

## TD 3 – Planification d'expériences numériques

Bertrand Iooss  
Polytech Nice Sophia

### 1. Plan maximin

- a) Créez un plan d'expériences maximin de 9 points et 2 variables  $U[0,1]$  (matrice de 9 lignes à 2 colonnes). L'idée est de générer un grand nombre de plans aléatoires et de prendre le meilleur au sens de la maximisation de la distance minimale entre 2 points du plan (critère dit maximin). La fonction codant cette distance se trouve dans le package « [DiceDesign](#) ». Visualisez ce plan par rapport à un plan aléatoire pur (Monte Carlo).
- b) Construisez un plan factoriel complet (2 variables et 9 points) en utilisant la fonction *factDesign()* du package « [DiceDesign](#) ». Visualisez ce plan par rapport aux 2 autres et comparez leur critère maximin.
- c) Quelles sont les problèmes que l'on entrevoit pour l'exploration d'un modèle numérique avec les plans factoriels complets ?

### 2. Plan à discrépance faible

- a) Construisez une suite de Sobol de 9 points et 2 variables  $U[0,1]$  en utilisant la fonction adéquate du package « [randtoolbox](#) ». Visualisez ce plan et comparez son critère maximin par rapport au plan maximin précédent.
- b) Calculez le critère de discrépance centrée de ce plan et des 3 plans du 1.1 (maximin, Monte Carlo, factoriel complet) et comparez-les. On utilisera la fonction adéquate du package « [DiceDesign](#) ».
- c) Construisez une suite de Halton (package « [randtoolbox](#) ») de 200 points et 8 variables  $U[0,1]$  (matrice de 200 lignes et 8 colonnes). Visualisez tous les scatterplots de ce plan avec la fonction *pairs()*. Quels types d'anomalies détectez-vous ? Quel problème cela pose-t-il en termes d'exploration de l'espace ?
- d) Répétez c) avec une suite de Sobol.
- e) Question optionnelle : Un des problèmes des suites à discrépance faible est la création d'alignements de points dans les plans de projection des différentes variables d'entrée. La technique du radar plot permet de détecter les alignements de points pour les sous-projections de dimension 2 en scannant toutes les directions et en y réalisant un test statistique d'uniformité des projections. Cette technique est disponible via la fonction *rss2d()* du package « [DiceDesign](#) ». Appliquez là sur la suite de Sobol. Que constatez-vous ?

### **3. Plan Hypercube Latin (LHS)**

- a) A l'aide de votre support de cours, construisez une fonction permettant de générer des LHS à marge uniforme.
- b) Construire un LHS de 20 points et 2 variables U[0,1].
- c) Construire deux LHS maximin (c'est-à-dire un LHS optimisé avec le critère mindist) avec deux méthodes différentes :
  - a. Générer un grand nombre de LHS et prendre le meilleur au sens maximin (comme en 1.1 a) mais cette fois avec des LHS).
  - b. Utiliser la fonction *maximinSA\_LHS()* du package « [DiceDesign](#) ». Il s'agit de l'algorithme de recuit simulé expliqué dans le support de cours. L'utilisation de l'algorithme du recuit simulé nécessite le réglage délicat de plusieurs options qu'il faudra s'efforcer de bien régler. Utiliser les outils disponibles (cf. l'aide de la fonction) afin d'analyser la convergence de l'algorithme.

Comparez-les avec un LHS standard (celui du b)) et un plan aléatoire pur en terme de visualisation graphique et de critère maximin.