
TP cours 1 : Mise en place des TP

Ecole-chercheur Mexico, La Rochelle

V. Picheny - N. Durrande

Cette session est dédiée à la mise en place du cas-test “piton de la fournaise” et à une première exploration à l’aide de différents plans d’expériences.

1 Description et installation du cas-test

Se référer au document `volcan_test_case.pdf`.

2 Génération de différents plans d’expériences

Les objectifs de cette section sont 1- d’observer différentes caractéristiques de plans d’expériences et 2- de générer un premier jeu de données pour le cas-test qui servira dans les TPs suivants.

Q1. Ecrire une fonction qui implémente un plan d’expériences uniforme (à l’aide des fonctions `R runif, matrix`). Cette fonction doit prendre en paramètres le nombre de points n et le nombre de variables d , et retourner une matrice $n \times d$ de valeurs entre 0 et 1. Générer un plan à 100 points en dimension 5. Visualiser.

Q2. Ecrire une fonction qui implémente un hypercube latin (aléatoire) : on pourra utiliser la fonction `sample`. Cette fonction utilisera les mêmes entrées / sorties que la précédente. Attention à la normalisation. Cette question peut être sautée par les stagiaires souhaitant directement explorer le cas-test. Générer un plan à 100 points en dimension 5. Visualiser.

Q3. A l’aide du paquet `DiceDesign`, générer un hypercube latin optimisé à 100 points et 5 variables, par exemple avec la fonction `maximinESE_LHS`, ou bien avec le paquet `lhs` et la fonction `improvedLHS`.

Q4. A l’aide du paquet `DiceDesign`, générer 100 points d’une suite à faible discrédance en dimension $d = 5$ à l’aide de `runif.faure`.

Q5. Comparer les 4 plans d'expériences générés. On pourra regarder les répartitions sur les marginales d'ordre 1 à l'aide d'histogrammes (`hist`) ou de graphes en bâton (`plot(seqce, rep(1, N), type="h")`), les répartitions sur les espaces bi-dimensionnels (fonction `pairs`) et encore des métriques de remplissage d'espace (`minDist`, `meshRatio` de `DiceDesign`).

Q6. Choisir un des plans et effectuer les appels au modèle de volcan correspondant. A noter : on a uniquement besoin d'utiliser la fonction `compute_wls`, qui accepte en entrée un vecteur ou une matrice.

Q7. Etudier les données générées. On pourra regarder en particulier comment la sortie du modèle varie en fonction de chaque paramètre d'entrée. Pour un plan d'expériences `X` et un vecteur de réponse `Y` : `plot(X[, 1], Y)`, etc.