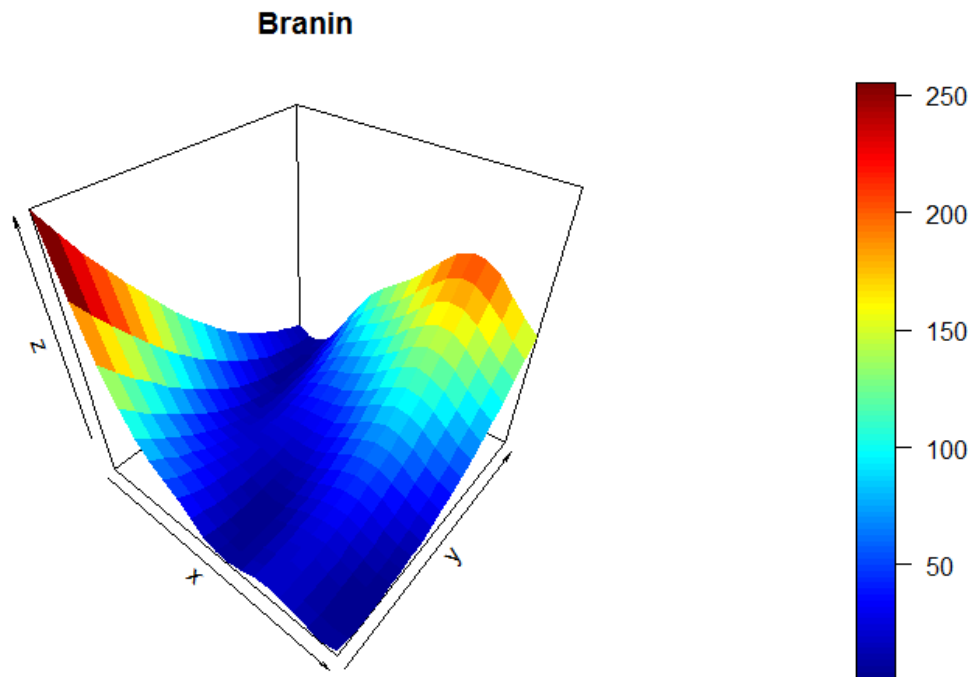


Figure 10 : Surface à reconstruire, fonction de Branin

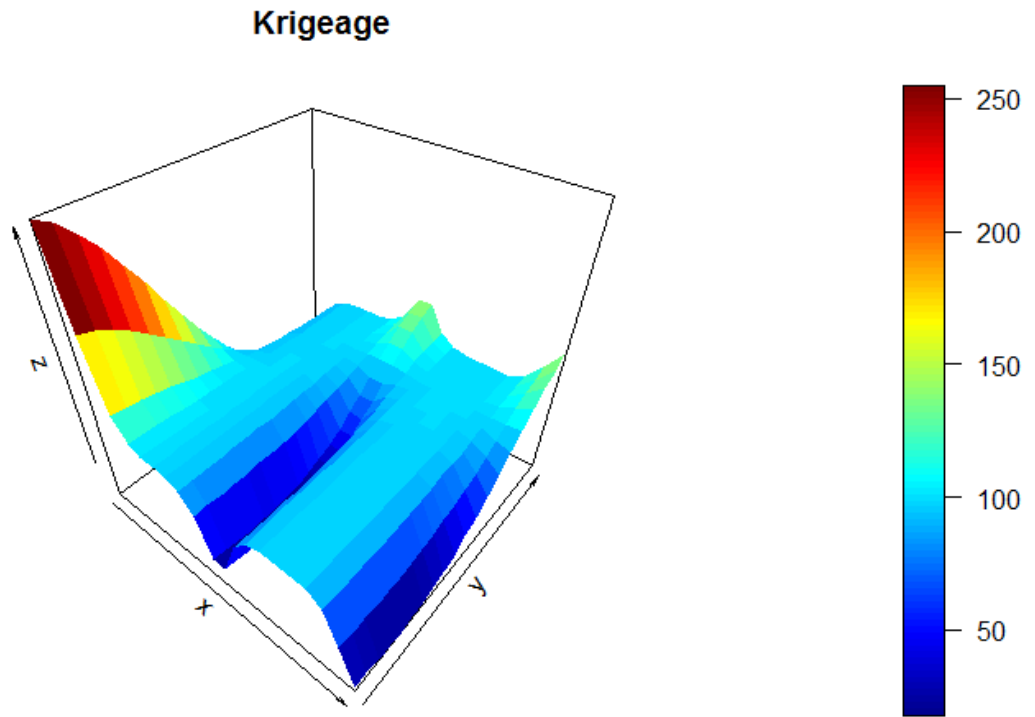


Figure 11 : Krigage ordinaire à 9 points

– Utilisation de l'EPH :

De la même façon, nous avons reconstruit les 400 même points que dans le cas du Krigage dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir des 9 points de mesure qui ont servi pour le Krigage en utilisant cette fois la méthode EPH.

La figure ci-après représente la surface obtenue pour l'estimation par EPH :

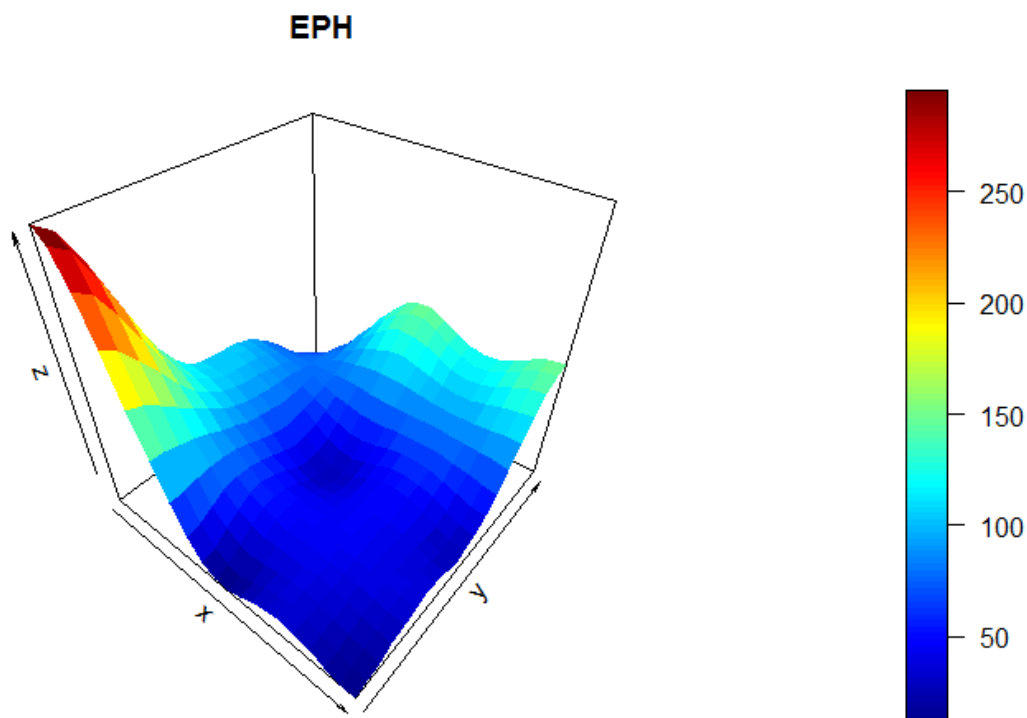


Figure 12 : EPH à 9 points

Après calcul des distances entre les estimations et les valeurs réelles, nous obtenons les résultats suivants :

	Dist_1	EQ_2
Krigeage	45	36 129
EPH	32	4 880

Tableau 13 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 9 points

On possède ici moins d'informations que dans le cas avec 16 points de mesure, mais aucun des 400 points reconstruits n'est en dehors des bornes inférieures et supérieures de l'intervalle à 95% pour le cas du Krigeage à partir de 9 points de mesure. Il en est de même pour l'estimation par EPH. Le test qualitatif est réussi pour les deux méthodes.

La distance moyenne de la valeur réelle à l'espérance sur ces 400 points est de 45 pour le Krigeage et de 32 pour l'estimation par EPH.

L'erreur quadratique moyenne est de 36 129 pour le Krigeage, et de 4 880 pour l'estimation par EPH, ce qui est nettement meilleur. On constate qu'à 9 points l'erreur quadratique est beaucoup plus forte qu'à 16 points.

L'estimation est de moins bonne qualité qu'avec 16 points de mesure. Ceci est dû à la perte d'informations.

L'estimation par EPH est ici de meilleure qualité que l'estimation par Krigeage à partir de 9 points.

3. Reconstruction de 400 points à partir de 3 points de mesure

– Krigeage Ordinaire :

Nous avons reconstruit 400 points choisis de manière régulière dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir de 3 points de mesure choisis eux aussi régulièrement dans $[0,1] \times [0,1]$ en utilisant la méthode de Krigeage.

La figure ci-après représente la surface attendue (les valeurs réelles) et la surface obtenue pour l'estimation par Krigeage :

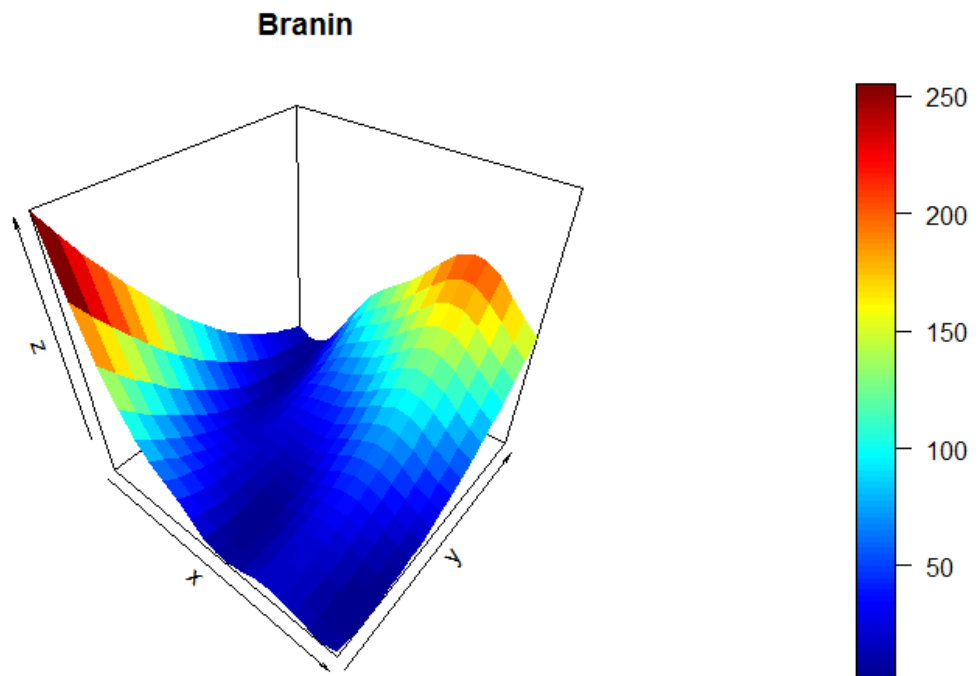


Figure 14 : Surface à reconstruire, fonction de Branin

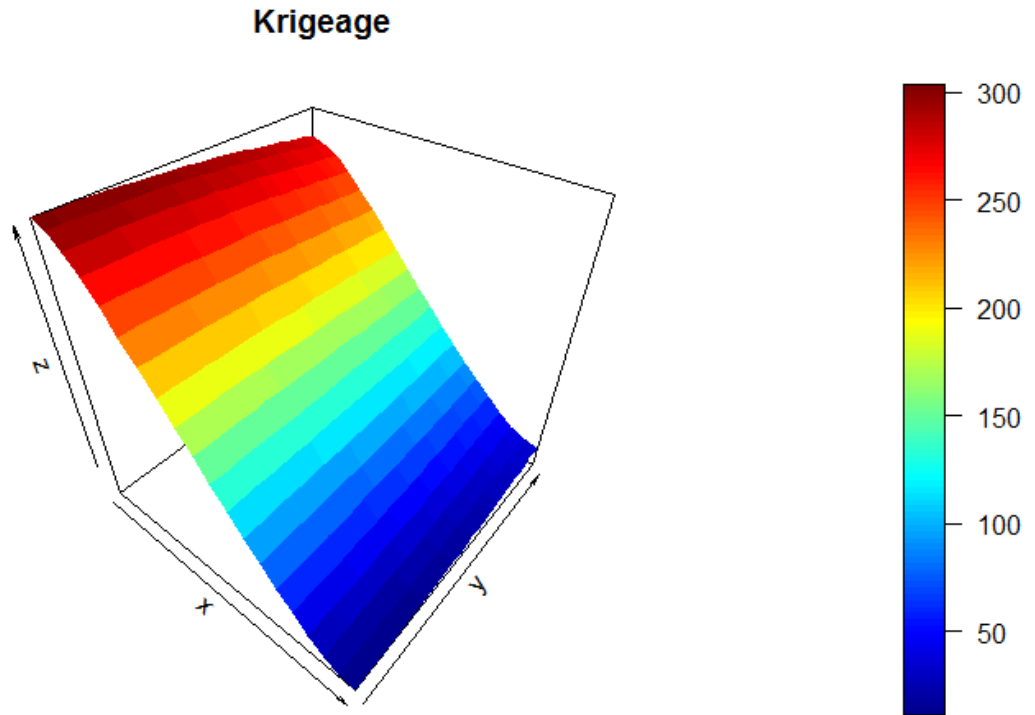


Figure 15 : Krigage à 3 points

– Utilisation de l'EPH :

De la même façon, nous avons reconstruit les 400 mêmes points que dans le cas du Krigage dans le carré $[0,1] \times [0,1]$ de la fonction de Branin à partir des 3 points de mesure qui ont servi pour le Krigage en utilisant cette fois la méthode EPH.

La figure ci-après représente la surface obtenue pour l'estimation par EPH :

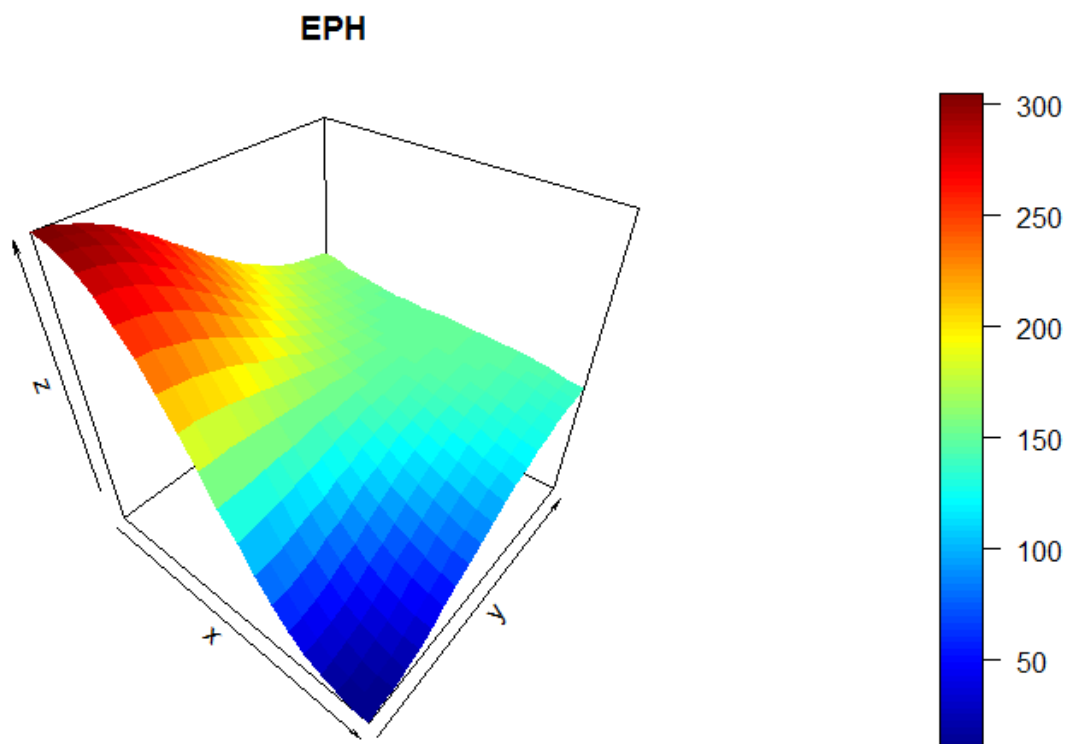


Figure 16 : EPH à 3 points

Après calcul des distances entre les estimations et les valeurs réelles, nous obtenons les résultats suivants :

	Dist_1	EQ_2
Krigeage	119	32 296
EPH	100	3 998

Tableau 17 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 3 points

Les estimations par EPH et par Krigeage à partir de 3 points de mesure réussissent le test qualitatif.

La distance moyenne de la valeur réelle à l'espérance sur ces 400 points est de 119 pour le Krigeage, et de 100 pour l'estimation par EPH.

Avec très peu de points de mesures, la reconstruction est très loin de la réalité, plus de 10 fois en moyenne que dans le cas de la reconstruction à partir de 16 points de mesure.

Les distances moyennes de la valeur réelle à l'espérance sur ces 400 points sont nettement supérieures aux distances moyennes avec 16 ou 9 points de mesure. La reconstruction est de moins bonne qualité à cause de la perte d'information.

L'erreur quadratique moyenne est de 32 296 pour le Krigage. Cette erreur est de 3 998 pour l'estimation par EPH, ce qui est nettement meilleur. On constate qu'à 3 points l'erreur quadratique est sensiblement identique à ce qu'elle était pour 9 points, ce qui n'est pas le cas de la distance moyenne à la valeur réelle.

L'estimation par EPH est en ce cas de meilleure qualité que l'estimation par Krigage.

C. Reconstructions à partir de points choisis de manière aléatoire

Nous avons reconstruit les mêmes 400 points que dans les exemples précédents, mais cette fois à partir de jeux de mesures choisis aléatoirement. Sur 5 tests à partir de 16 points de mesures les résultats sont les suivants :

Jeu de mesures n°1 :

	Dist_1	EQ_2
Krigage	13	2 469
EPH	30	5 714

Tableau 18 : Comparaison Krigage et EPH par rapport à la valeur réelle, Branin à 16 points de mesure choisis aléatoirement

Jeu de mesures n°2 :

	Dist_1	EQ_2
Krigage	25	16 996
EPH	27	5 624

Tableau 19 : Comparaison Krigage et EPH par rapport à la valeur réelle, Branin à 16 points de mesure choisis aléatoirement

Jeu de mesures n°3 :

	Dist_1	EQ_2
Krigeage	19	6 794
EPH	34	4 426

Tableau 20 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 16 points de mesure choisis aléatoirement

Jeu de mesures n°4 :

	Dist_1	EQ_2
Krigeage	6	788
EPH	25	5 855

Tableau 21 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 16 points de mesure choisis aléatoirement

Jeu de mesures n°5 :

	Dist_1	EQ_2
Krigeage	6	701
EPH	24	5 177

Tableau 22 : Comparaison Krigeage et EPH par rapport à la valeur réelle, Branin à 16 points de mesure choisis aléatoirement

Le Krigeage est plus performant en termes d'écarts moyens à la valeur réelle ; cette mesure est toujours plus petite pour le Krigeage, avec 16 points de mesure.

Cependant l'EPH est plus stable en ce qui concerne l'erreur quadratique moyenne.

Nous avons également fait des tests sur des jeux de données complètement aléatoires, les 16 points de mesures choisis aléatoirement ainsi que les 400 points à reconstituer.

Jeu de données n°1 :

	Dist_1	EQ_2
Krigeage	19	39 018
EPH	32	6 386

Tableau 23 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple tout aléatoire

Jeu de données n°2 :

	Dist_1	EQ_2
Krigeage	53	5 129
EPH	35	4 874

Tableau 24 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple tout aléatoire

Jeu de données n°3 :

	Dist_1	EQ_2
Krigeage	22	697
EPH	7	4 553

Tableau 25 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple tout aléatoire

Jeu de données n°4 :

	Dist_1	EQ_2
Krigeage	7	499
EPH	20	3 715

Tableau 26 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple tout aléatoire

On constate que, selon les cas, l'une ou l'autre méthode donne de meilleurs résultats.

V. Limites

L'étude n'a été effectuée que sur la fonction de Branin-Hoo qui n'a pas nécessairement le comportement d'un jeu de mesures issues d'expériences.

Nous avons également appliqué la méthode à la fonction définie par :

$$f(x_1, x_2) = \sin(5.5x_1)\cos(5x_2) + x_1^2 + 1$$

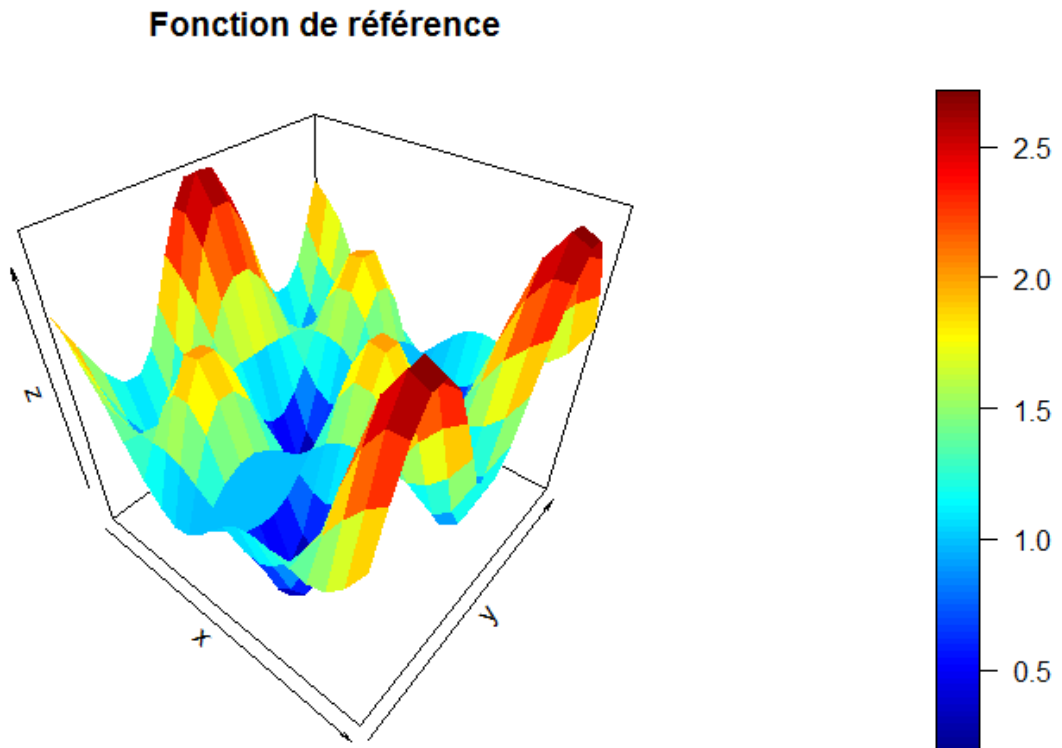


Figure 27 : représentation graphique d'une fonction test

Jeu de données n°1 :

	Dist_1	EQ_2
Krigeage	0.21	0.67
EPH	0.32	1.23

Tableau 28 : Comparaison Krigeage et EPH par rapport à la valeur réelle, exemple tout aléatoire sur fonction de référence

Les résultats sont similaires à ceux que nous avons obtenus pour la fonction de Branin pour les jeux de données aléatoires testés.

VI. Recherche des points à explorer

A. Optimiseur Global EGO utilisé par l'IRSN

L'optimiseur global EGO se base sur l'agrégation du point de données qui maximise l'Expected Improvement à chaque itération pour trouver les maxima et minima locaux de la fonction à estimer.

EGO (Efficient Global Optimization) est un algorithme d'optimisation basé sur le Krigeage, et plus particulièrement sur le critère d'amélioration espérée. Partant d'un plan d'expériences initial X , EGO forme une séquence de points en alternant à chaque itération la visite du maximiseur courant de l'amélioration espérée (disons le premier visité s'il y en a plusieurs) et l'actualisation du modèle de Krigeage, y compris via ré-estimation des paramètres de covariance :

Algorithm 1 L'algorithme EGO

```
1: function EGO( $X, Y, n$ )
2:   for  $i \leftarrow 1, n$  do
3:      $\psi^* = \operatorname{argmax}_{\psi \in \Psi} L(\psi; Y(X) = Y)$  ▷ Estimation de  $\psi$  par MV
4:      $x^* = \operatorname{argmax}_{x \in D} \{EI_{\psi^*}(x)\}$  ▷ Maximisation de l'amélioration espérée
5:      $X = X \cup \{x^*\}$  and  $Y = Y \cup \{y(x^*)\}$  ▷ Actualisation du plan d'exp.
6:   end for
7: end function
```

L'algorithme boucle jusqu'à ce que la croissance de la fonction objectif atteigne un epsilon choisi par l'utilisateur suffisamment petit, et dépendant des précédentes évaluations de la fonction objectif.

Il n'est pas possible de donner une expression explicite de l'Expected Improvement dans le cas de l'EPH. Nous allons donc procéder d'une façon différente pour localiser les points sur lesquels effectuer les prochaines mesures.

B. Recherche des zones dangereuses, une méthode développée par la SCM

A partir d'un petit nombre de mesures, l'EPH permet de trouver des zones dites dangereuses. Une fois ces zones caractérisées, il faut concentrer les mesures suivantes à l'intérieur de ces zones. En d'autres termes, l'EPH permet de caractériser l'espace pour concentrer les mesures là où le réel besoin se situe.

Les zones dangereuses sont caractérisées par une propriété mathématique (plus grande proximité des extrema que des autres points de mesures), qui est analogue aux diagrammes de Voronoï, mais ces diagrammes ne peuvent être construits explicitement en trop grande dimension puisque la complexité est en $n^{d/2}$.

1. Recherche des zones dangereuses

Pour trouver le point situé à distance maximale d'un extremum donné, dans un diagramme, on utilise une construction probabiliste : On choisit une direction aléatoire et on mesure la distance pour atteindre les limites de l'hypercube.

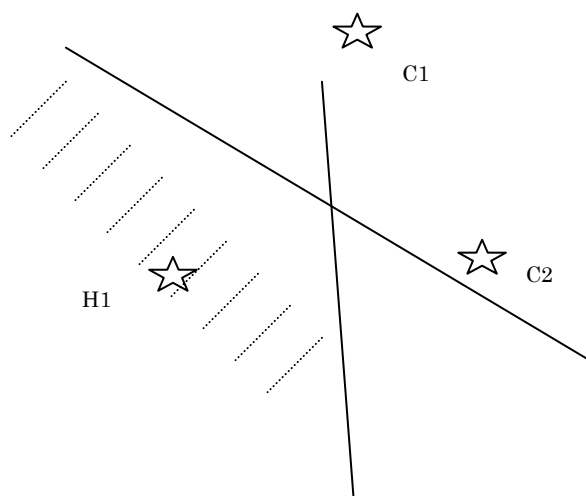
Pour commencer, on choisit un seuil T_0 et on veut trouver les zones de l'hypercube dans lesquelles la sortie a une forte probabilité d'être au dessus de ce seuil. Cela se produit logiquement dans le voisinage des points où la mesure observée est forte.

On appelle H l'ensemble des points dont la mesure est au dessus du seuil préalablement défini. De la même façon, on appelle C l'ensemble des points dont la mesure est en dessous du seuil. Soit H_1 le premier point de H . Soit C_1 le premier point de C .

On considère l'hyperplan situé à égale distance de H_1 et C_1 . D'un côté de cet hyperplan, la distance à C_1 est plus petite que la distance à H_1 , de l'autre côté, c'est le contraire. On a donc, d'un côté de l'hyperplan, une influence plus forte provenant de C_1 et de l'autre, une influence de plus forte provenant de H_1 .

Par conséquent, dans le côté qui contient C_1 , la valeur de la mesure ne peut pas être au dessus du seuil défini si l'on ne considère que l'influence des points H_1 et C_1 .

Comparons maintenant H_1 et C_2 . On construit un hyperplan similaire au précédent, à égale distance des deux points. L'influence de C_2 est plus forte d'un côté que de l'autre.



L'intersection des deux demi-espaces est représentée par la zone grisée sur le figure ci-avant : dans cette zone, seul H_1 a une forte influence, dans les autres parties de l'espace, l'influence de H_1 est moins forte que celle de C_1 ou C_2 . Appelons cette zone de l'espace $E(H_1; C_1, C_2)$.

On répète la même opération pour chacun des points de C ; on obtient une portion de plus en plus petite de l'espace $E(H_1; C_1, C_2, \dots, C_k)$ qui est un espace convexe, non vide puisqu'il contient au moins la boule de centre H_1 et de rayon : $r = \frac{1}{2} \min\{dist(H_1, C_i); i = 1 \dots k\}$

Analytiquement $E(H_1; C_1, C_2, \dots, C_k)$ est l'ensemble des points dans lequel H_1 a plus d'influence que n'importe quel point sous le seuil défini.

On procède de la même façon pour chacun des points au dessus du seuil et on obtient la zone dangereuse définie comme l'union de chacun des convexes associés à chaque H_i :

$$DZ = \bigcup_{j=1}^{length(H)} E(H_j; C_1, C_2, \dots, C_k)$$

Il s'agit maintenant de trouver le point le plus éloigné de H_1 dans cette zone.

2. Recherche du point le plus éloigné de H_1

Construction de l'hypercube

Considérons la valeur maximale des mesures observées : On appelle ce point H_0 .

On fixe le seuil T_0 et on souhaite trouver le point où la probabilité d'être au dessus de ce seuil est la plus forte.

Il y a NT points où la mesure est plus petite que le seuil fixé. On les note C_n , $n = 1, \dots, NT$.

Pour chaque point C_n , on considère l'hyperplan E_n équidistant de C_n et H_0 et on considère le demi-espace F_n déterminé par cet hyperplan, contenant H_0 : C'est l'ensemble des points dans $\mathbb{R}^K$ qui sont plus près de H_0 que de C_n . On a donc NT demi-espaces comme celui-ci.

On considère également les demi-espaces $x_j \geq 0, j = 1, \dots, K$ et les demi-espaces $x_j \leq 1, j = 1, \dots, K$. On a donc un total de $NT + 2 * K$ demi-espaces.

L'intersection A de ces $NT + 2 * K$ demi-espaces est non vide, puisqu'elle contient le point H_0 . C'est un simplexe et on veut trouver le point qui est le plus éloigné de H_0

Recherche du point extremum de l'hypercube

Notre problème se définit comme suit :

Trouver le point X dans l'espace $[0,1]^K$ tel que :

$$d(X, H_0) \text{ est maximal}$$

Satisfaisant les contraintes :

$$d(H_0, X) \leq d(C_m, X) \text{ pour tout } m = 1, \dots, M$$

Ici, d représente la distance euclidienne usuelle. Mais nous travaillerons toujours avec le carré de cette distance, d^2 , donc c'est un problème convexe, avec des contraintes convexes. Puisque le point X est dans $[0,1]^K$, toutes ses coordonnées doivent satisfaire :

$$0 \leq x_k \leq 1,$$

La première étape consiste à générer $MT = 5000$ directions aléatoires, dans $\mathbb{R}^K$, avec une loi uniforme. On souhaite une loi uniforme pour que toutes les directions soient considérées de la même façon.

Pour cela, on génère 5000 vecteurs aléatoires suivant une loi normale de médiane 0 et de variance 1 et on les remplace par :

$$\left(\frac{X_1}{(X_1^2 + \dots + X_K^2)^{1/2}}, \dots, \frac{X_K}{(X_1^2 + \dots + X_K^2)^{1/2}} \right)$$

Les directions aléatoires sont alors définies comme les vecteurs $H_0 + V_m$, où V_m est le m -ième vecteur aléatoire créé ci-avant.

On note $H(k), k = 1, \dots, K$ les coordonnées du point H_0 et $C(n, k), n = 1, \dots, NT, k = 1, \dots, K$ les coordonnées des points dont la mesure est plus petite que le seuil fixé.

On calcule les milieux M_n entre H_0 et chaque point C_n :

$$M(n, k) = \frac{1}{2}(H(k) + C(n, k))$$

Le vecteur $\overrightarrow{HC_n}$ de H_0 à C_n :

$$HC(n, k) = C(n, k) - H(k)$$

L'hyperplan E_n est orthogonal au vecteur $\overrightarrow{HC_n}$, donc son équation est de la forme :

$$\sum_{k=1}^K HC(n,k)x_k = D_n$$

pour une valeur quelconque D_n .

Puisque cet hyperplan contient le point milieu M_n , on obtient :

$$\sum_{k=1}^K HC(n,k)M(n,k) = D_n$$

Donc l'équation de l'hyperplan est :

$$\sum_{k=1}^K HC(n,k)x_k = \sum_{k=1}^K HC(n,k)M(n,k)$$

Soit $H_0 + \lambda * V_m$ une des directions définies plus tôt, de coordonnées : $H(k) + \lambda * V(m,k)$.

Son intersection avec l'hyperplan E_n est obtenue pour :

$$\sum_{k=1}^K HC(n,k)(H(k) + \lambda V(m,k) - M(n,k)) = 0$$

C'est à dire :

$$\lambda = \frac{\sum_{k=1}^K HC(n,k)(M(n,k) - H(k))}{\sum_{k=1}^K HC(n,k)V(m,k)}$$

Puisque V_m est normalisé, la distance entre H_0 et l'intersection avec l'hyperplan est $|\lambda|$.

L'intersection de la ligne avec l'hyperplan $x_k = 0$ est :

$$\lambda = -\frac{H(k)}{V(m,k)}$$

Et l'intersection avec l'hyperplan $x_k = 1$ est :

$$\lambda = \frac{1 - H(k)}{V(m,k)}$$

Chaque vecteur V_m génère une ligne et deux directions, représentées par $\pm V_m$. Pour chacune de ces deux directions, on considère toutes les intersections avec l'hyperplan et on conserve la plus proche de H_0 .

On calcule λ pour chaque hyperplan défini par les points C_n et les hyperplans $x_j \geq 0$ et $x_j \leq 1$. On conserve le vecteur correspondant à la plus petite valeur absolue de λ .

On répète l'opération pour chacune des directions et on conserve la direction dont la valeur absolue de λ est la plus élevée.

On obtient ainsi pour M_1 le point le plus éloigné de H_0 appartenant à l'hypercube. Soit d la distance entre ces deux points :

$$d = \sqrt{\sum (M_1(k) - H(k))^2}$$

Une vérification est faite, dans le but de s'assurer que le point M_1 est situé du bon côté de tous les hyperplans. Il doit satisfaire :

$$\text{dist}(H_0, M_1) \leq \text{dist}(M_1, C_n) \text{ pour tout } n$$

Le point M_1 doit être plus proche de l'extremum local que de chacun des points sous le seuil défini.

$$0 \leq M_1(k) \leq 1 \text{ pour tout } k$$

3. Calcul de la probabilité d'être au delà du seuil au point M_1

Le point M_1 est influencé uniquement par le point H_0 .

La loi de probabilité discrète au point M_1 est donnée par la formule :

$$p_j(M_1) = \frac{c\tau}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_j - \theta)^2}{2\sigma^2}\right)$$

où σ est donné par la formule :

$$\sigma = \frac{\tau e^{\lambda d}}{\sqrt{2\pi e}}$$

où τ représente le pas de discrétisation, et $d = d(M_1, H_0)$.

D. Etape suivante de la méthode

Le point M_1 n'est pas nécessairement le point final de notre recherche : il est le point le plus éloigné de H_0 parmi les MT directions aléatoires explorées. Il est donc possible de trouver un point M_2 proche de M_1 dans l'hypercube des solutions possibles vérifiant :

$$dist(H_0, M_1) \leq dist(H_0, M_2) \leq dist(M_2, C_n) \text{ pour tout } n$$

Pour le trouver, il faut procéder de façon analogue, c'est à dire par tirage aléatoire.

Par construction, le point M_1 est sur un bord de notre simplexe. Il doit être sur l'un des hyperplans limitant. Le point se situe soit sur un des axes limitant l'espace des solutions : les hyperplans $x_j \geq 0$ et $x_j \leq 1$, soit il est à égale distance entre H_0 et un des points C_n . Cette propriété doit se conserver: On souhaite rester dans l'hyperplan dans lequel on se trouve.

1. Tirage aléatoire dans un hyperplan

Le précédent tirage aléatoire est effectué dans l'espace entier. Il faut maintenant faire un tirage aléatoire dans un hyperplan donné.

Il est possible de construire une base orthonormale telle que $x_1 = 0$ soit l'équation de l'hyperplan dans cette base.

Il est également possible de générer un échantillon dans l'espace entier et de le projeter sur l'hyperplan voulu : la projection orthogonale de la loi uniforme sur l'espace entier et une loi uniforme sur l'hyperplan.

Dans les cas où le point M_1 est situé sur une des frontières de l'espace $[0,1]^K$, il suffit de générer MT vecteurs aléatoires de loi uniforme sur la sphère de dimension $K - 1$. On insère ensuite une K-ième colonne nulle à l'endroit où la valeur du paramètre vaut 0 ou 1 pour M_1 .

On conserve les points milieux, les vecteurs associés et les hyperplans E_n .

Les directions aléatoires sont maintenant définies comme $M_1 + \lambda V_m$. Leur intersection avec les hyperplans est obtenue pour :

$$\sum_{k=1}^K HC(n, k) (M_1(k) + \lambda V(m, k) - M(n, k)) = 0$$

On a donc:

$$\lambda = \frac{\sum_{k=1}^K HC(n, k)(M(n, k) - M_1(k))}{\sum_{k=1}^K HC(n, k)V(m, k)}$$

Puisque V_m est normalisé, la distance entre H_0 et l'intersection avec l'hyperplan est $|\lambda|$.

L'intersection de la ligne avec l'hyperplan $x_k = 0$ est :

$$\lambda = -\frac{M_1(k)}{V(m, k)}$$

Et l'intersection avec l'hyperplan $x_k = 1$ est :

$$\lambda = \frac{1 - M_1(k)}{V(m, k)}$$

On calcule ensuite la position de M_2 .

E. Implémentation

La SCM a implémenté la méthode de recherche des zones dangereuses en langage R sous sa forme la plus générale.

Elle prend en argument un objet de classe eph, le seuil souhaité par l'utilisateur et le type d'extremum recherché : il est possible de rechercher les extrema hauts ou bas de la fonction objectif.

La fonction renvoie une ou plusieurs possibilités de points à explorer, chacun appartenant à une zone dangereuse de l'espace des solutions.

VII. Exemple d'utilisation

A. Recherche d'optima

1. Recherche d'un maximum

On applique cette recherche des zones dangereuses (ou zones d'intérêt) à un exemple où tous les points de mesure (les points bleus sur la figure ci-après) ont été choisis aléatoirement.

La valeur du seuil T_0 a été fixée à 140 dans ce cas précis. Les coordonnées du point H_0 sont : $x_1 = 0.72$ et $x_2 = 0.97$

On obtient le point M_1 suivant (en rouge) correspondants à la zone dangereuse (c'est-à-dire où la valeur peut être élevée), associée au point de mesure haute H_0 :

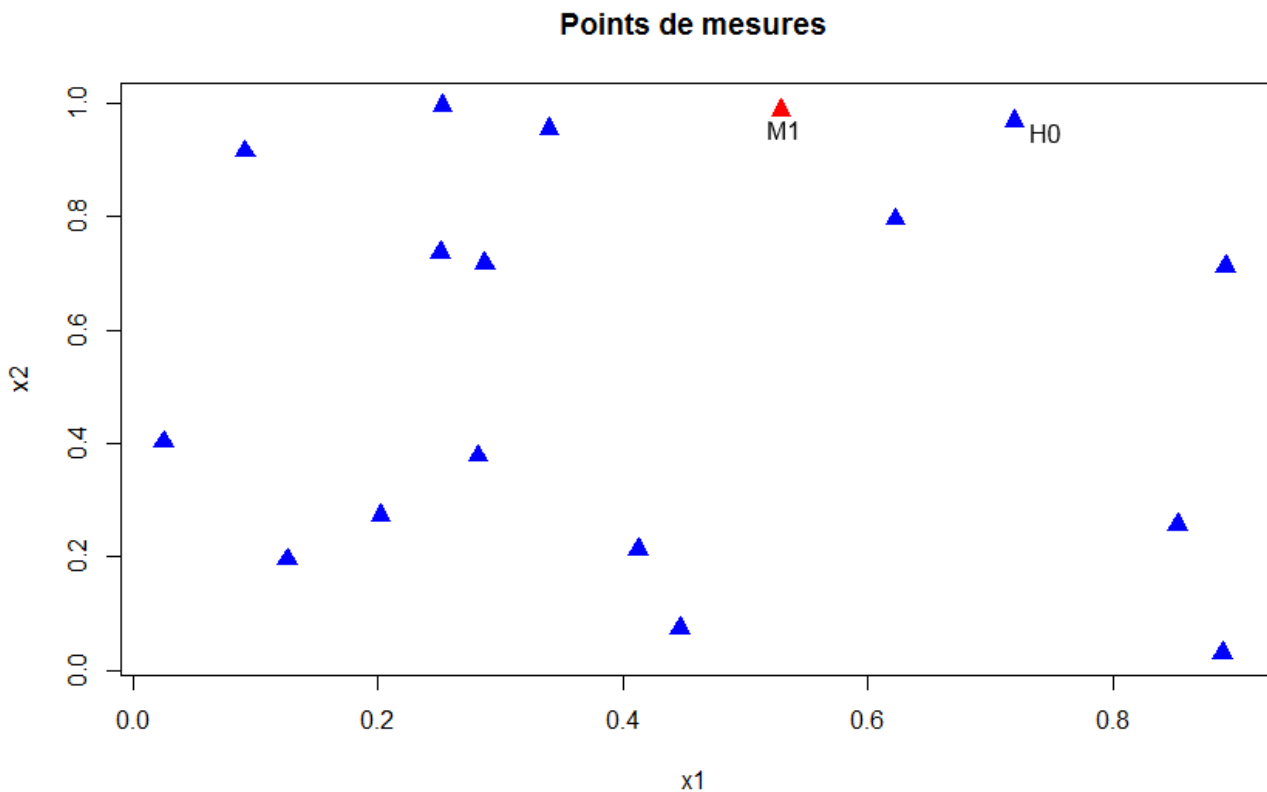


Figure 28 : Points de mesure pour l'EPH seuil haut

Les coordonnées du point M_1 sont :

$x_1 = 0.53$ et $x_2 = 0.99$ à partir du point H_0 de plus grande mesure.

On trace la loi de probabilité au point M_1 :

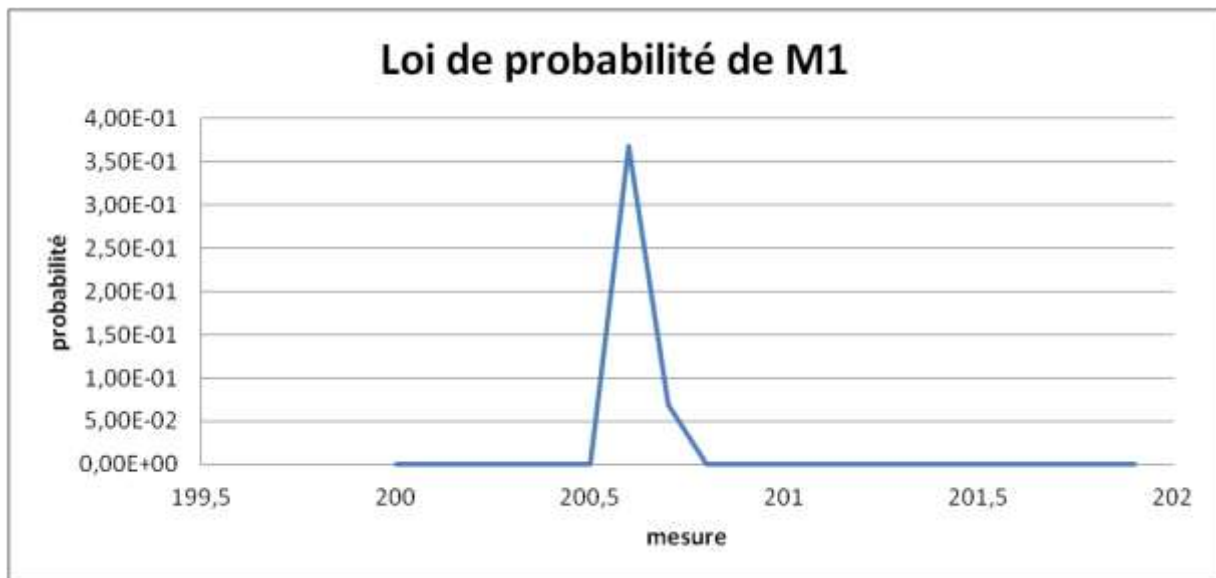


Figure 29 : Loi de probabilité de M1 pour le seuil haut

Cette loi a les caractéristiques suivantes :

$$\sigma = 0.023$$

$$\lambda = -0.193$$

La valeur réelle de la fonction de Branin-Hoo en ce point est : $branin(M_1) = 156.8$.

On a ainsi trouvé un point, où l'on n'avait pas d'information au préalable, qui donne une valeur assez élevée (sans être maximale), mais il pourra servir de base à une nouvelle itération de l'EPH.

2. Recherche d'un minimum

On applique la recherche des zones dangereuses (ou zones d'intérêt) aux mêmes points de mesures aléatoires que précédemment, mais avec cette fois pour objectif de trouver un minimum de la fonction.

La valeur du seuil T_0 a été fixée à 3 dans ce cas. Les coordonnées du point H_0 sont :

$$x_1 = 0.09 \text{ et } x_2 = 0.91.$$

On obtient le point M_1 suivant (en rouge) correspondant à la zone d'intérêt du point de mesure haute H_0 :

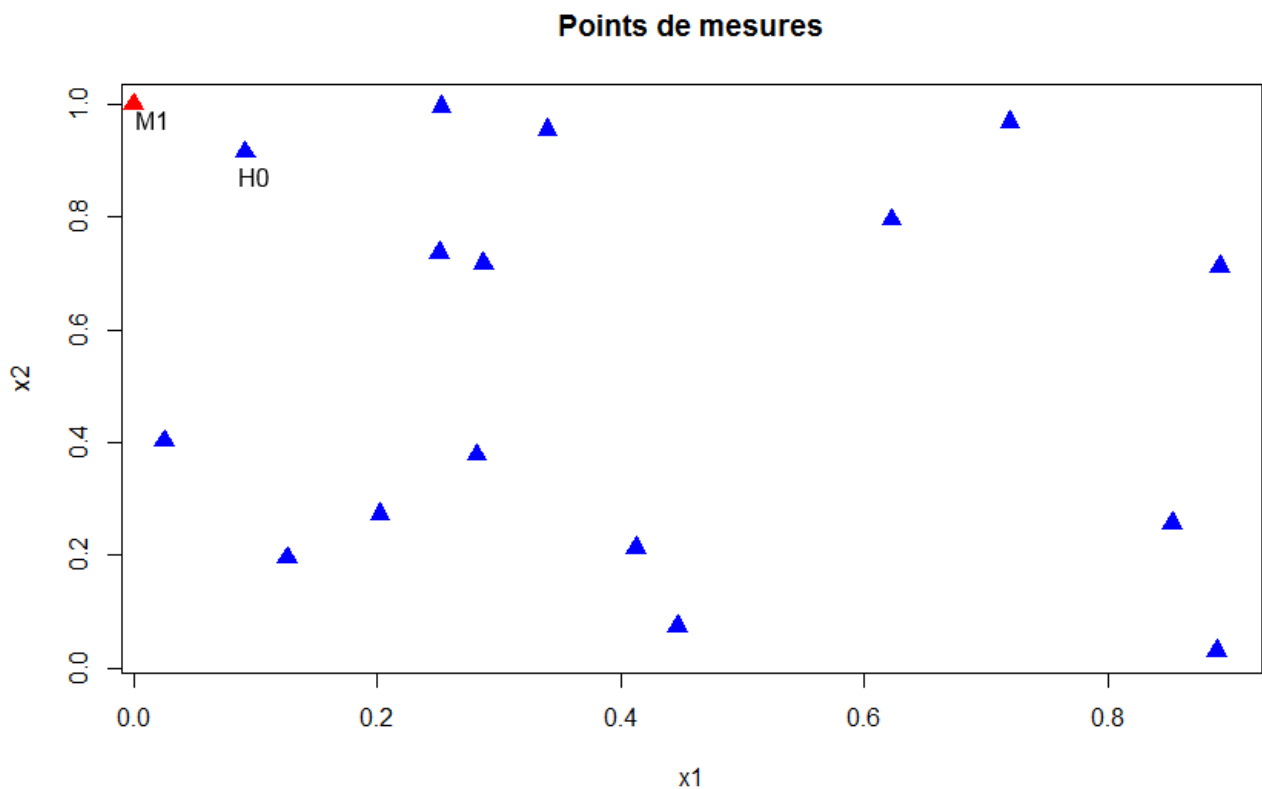


Figure 30 :Points de mesure pour l'EPH seuil bas

Les coordonnées du point M_1 sont :

$x_1 = 0$ et $x_2 = 0.999$ à partir du point H_0 de plus grande mesure.

On trace la loi de probabilité au point M_1 :

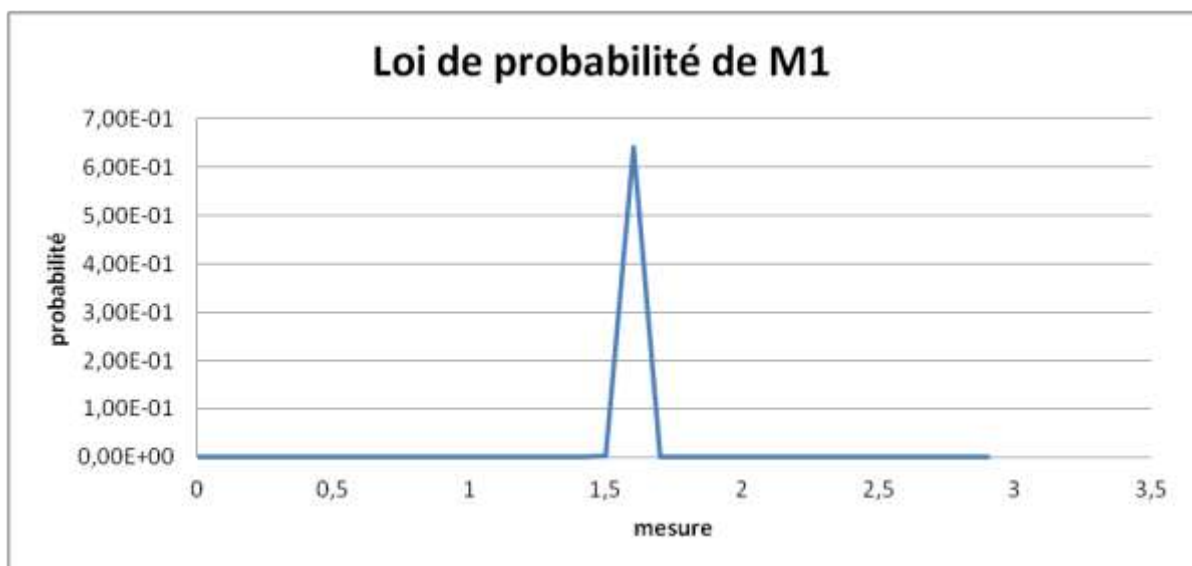


Figure 31 : Loi de probabilité de M1 pour le seuil bas

Cette loi a les caractéristiques suivantes :

$$\sigma = 0.024$$

$$\lambda = -0.125$$

La valeur réelle de la fonction de Branin-Hoo en ce point est : $branin(M_1) = 17.3$.

Le point M_1 est sur l'hyperplan $x_1 = 0$, on utilise donc la recherche du point M_2 décrite ci-avant. On obtient le point de coordonnées $x_1 = 0$ et $x_2 = 1$, la fonction de Branin-Hoo en ce point vaut : $branin(M_2) = 17.2$.

3. Conclusion

L'Algorithme de recherche des zones d'intérêt nous aide à localiser les endroits avec peu d'informations dont les valeurs sont supposées au dessus (ou en dessous) du seuil fixé par l'utilisateur. De nouvelles mesures en ces endroits nous permettent de mieux reconstruire la surface attendue afin d'identifier les optima locaux.

On peut réitérer l'opération pour se rapprocher des optima locaux.

B. EGO n-steps sur le même jeu de données

On applique l'optimiseur global EGO sur le même jeu de données, on obtient le résultat suivant :

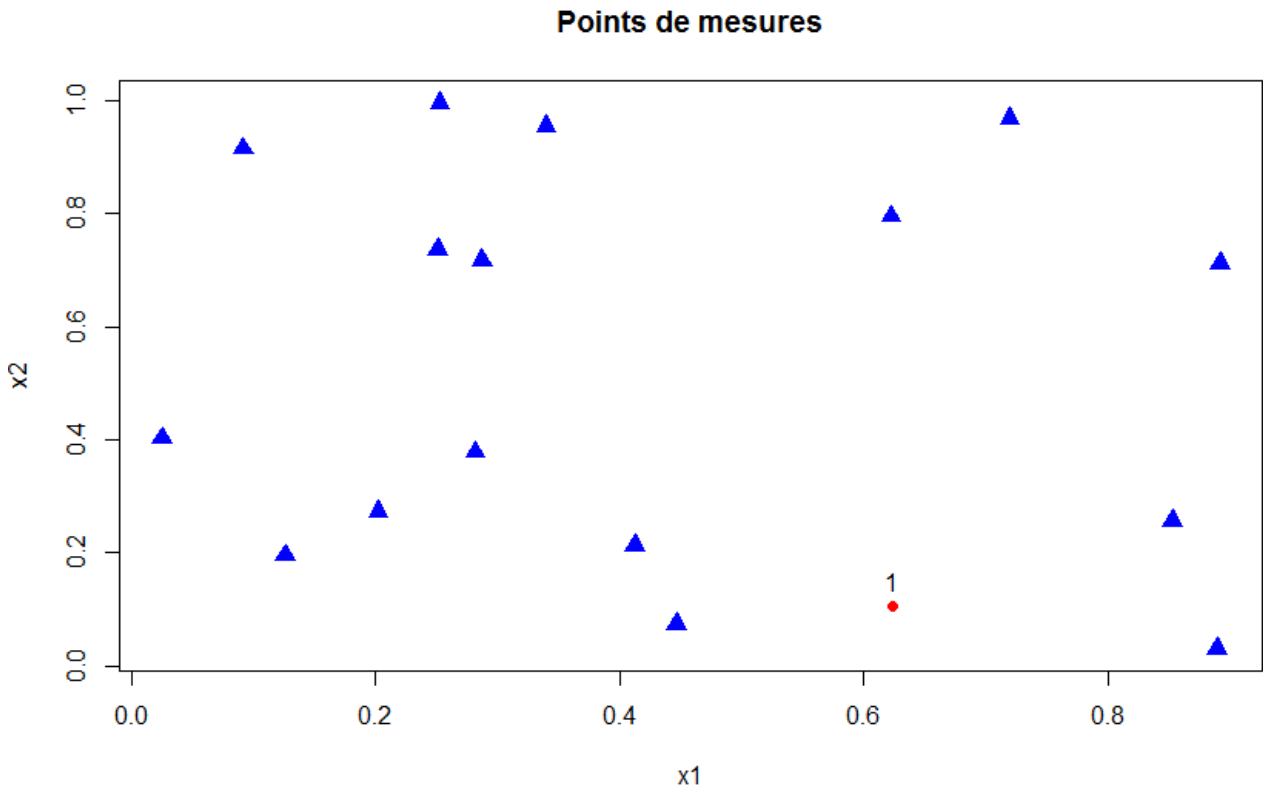


Figure 32 : Points de mesures pour l'optimiseur global EGO

Les coordonnées de M_1 sont : $x_1 = 0.62$ et $x_2 = 0.11$

La fonction de Branin-Hoo en ce point vaut : $branin(M_1) = 6.6$

VIII. Références

[PIT] Olga Zeydina et Bernard Beauzamy : Probabilistic Information Transfer. Ouvrage édité et commercialisé par la Société de Calcul Mathématique SA. ISBN: 978-2-9521458-6-2, ISSN : 1767-1175. Relié, 208 pages, mai 2013.

[ICAPP] Bernard Beauzamy, Hélène Bickert, Olga Zeydina (SCM), Giovanni Bruna (IRSN) : Probabilistic Safety Assessment and Reliability Engineering: Reactor Safety and Incomplete Information; Proceedings of ICAPP 2011 Nice, France, May 2-5, 2011 Paper 11399

IX. Annexe :

A. Estimation au point de coordonnées (0.5 ; 0.5) dans R^2

```
# -----  
# a 2D example  
# -----  
  
library(DiceKriging)  
n.grid <- 20  
x.grid <- y.grid <- seq(0,1,length=n.grid)  
design.grid <- expand.grid(x.grid, y.grid)  
response.grid <- apply(design.grid, 1, branin)  
pas=cbind(c(0.1,0.1,0.1))  
response.grid<-cbind(response.grid)  
mat=matrix(data=c(0,0,0,500,5,5), ncol=2)  
boundariesP=rbind(c(0,500))  
inconnus=rbind(c(0.5,0.5))  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')  
eph1 <- eph(design.grid,response.grid,mat,boundariesP,pas)  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')  
predict.eph(eph1, inconnus)  
  
# -----
```

B. Reconstruction de 400 points à partir de 16 points de mesure

1. Krigeage Ordinaire :

```
# -----  
# A 2D example - Branin-Hoo function  
# -----  
  
# a 16-points factorial design, and the corresponding response  
d <- 2; n <- 16  
design.fact <- expand.grid(x1=seq(0,1,length=4), x2=seq(0,1,length=4))  
y <- apply(design.fact, 1, branin)  
  
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect  
m1 <- km(design=design.fact, response=y)  
  
# graphics  
n.grid <- 20  
x.grid <- y.grid <- seq(0,1,length=n.grid)  
design.grid <- expand.grid(x1=x.grid, x2=y.grid)  
response.grid <- apply(design.grid, 1, branin)  
  
predicted.values <- predict(m1, design.grid, "UK")  
predicted.values.model5 <- predicted.values$mean  
predicted.values.model3 <- predicted.values$lower95  
predicted.values.model4 <- predicted.values$upper95
```



```

par(mfrow=c(2,1))
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")

points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,
      main="Ordinary Kriging")
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")

```

```

# -----

```

2. Hypersurface Probabiliste Experimentale :

```

# -----
# A 2D example - Branin-Hoo function
# -----

setClass("eph",slots=c(NofParam="integer", NofMeas="integer", NofInk2="integer",
x="matrix",          y="matrix",          step="matrix",          boundarie-
sin="matrix",boundariesout="matrix", covariance="covTensorProduct"))
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')
# a 16-points factorial design, and the corresponding response
d <- 2; n <- 16
design.fact <- expand.grid(x1=seq(0,1,length=4), x2=seq(0,1,length=4))
y <- apply(design.fact, 1, branin)

# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
response.fact<-cbind(y)
mat=matrix(data=c(0,0,0,500,5,5), ncol=2)
boundariesP=rbind(c(0,500))
pas=cbind(c(0.1,0.1,0.1))
eph1 <- eph(design.fact, response.fact, mat, boundariesP, pas)
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')
# graphics
n.grid <- 20
x.grid <- y.grid <- seq(0,1,length=n.grid)
design.grid <- expand.grid(x1=x.grid, x2=y.grid)
response.grid <- apply(design.grid, 1, branin)

predicted.values <- predict.eph(eph1, design.grid)
predicted.values.model1 <- predicted.values$mean
predicted.values.model3 <- predicted.values$lower95
predicted.values.model4 <- predicted.values$upper95
predicted.values.model5 <- predicted.values$esperance

par(mfrow=c(2,1))
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")

points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,
      main=" experimental probabilistic hypersurface")
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")

```

```

# -----

```

C. Reconstruction de 400 points à partir de 9 points de mesure

1. Krigeage Ordinaire :

```
# -----  
# A 2D example - Branin-Hoo function  
# -----  
# a 9-points factorial design, and the corresponding response  
d <- 2; n <- 9  
design.fact <- expand.grid(x1=seq(0,1,length=3), x2=seq(0,1,length=3))  
y <- apply(design.fact, 1, branin)  
  
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect  
m1 <- km(design=design.fact, response=y)  
  
# graphics  
n.grid <- 20  
x.grid <- y.grid <- seq(0,1,length=n.grid)  
design.grid <- expand.grid(x1=x.grid, x2=y.grid)  
response.grid <- apply(design.grid, 1, branin)  
  
predicted.values <- predict(m1, design.grid, "UK")  
predicted.values.model5 <- predicted.values$mean  
predicted.values.model3 <- predicted.values$lower95  
predicted.values.model4 <- predicted.values$upper95  
  
par(mfrow=c(2,1))  
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")  
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")  
  
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,  
      main="Ordinary Kriging")  
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")  
  
# -----
```

2. Hypersurface Probabiliste Experimentale :

```
# -----  
# A 2D example - Branin-Hoo function  
# -----  
  
setClass("eph",slots=c(NofParam="integer", NofMeas="integer", NofInk2="integer",  
x="matrix", y="matrix", step="matrix", boundarie-  
sin="matrix",boundariesout="matrix", covariance="covTensorProduct"))  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')  
# a 9-points factorial design, and the corresponding response  
d <- 2; n <- 9  
design.fact <- expand.grid(x1=seq(0,1,length=3), x2=seq(0,1,length=3))  
y <- apply(design.fact, 1, branin)  
  
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect  
response.fact<-cbind(y)  
mat=matrix(data=c(0,0,0,500,5,5), ncol=2)
```

```

boundariesP=rbind(c(0,500))
pas=cbind(c(0.1,0.1,0.1))
eph1 <- eph(design.fact, response.fact, mat, boundariesP, pas)
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')
# graphics
n.grid <- 20
x.grid <- y.grid <- seq(0,1,length=n.grid)
design.grid <- expand.grid(x1=x.grid, x2=y.grid)
response.grid <- apply(design.grid, 1, branin)

predicted.values <- predict.eph(eph1, design.grid)
predicted.values.model1 <- predicted.values$mean
predicted.values.model3 <- predicted.values$lower95
predicted.values.model4 <- predicted.values$upper95
predicted.values.model5 <- predicted.values$esperance

par(mfrow=c(2,1))
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")

points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,
      main=" experimental probabilistic hypersurface")
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")

# ---
-----

```

D. Reconstruction de 400 points à partir de 3 points de mesure

1. Krigeage Ordinaire :

```

# -----
# A 2D example - Branin-Hoo function
# -----

# a 3-points factorial design, and the corresponding response
d <- 2; n <- 4
design.fact<-      matrix(data=c(0,0.5,1,0,1,0),      ncol=2,      dimnames      =
list(c(1,2,3),c("x1","x2")))
y <- apply(design.fact, 1, branin)
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
m1 <- km(design=design.fact, response=y)

# graphics
n.grid <- 20
x.grid <- y.grid <- seq(0,1,length=n.grid)
design.grid <- expand.grid(x1=x.grid, x2=y.grid)
response.grid <- apply(design.grid, 1, branin)

predicted.values <- predict(m1, design.grid, "UK")
predicted.values.model5 <- predicted.values$mean
predicted.values.model3 <- predicted.values$lower95
predicted.values.model4 <- predicted.values$upper95

```

```

par(mfrow=c(2,1))
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")

points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,
      main="Ordinary Kriging")
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
# -----

```

2. Hypersurface Probabiliste Experimentale :

```

# -----
# A 2D example - Branin-Hoo function
# -----

setClass("eph",slots=c(NofParam="integer", NofMeas="integer", NofInk2="integer",
x="matrix", y="matrix", step="matrix", boundaries="matrix",
sin="matrix",boundariesout="matrix", covariance="covTensorProduct"))
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')
# a 3-points factorial design, and the corresponding response
d <- 2; n <- 4
design.fact<- matrix(data=c(0,0.5,1,0,1,0), ncol=2, dimnames =
list(c(1,2,3),c("x1","x2"))))
y <- apply(design.fact, 1, branin)

# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
response.fact<-cbind(y)
mat=matrix(data=c(0,0,0,500,5,5), ncol=2)
boundariesP=rbind(c(0,500))
pas=cbind(c(0.1,0.1,0.1))
eph1 <- eph(design.fact, response.fact, mat, boundariesP, pas)
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')
# graphics
n.grid <- 20
x.grid <- y.grid <- seq(0,1,length=n.grid)
design.grid <- expand.grid(x1=x.grid, x2=y.grid)
response.grid <- apply(design.grid, 1, branin)

predicted.values <- predict.eph(eph1, design.grid)
predicted.values.model1 <- predicted.values$mean
predicted.values.model3 <- predicted.values$lower95
predicted.values.model4 <- predicted.values$upper95
predicted.values.model5 <- predicted.values$esperance

par(mfrow=c(2,1))
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")

points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,
      main=" experimental probabilistic hypersurface")
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")
# -----

```

E. Reconstructions à partir de points choisis de manière aléatoire

1. Fonction de Branin

```
# -----  
# A 2D example - Branin-Hoo function  
# -----  
  
setClass("eph", slots=c(NofParam="integer", NofMeas="integer", NofInk2="integer",  
x="matrix", y="matrix", step="matrix", boundarie=  
sin="matrix", boundariesout="matrix", covariance="covTensorProduct"))  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')  
# a 16-points factorial design, and the corresponding response  
d <- 2; n <- 16  
design.fact <- data.frame(x1= runif(16, 0, 1), x2= runif(16, 0, 1))  
y <- apply(design.fact, 1, branin)  
  
# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect  
response.fact<-cbind(y)  
mat=matrix(data=c(0,0,0,320,1.1,1.1), ncol=2)  
boundariesP=rbind(c(0,320))  
pas=cbind(c(0.1,0.1,0.1))  
eph1 <- eph(design.fact, response.fact, mat, boundariesP, pas)  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')  
# graphics  
n.grid <- 20  
x.grid <- y.grid <- seq(0,1,length=n.grid)  
design.grid <- expand.grid(x1=x.grid, x2=y.grid)  
response.grid <- apply(design.grid, 1, branin)  
  
predicted.values<- predict.eph(eph1, design.grid)  
predicted.values.model1 <- predicted.values$mean  
predicted.values.model3 <- predicted.values$lower95  
predicted.values.model4 <- predicted.values$upper95  
predicted.values.model5 <- predicted.values$esperance  
  
par(mfrow=c(2,1))  
contour(x.grid, y.grid, matrix(response.grid, n.grid, n.grid), 20,main="Branin")  
  
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")  
contour(x.grid, y.grid, matrix(predicted.values.model5, n.grid, n.grid), 20,  
main=" experimental probabilistic hypersurface")  
points(design.fact[,1], design.fact[,2], pch=17, cex=1.5, col="blue")  
# -----
```

2. Autre fonction de test

```
# -----  
# A 2D example - Function  $(x_1, x_2) \rightarrow \sin(5.5 * x_1) * \cos(5 * x_2) + x_1^2 + 1$ .  
# -----  
  
setClass("eph", slots=c(NofParam="integer", NofMeas="integer", NofInk2="integer",  
x="matrix", y="matrix", step="matrix", boundarie=  
sin="matrix", boundariesout="matrix", covariance="covTensorProduct"))  
source('C:/Users/scm sa/Desktop/Frederic/IRSN/eph.r')
```

```

# a 16-points factorial design, and the corresponding response
d <- 2; n <- 16
design.fact <- data.frame(x1= runif(32, -1, 1), x2= runif(32, -1, 1))
y <- sin(5.5* design.fact$x1)*cos(5* design.fact$x2)+ design.fact$x1* de-
sign.fact$x1+1

# kriging model 1 : matern5_2 covariance structure, no trend, no nugget effect
response.fact<-cbind(y)
mat=matrix(data=c(0.1,-1,-1,2.8,1,1), ncol=2)
boundariesP=rbind(c(0.1,2.8))
pas=cbind(c(0.1,0.1,0.1))
eph1 <- eph(design.fact, response.fact, mat, boundariesP, pas)
source('C:/Users/scm sa/Desktop/Frederic/IRSN/irsn_scm4_function (2).r')
# graphics
n.grid <- 20
x.grid <- y.grid <- seq(-1,1,length=n.grid)
design.grid <- data.frame(x1= runif(400, -1, 1), x2= runif(400, -1, 1))
response.grid <- sin(5.5* design.grid$x1)*cos(5* design.grid$x2)+ de-
sign.grid$x1* design.grid$x1+1
predicted.values<- predict.eph(eph1, design.grid)
predicted.values.model1 <- predicted.values$mean
predicted.values.model3 <- predicted.values$lower95
predicted.values.model4 <- predicted.values$upper95
predicted.values.model5 <- predicted.values$esperance

m1 <- km(design=design.fact, response=y)
predicted.values.model51 <- predict(m1, design.grid, "UK")$mean
predicted.values.model31 <- predict(m1, design.grid, "UK")$lower95
predicted.values.model41 <- predict(m1, design.grid, "UK")$upper95

```