

Métamodélisation pour une conception robuste des systèmes mécatroniques

Metamodeling for robust design of mechatronic systems

Hamid Hamdani^{1,2}, Bouchaïb Radi¹, Abdelkhalak El Hami²

¹ Laboratoire d'Ingénierie, Management Industriel et Innovation (LIMII), université Hassan 1^{er}, Settat, Maroc, hamid.hamdani@insa-rouen.fr, bouchaib.radi@yahoo.fr

² Laboratoire de Mécanique de Normandie (LMN), INSA Rouen, France, abdelkhalak.elhami@insa-rouen.fr

RÉSUMÉ. Face aux exigences concurrentielles et économiques actuelles dans le secteur industriel, Les outils de simulation numérique, tels que les méthodes des éléments finis, sont de plus en plus largement appliqués aux problèmes de conception des systèmes mécatroniques. De nombreux problèmes nécessitent un grand nombre de simulations pour évaluer une fonction objectif. Cependant, pour de nombreux cas, une seule simulation peut prendre plusieurs minutes, heures, ou même des jours pour converger. Par conséquent, les tâches à forte intensité de simulations, telles que l'analyse de sensibilité, l'analyse de fiabilité, l'optimisation, et l'optimisation fiabiliste deviennent impraticables ou presque impossibles, car elles nécessitent des centaines, des milliers ou même des millions de simulations. La construction des modèles d'approximation devient la méthode la plus robuste pour remédier à ce problème. Ces modèles connus sous le nom de métamodèles, permettent de rapprocher le plus possible la relation entrée-sortie (input-output) du modèle de simulation élément finis, tout dans le but de réduire le coût d'évaluation. Finalement les tâches à grand nombre de simulations peuvent être mises en œuvre en utilisant le métamodèle construit. Cet article présente les métamodèles les plus populaires, leurs méthodes de validation, ainsi que des exemples d'étude comparative pour un choix optimal du métamodèle convenable au problème.

ABSTRACT. In the face of current competitive and economic demands in the industrial sector, numerical simulation tools, such as finite element methods, are increasingly applied to the design problems of mechatronic systems. Many problems require a large number of simulations to evaluate an objective function. However, for many cases, a single simulation can take several minutes, hours, or even days to converge. As a result, simulation-intensive tasks such as sensitivity analysis, reliability analysis, optimization, and reliability based design optimization become impractical or almost impossible, as they require hundreds, thousands or even millions of simulations. The construction of the approximation models becomes the most robust method to remedy this problem. These models, known as metamodels, make it possible to approximate as much as possible the input-output relation of the finite element simulation model, all with the aim of reducing the evaluation cost. Finally, tasks with large number of simulations can be implemented using the meta model built. This article present the most popular metamodeling techniques, their validation methods, as well as examples of comparative studies for an optimal choice of the metamodel suitable for the problem.

MOTS-CLÉS. Systèmes mécatroniques, Plans d'expériences, Métamodèle, Modèle numérique.

KEYWORDS. Mechatronic systems, Design of experiments, Metamodel, Numerical model.

Introduction

Les principales méthodologies pour l'amélioration des techniques numériques de conception fiabiliste des systèmes mécatroniques complexes [ELH 15], se basent sur la construction d'un métamodèle qui peut être considéré comme un substitut du modèle numérique (modèle élément finis). La construction du métamodèle se fait par l'exécution d'un nombre limité de simulations (plan d'expérience DOE) représentant des combinaisons des différents facteurs (paramètres).

Dans ce contexte, Le choix du plan d'expérience et la détermination du métamodèle, qui dépend fortement de la nature et l'objectif du problème, revêtent une grande importance. Une fois le métamodèle est construit, Une phase de validation du modèle est alors requise. Tout d'abord Il faut quantifier l'erreur d'apprentissage et vérifier la validité du modèle. Deuxièmement, ce Métamodèle est le point clé de l'étude car il sera utilisé comme substitut du modèle éléments finis. Il est alors important de tester la performance prédictive du modèle à des points qui n'appartiennent pas à l'ensemble des points de formation.

Dans cet article, la première partie est dédiée à la construction des plans d'expériences, et les critères de distance pour l'étude des distributions des points. La deuxième partie consiste à présenter les métamodèles les plus populaires, ainsi que les outils pour leur évaluation et leur validation. La troisième partie est dédiée à l'étude d'une application pour illustrer et mettre en œuvre les différentes parties de choix et création des métamodèles.

1. Création des plans d'expériences

Avant la création des plans d'expériences il est indispensable de déterminer les paramètres influents dans l'ensemble des facteurs initiaux, et cela se fait via un procédé de criblage (screening) [ELH 15] qui permet de conserver seulement les paramètres dont l'effet est jugé significatif. Une fois les paramètres sont définis, on peut passer à la création des plans d'expériences, ces derniers sont généralement basés sur des Latin hypercubes [MCK 79, STE 87], des autres techniques de plans d'expériences qui sont plus classique sont connus sous le nom de plans factoriels [ELH 15], tel que :

- Plans factoriel complet.
- Plans centraux composites (CC).
- Plans de Box Behnken.
- Plans de Doehlert ...

En plus de ces plans d'expérience, d'autres plans y compris, les plans quasi-Monte Carlo, les plans Uniformes, les plans de Strauss [FRA 8] et les plans Dmax (Maximum Entropy Desing) [SHE 87], peuvent être trouvés dans la littérature. Les plans d'expérience cités au-dessus sont tous implémentés dans le logiciel R sous forme de packages. (Voir, par exemple, les packages lhs, [CAR 16], pour hypercube latin), la figure 1 présente un plan latin hypercube de dimension 2.

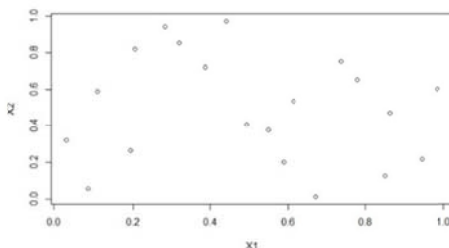


Figure 1. Plan latin hypercube de 20 point

Un bon résumé des plans d'expériences peut être trouvé à [SAC 12, GIU 3].

1.1. Métamodèles

Une fois le plan d'expérience est choisi ainsi que l'ensemble des cofacteurs est défini par ce dernier, l'étape qui suit consiste à construire un métamodèle qui décrit la relation entre les entrées et les sorties.

Généralement les n échantillons, de dimension d , tiré du plan d'expérience, peuvent être représentés par une matrice X de dimension $n \times d$ et leur réponse par un vecteur Y .

$$X = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \dots & \vdots \\ x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix}, \quad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} f(X^{(1)}) \\ \vdots \\ f(X^{(n)}) \end{bmatrix} \quad [1]$$

où $x^{(i)}$, $i = 1, \dots, n$ est le i -ième échantillon, composé de de d variables, $x_{(j)}^{(i)}$, $i = 1, \dots, n$ et $j = 1, \dots, d$ est le j -ième composant de l'échantillon i .

f est une fonction qui représente le modèle numérique à remplacer par le métamodèle. Les $y^{(i)}$, $i = 1, \dots, n$ correspondent aux réponses du modèle numérique aux entrées $x^{(i)}$. Notre objectif est de construire un métamodèle qui peut imiter la réponse du modèle numérique coûteux, et qu'on utilise ce métamodèle comme un substitut du modèle numérique dans la prédiction des réponses de ce dernier, Dans cette partie on présente les techniques de méta-modélisation les plus populaires. [SAC 12].

1.2. Méthode des surfaces de réponse

La surface de réponse polynomiale (SRP), guidée par les séries de Taylor, est une approximation de forme polynomiale. Le nombre de terme dans SRP dépend fortement de la fonction réelle et de l'ordre de précision désiré. La SRP la plus largement utilisée est celle de second ordre, ou la surface de réponse quadratique (QRS) [HUA 16]. Le QRS d'une variable de dimension m est détaillé dans ce qui suit :

L'expression générale d'une QRS est :

$$y(x) = y(x) + \varepsilon \quad [2]$$

où $y(x)$ est l'approximation polynomiale quadratique et ε est l'erreur aléatoire supposé être distribué suivant une loi normale de moyen zéro et de variance σ^2 . Le prédicteur QRS $y(x)$ peut être défini comme :

$$y(x) = d_0 + \sum_{i=1}^m d_i x_i + \sum_{i=1}^m \sum_{j>1}^m d_{ij} x_i x_j + \sum_i^m d_{ii} x_i^2 \quad [3]$$

où d_0, d_i, d_{ij} , et d_{ii} sont des coefficients de régressions à déterminer. Soit θ le vecteur contenant les p coefficients inconnus. Ce qui veut dire qu'on est besoin de déterminer les $p = \frac{(m+1)(m+2)}{2}$ coefficients, nécessaire pour la construction du métamodèle QRS. L'équation 3 peut s'écrire sous forme matricielle :

$$y_s = X\theta \quad [4]$$

Avec

$$y_s = [y^{(1)} \quad y^{(2)} \quad \dots \quad y^{(n)}]^T \quad [5]$$

$$\theta = [d_0 \quad d_1 \quad d_m \dots d_{12} \quad \dots \quad d_{[m-1]m} \quad d_{11} \quad d_{mm}]^T \quad [6]$$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_m^{(1)} & x_1^{(1)} & x_2^{(1)} & \dots & x_{m-1}^{(1)} & x_m^{(1)} & (x_1^{(1)})^2 & \dots & (x_1^{(1)})^2 \\ 1 & x_1^{(2)} & \dots & x_m^{(2)} & x_1^{(2)} & x_2^{(2)} & \dots & x_{m-1}^{(2)} & x_m^{(2)} & (x_1^{(2)})^2 & \dots & (x_m^{(2)})^2 \\ \vdots & \vdots & & \vdots & & & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \dots & x_m^{(n)} & x_1^{(n)} & x_2^{(n)} & \dots & x_{m-1}^{(n)} & x_m^{(n)} & (x_1^{(n)})^2 & \dots & (x_m^{(n)})^2 \end{bmatrix} \quad [7]$$

Afin d'utiliser ce modèle, nous devons déterminer le vecteur θ . On va l'estimer par méthode des moindres carrés. On dispose pour cela d'un échantillon de n observations (X_i, Y_i) et on cherche le vecteur θ qui minimise l'erreur d'estimation au carré. L'estimateur des moindres carrés obtenu est :

$$\theta = (X^T X)^{-1} X^T y_S \quad [8]$$

Une fois le vecteur des coefficients θ est déterminé, le modèle d'approximation $y(x)$ à n'importe quel point x , hors des points du plan d'expérience, peut être prédit efficacement par l'équation 3. Le métamodèle SRP, résultant, ne passe pas nécessairement à travers les données d'échantillon de formation, c'est-à-dire qu'elle n'a pas nécessairement une erreur zéro lors des points de formation. C'est pourquoi les fonctions PRS sont souvent appelées fonctions d'approximation.

1.3. Métamodèle de Kriging

Le métamodèle de Kriging est une technique de métamodélisation souvent utilisée dans le cadre géostatistique. Le Kriging s'apparente à un modèle linéaire avec résidus non indépendants tout en minimisant la variance de l'erreur d'estimation [SEE 4, KLE 9], ceci menant à des interpolations. Le but est d'exprimer un échantillon des observations $Y_i \ i = 1, \dots, n$ en fonction d'un ensemble de données $x_i = [x_1 \ x_2 \ \dots \ x_p]$ avec $x_i \in R^n$.

Le modèle de kriging peut être considéré comme la réalisation d'un modèle de régression et d'un processus aléatoire Z , nous donnant ainsi :

$$y(x) = R(x) + Z(x) \quad [9]$$

où $R(x)$ est une fonction de tendance globale, qui est généralement de forme polynômial et supposée être présentée comme une combinaison linéaire de fonctions de régression choisies (constante, linéaire, quadratique) $f_0(x) \ f_1(x) \ \dots \ f_{p-1}(x)$ avec des coefficients non nuls $\beta_0, \beta_1, \dots, \beta_{p-1}$, donc :

$$R(x) = \sum_{i=0}^{p-1} \beta_i f_i(x) = f^T(x) \beta \quad [10]$$

Le vecteur $\beta = [\beta_0 \ \beta_1 \ \dots \ \beta_{p-1}]^T$ est l'ensemble des paramètres de la régression, et $f(x) = [f_0(x), f_1(x), \dots, f_{p-1}(x)]^T$ est le vecteur des fonctions de régression.

Le processus aléatoire Z , d'espérance nulle (kriging simple) et de covariance entre $Z(v)$ et $Z(w)$ est donnée par :

$$\text{Cov}(Z(v), Z(w)) = \sigma^2 R(\theta, v, w) \quad [11]$$

Où σ^2 est la variance du processus Z et $R(\theta, v, w)$ la fonction de corrélation de paramètre θ . Ces fonctions de corrélation sont définies comme suit :

$$R(\theta, v, w) = \prod_{k=1}^m R_k(\theta_k, v_k, w_k) \quad [12]$$

où v_k et w_k indique respectivement la $k^{\text{ième}}$ composant du point d'expérience et w , m est la dimension du plan d'expérience. Il existe plusieurs fonctions de corrélations dans la littérature, dont les plus appliquées sont [SAC 12] :

$$\begin{aligned} R(\theta, v, w) &= \exp(-\theta_k |v_k - w_k|) \\ R(\theta, v, w) &= \exp(-\theta_k |v_k - w_k|^2) \end{aligned} \quad [13]$$

où θ_k ($k=1, 2, \dots, m$) sont des paramètres de la fonction de corrélation, et $|v_k - w_k|$ et la distance absolue entre v_k et w_k , en remplaçant les expressions des équations 13 dans 12 on obtient :

$$R(\theta, v, w) = \exp \left[- \sum_{k=1}^m \theta_k |v_k - w_k| \right] \quad [14]$$

$$R(\theta, v, w) = \exp \left[- \sum_{k=1}^m \theta_k |v_k - w_k|^2 \right] \quad [15]$$

On utilise un vecteur $\theta = [\theta_1 \ \theta_2 \ \dots \ \theta_m]^T$ de dimension $m \times 1$ pour représenter les paramètres inconnus des fonctions de corrélation.

Finalement le modèle prédicteur de Kriging pour tout point x peut se mettre sous la forme :

$$\hat{y}(x) = f^T(x) \beta + r^T(x) R^{-1} (y_s - F^T \beta) \quad [16]$$

avec $f^T(x)$ est le vecteur des fonctions de régression, β sont les paramètres de la régression, $r(x)$ est le vecteur des fonctions de corrélations entre les points d'observations, et R est une matrice de dimension $n \times n$ de l'ensemble de données (19), y_s est le vecteur des réponses observées correspondant à l'ensemble des entrées, F est la matrice des fonctions de régression.

Le vecteur $(y_s - F^T \beta)$ indique le vecteur des erreurs (résidus), de l'ensemble des données lorsque le terme du processus aléatoire est éliminé.

$$F = [f^{(T)}(X^{(1)}) \ f^{(T)}(X^{(2)}) \ \dots \ f^{(T)}(X^{(n)})]^T \quad [17]$$

$$r(x) = \begin{bmatrix} R(X, X^{(1)}) \\ R(X, X^{(2)}) \\ \vdots \\ R(X, X^{(n)}) \end{bmatrix} \quad [18]$$

$$R(x) = \begin{bmatrix} R(X^{(1)}, X^{(1)}) & R(X^{(1)}, X^{(2)}) & \dots & R(X^{(1)}, X^{(n)}) \\ R(X^{(2)}, X^{(1)}) & R(X^{(2)}, X^{(2)}) & \dots & R(X^{(2)}, X^{(n)}) \\ \vdots & \vdots & & \vdots \\ R(X^{(n)}, X^{(1)}) & R(X^{(n)}, X^{(2)}) & \dots & R(X^{(n)}, X^{(n)}) \end{bmatrix} \quad [19]$$

Pour construire le métamodèle de Kriging, il faut déterminer les coefficients de régression β les paramètres de corrélation θ ainsi que la variance du processus aléatoire σ^2 , pour se faire la méthode du maximum de vraisemblance est utilisé.

Le modèle de kriging suppose que les observations (réponse) sont obtenues à partir d'un processus gaussien, Les réponses aux sites d'échantillonnage sont considérées comme des fonctions aléatoires, corrélées avec la fonction de vraisemblance correspondante donnée par :

$$L(\beta, \sigma^2, \theta) = \frac{1}{(2\pi\sigma^2)^{n/2} |R|^{1/2}} \times \exp \left[- \frac{(y_s - F^T \beta) R^{-1} (y_s - F^T \beta)}{2\sigma^2} \right] \quad [20]$$

En prenant le logarithme népérien de l'expression ci-dessus en obtient :

$$\ln(L) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{1}{2} |R| - \frac{(y_s - F^T \beta) R^{-1} (y_s - F^T \beta)}{2\sigma^2} \quad [21]$$

On obtient le maximum de vraisemblance par dérivation de l'équation 21 par rapport à β et σ^2 , ainsi :

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y_s \quad [22]$$

$$\hat{\sigma}^2 = \frac{1}{n} (y_s - F^T \hat{\beta})^T R^{-1} (y_s - F^T \hat{\beta}) \quad [23]$$

En remplaçant les équations 22 et 23 dans l'équation 20 Nous obtenons :

$$\ln(L) - \frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} |R| \quad [24]$$

Finalement l'estimateur $\hat{\theta}$ du vecteur θ est obtenu par maximisation de l'équation 24 sous la contrainte $\theta_l > 0, l = 1, 2, \dots, m$, autrement dit le vecteur θ est obtenu par résolution d'un problème d'optimisation suivant :

$$\begin{cases} \text{Max}_{\theta} L(\theta) \\ \text{s. t. } \theta_l > 0, l = 1, 2, \dots, m \end{cases} \quad [25]$$

Ce problème d'optimisation peut être résolu numériquement, des méthodes de recherche de l'optimum global, comme l'algorithme génétique donne des bons résultats.

Une fois l'estimateur $\hat{\theta}$ est obtenu, $\hat{\beta}$ et $\hat{\sigma}^2$ sont obtenus à partir des équations 22 et 23, et finalement tous les inconnus nécessaires à la construction du modèle de Kriging sont déterminés, et le modèle peut être complètement construit, et la prédiction par le modèle peut être estimée par l'équation 16.

1.4. Réseaux à fonctions de bases radiales (RBF)

Appelées aussi réseaux de neurones à fonctions radiales de base, sont des réseaux de neurones qui utilisent des bases radiales comme des fonctions d'activation (cf. figure 2).

Les RBF sont composées de trois couches principales : une couche d'entrée, une couche cachée composée de fonctions noyaux et une couche de sortie [WU 12], dont les neurones sont généralement animés par une fonction d'activation linéaire.

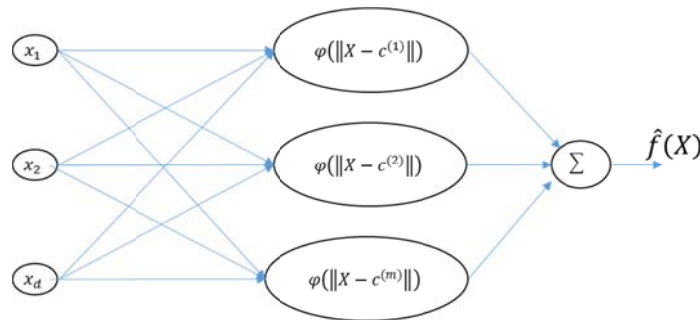


Figure 2. Structure du Métamodèle RBF

Chaque neurone de la couche cachée réalise une fonction noyau. Ces neurones calculent la distance de l'entrée X à leurs noyaux $c^{(i)}$ et répondent par une activation en proportion de cette distance.

Le modèle prédictif RBF pour tout point x est donné par une combinaison des fonctions de bases radiales :

$$f(X) = \sum_{i=1}^m w_i \varphi(\|X - c^{(i)}\|) \quad [26]$$

avec w_i et le poids du le i-eme fonction de base radial.

Les fonctions de base non-paramétrique les plus utilisées [MES 15] sont : linéaire : $\varphi(r) = r$; cubique : $\varphi(r) = r^3$ et Spline en Plaque Mince $=r^2 \ln(r)$. Des fonctions de base paramétriques peuvent être utilisés pour améliorer les propriétés de généralisation du model : multi-quadratique : $\varphi(r) = (r^2 + \sigma^2)^{-1/2}$; Gaussien : $\varphi(r) = e^{-r^2/(2\sigma^2)}$ et inverse multi quadratique : $\varphi(r) = (r^2 + \sigma^2)^{-1/2}$. Après le choix des fonctions de base à utiliser (paramétrique ou non-paramétrique), le vecteur des poids $W = [w_1, \dots, w_m]^T$ peut être évalué via la condition d'interpolation [FOR 8], $X = [X^{(1)}, \dots, X^{(n)}]^T$ est l'ensemble des données d'apprentissage et $y = [y^{(1)}, \dots, y^{(n)}]^T$ leurs réponses correspondantes, la condition d'interpolation peut s'écrire sous forme matricielle

$$\varphi W = y \quad [27]$$

avec

$$\varphi = \begin{bmatrix} \varphi(\|X^{(1)} - c^{(1)}\|) & \varphi(\|X^{(1)} - c^{(2)}\|) & \dots & \varphi(\|X^{(1)} - c^{(m)}\|) \\ \varphi(\|X^{(2)} - c^{(1)}\|) & \varphi(\|X^{(2)} - c^{(2)}\|) & \dots & \varphi(\|X^{(2)} - c^{(m)}\|) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|X^{(n)} - c^{(1)}\|) & \varphi(\|X^{(n)} - c^{(2)}\|) & \dots & \varphi(\|X^{(n)} - c^{(m)}\|) \end{bmatrix} \quad [28]$$

A noter que si $m = n$ les noyaux des fonctions de bases coïncident avec les données d'apprentissages, $c^{(i)} = X^{(i)}$, $i=1, \dots, n$, et φ est une Matrice carrée ordinaire, et le système à une solution unique.

$$W = \varphi^{-1}y \quad [29]$$

1.5. Autres métamodèles

De plus des métamodèles cités précédemment, d'autre techniques de métamodélisation y compris les réseaux de nouerons, le modèle MARS (multivariate adaptative regression splines [FRI 91], le modèle PolyMARS (Polychotomous Regression Based in MARS [KOO 97]) peut être trouvées dans la littérature.

2. Validation du Méta modèle

2.1. Validation

Il est intéressant d'avoir des critères d'évaluation des différents méta-modèles construits, afin de pouvoir estimer leurs qualités. Deux cas à distinguer : l'évaluation de la qualité du model sur l'ensemble d'apprentissage, et l'évaluation de la qualité sur un jeu de données hors de l'ensemble d'apprentissage.

Soit y_i la valeur de la réponse du modèle de simulation au point d'apprentissage x_i et \hat{y}_i est la valeur prédite par le métamodèle. La première méthode la plus populaire pour mesurer la qualité d'un métamodèle est le coefficient de détermination défini par :

$$R2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad [30]$$

avec \bar{y} est la moyenne de l'ensemble des observations y_i .

D'autres méthodes pour quantifier la qualité du métamodèle sont :

La racine des erreurs des moindres carrés :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad [31]$$

– La moyenne des erreurs absolues :

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad [32]$$

– Et Erreur maximum relative absolue :

$$RMA = \frac{\max_{1 \leq i \leq n} |y_i - \hat{y}_i|}{\sigma_y} \quad [33]$$

avec σ_y est l'écart-type de y_1, \dots, y_n .

En plus de ces différents critères de validation des métamodèles, la validation peut se faire par la méthode la plus populaire qui correspond à l'étude de la distribution des résidus. $y_i - \hat{y}_i$.

2.1. Validation croisée

La validation croisée est une méthode de rééchantillonnage utilisée pour tenir compte de l'erreur de généralisation lorsqu'aucun jeu de test n'est disponible. Cette technique peut également être utilisée pour la sélection du métamodèle, pour détecter le dépassement ou l'instabilité du modèle. Le principe est assez simple : l'ensemble de données est divisé de manière aléatoire en sous-ensembles K de taille approximativement égale : A_1, \dots, A_K . Pour chaque $k = 1, \dots, K$, un modèle \hat{Y}^{-k} est construit à base des de l'extrait de données $\cup_{j \neq k} A_j$, et ce modèle est validé sur le sous-ensemble A_k .

L'indice Q2 utilisé dans la validation croisé et donné par :

$$RMSE_{CV} = Q2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{Y}^{k(i)}(x^i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad [34]$$

Il s'agit de la même expression de R2 sauf qu'on remplace les valeurs \hat{y}_i prédite, par les valeurs obtenues par validation croisée.

3. Exemple d'Application

On considère l'exemple d'une fonction à deux dimensions :

$$f(x_1, x_2) = 3.5[\exp(-(4x_1 - 1)^2) + \exp(-((4x_1 - 3)^2))] - \sin(2\pi x_2) + 2x_1 x_2 - 1.5 \quad [35]$$

avec $0 \leq x_1 \leq 1$ et $0 \leq x_2 \leq 1$.

L'objectif de cette application est de mettre en œuvre les différentes étapes citées précédemment, ainsi que visualiser l'influence du choix du plan d'expérience sur la qualité du métamodèle. Pour se faire, le logiciel R doté de ses différentes packages [DUP 15] est utilisé comme outil pour la création des différents plans d'expériences, la construction des métamodèles, et leur validation ainsi que leur comparaison.

3.1. Choix du plan d'expérience

La première étape dans cette étude consiste à choisir un plan d'expérience qui sera la base de construction des métamodèles. Pour cela nous avons choisi de générer un plan d'expérience de 20 points. Dans le cas de cet exemple, quatre types de plan d'expérience sont choisis et comparés, le premier correspond à un plan de distribution Uniform, le deuxième est celui de latin hypercube, le troisième est plan Dmax avec les paramètres (range=1, niter_max=10000), et le dernier correspond à un plan de Strauss optimisé avec les paramètres (RND = 0.21, $\alpha = 0.2$ & $\gamma = \exp(-10)$). Le critère de mindist (distance minimale) dans la figure 3 montre une grande différence entre le plan de Strauss et les autres plans, le plan de Strauss à un critère mindist, plus que 0.2 (voir figure 3).

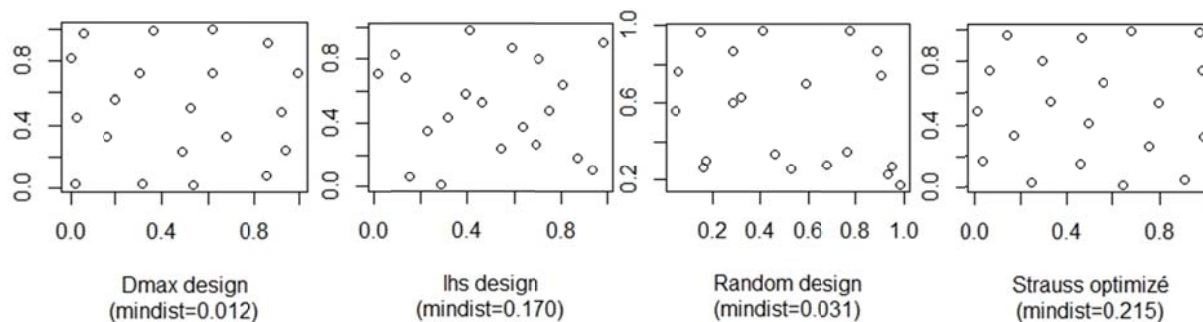


Figure 3. Plans d'expériences de 20 points en dimension 2

3.2. Construction, comparaison et validation des métamodèles

Le logiciels R, muni de ces différentes packages [CAR 16,DUP 15], contient des fonctions principales dédiés à la construction et l'étude des critères de validation cités précédemment. Pour ce cas d'exemple, cinq techniques de métamodélisation sont choisis, le modèle polynomial(PRS), le modèle additive, MARS, PolyMARS et le modèle de Kriging. Le tableau 1 présente une étude comparative pour l'évaluation de la qualité des différents plans d'expériences, ainsi que les différents métamodèles construits, pour l'exemple de deux dimensions. Le critère R2 du paragraphe VI.1 est utilisé comme critère de validation.

	MARS (degree =2)	PolyMARS (gcv=2)	PRS	Additive	Kriging
Random design	0.635	0.714	0.623	0.909	0.973
Dmax	0.653	0.604	0.645	0.933	0.977
Optimized Strauss	0.561	0.654	0.677	0.935	0.988
LHS design	0.261	0.175	0.386	0.549	0.874

Tableau 1. Comparaison de la qualité de prédiction des différents plans d'expériences, et les différents types de métamodèles construit

La construction des métamodèles est réalisée à travers le package DiceEval [DUP 15] du logiciel R sauf que pour le modèle de Kriging qui était construit en utilisant le package DiceKriging [ROU 12]. Pour tous les plans et les métamodèles, les résultats de l'exemple présentent des valeurs élevées de R2 pour le modèle de Kriging, et cela montre sa performance par rapport aux autres métamodèles.

Conclusion

Cet article, représente une vue d'ensemble des techniques de métamodélisation, dans le but de faire une étude comparative, afin de mettre le choix sur le métamodèle le plus performant. L'étude réalisée, a permis d'effectuer une synthèse de plusieurs méthodes de méta-modélisation permettant ainsi d'en avoir une approche simple. Ces méthodes ont ensuite été comparées sur un exemple, en utilisant le logiciel R comme outil, ces comparaisons permettent d'évaluer, de comparer, et de choisir le métamodèle le plus performant pour remplacer le modèle numérique d'origine.

Bibliographie

- [CAR 16] Carnell, R. (2016). Package “lhs.”
- [DUP 15] Dupuy, D., Helbert, C., & Franco, J. (2015). DiceDesign and DiceEval: Two R Packages for Design and Analysis of Computer Experiments. *Journal of Statistical Software*, 65(11), 1–38. <https://doi.org/10.18637/jss.v065.i11>
- [ELH 15] El Hami, A., & Pougnet, P. (2015). Les systèmes mécatroniques embarqués 2 Analyse des causes de défaillances, modélisation, simulation et optimisation.
- [FOR 8] Forrester, A. I. J., Sobester, A., & Keane, A. J. (2008). Engineering Design via Surrogate Models.
- [FRA 8] Franco, J., Bay, X., Dupuy, D., Corre, B., Franco, J., Bay, X., ... Corre, B. (2008). eriences num ´ a partir du processus ponctuel de Strauss To cite this version :
- [FRI 91] Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19, 1–67. <https://doi.org/10.1214/aos/1176347963>
- [GIU 3] Giunta, A. A., Wojtkiewicz, S., & Eldred, M. S. (2003). Overview of modern design of experiments methods for computational simulations. *Aiaa*, 649, 6–9. <https://doi.org/10.2514/6.2003-649>
- [HUA 16] Huang, C., Radi, B., & Hami, A. El. (2016). Uncertainty analysis of deep drawing using surrogate model based probabilistic method. *The International Journal of Advanced Manufacturing Technology*, 86(9), 3229–3240. <https://doi.org/10.1007/s00170-016-8436-4>
- [KLE 9] Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*. <https://doi.org/10.1016/j.ejor.2007.10.013>
- [KOO 97] Kooperberg, C., Bose, S., & Stone, C. J. (1997). Polychotomous regression. *Journal of the American Statistical Association*, 92(437), 117–127. <https://doi.org/10.1080/01621459.1997.10473608>
- [MCK 79] McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2), 239–245. <https://doi.org/10.1080/00401706.1979.10489755>
- [MES 15] Messac, A. (2015). Optimization in Practice with MATLAB®: For Engineering Students and Professionals. Cambridge University Press.
- [ROU 12] Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging , DiceOptim : Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal Of Statistical Software* Volume, 51(1), 1–55. <https://doi.org/10.1359/JBMR.0301229>
- [SAC 12] Sacks, J., Welch, W. J., Mitchell, T. J., Wynn, H. P., Welch, J., & Toby J. Mitchell. (2012). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4), 409–423. <https://doi.org/10.1214/ss/1177012413>
- [SEE 4] Seeger, M. (2004). Gaussian Processes for Machine Learning. *International Journal of Neural Systems*, 14(2), 69–106. <https://doi.org/10.1142/S0129065704001899>
- [SHE 87] Shewry, M. C., & Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics*, 14(2), 165–170. <https://doi.org/10.1080/02664768700000020>
- [STE 87] Stein, M. (1987). Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*, 29(2), 143–151. <https://doi.org/10.2307/1269769>
- [WU 12] Wu, Y., Wang, H., Zhang, B., & Du, K.-L. (2012). Using Radial Basis Function Networks for Function Approximation and Classification. *ISRN Applied Mathematics*, 2012, 1–34. <https://doi.org/10.5402/2012/324194>