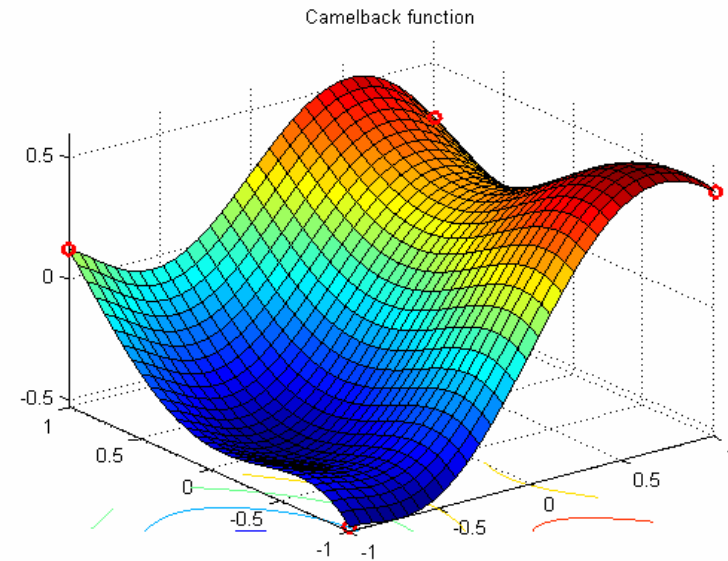


### III- Plans *remplissant l'espace* (*space-filling*)

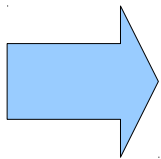
# Quel contexte ?

- Variables continues
- Pas d'information a priori sur la réponse (modèle non déterminé)
- Réponse potentiellement complexe (voire discontinue)
- Possibilité de variables peu influentes
- Nombre de facteurs pouvant être élevé



# Objectifs

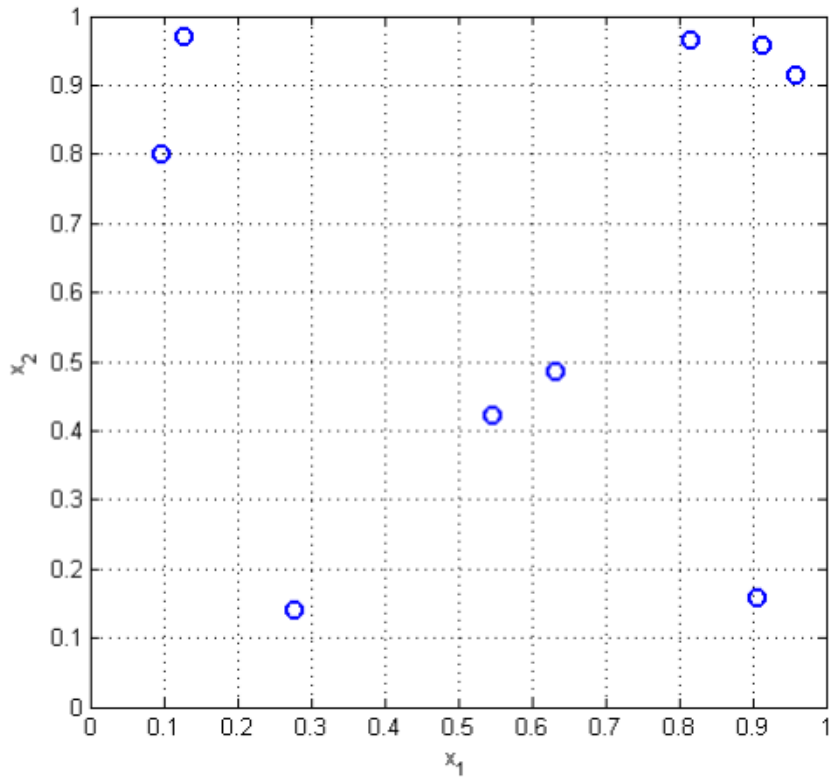
- Les plans doivent :
  - permettre d'utiliser une grande variété de modèles
  - donner de l'information pour n'importe quelle partie du domaine
  - avoir de bonnes projections dans les sous-espaces
  - être robustes à la montée en dimension



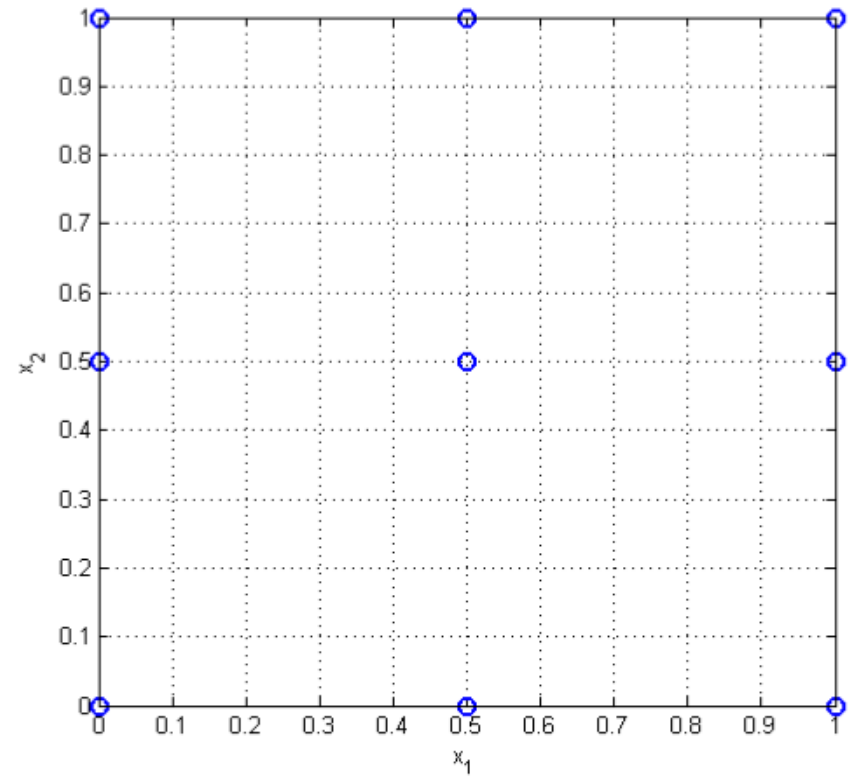
« **Remplir l'espace** » : explorer au maximum l'espace des variables

# Deux contre-exemples

- Plan aléatoire

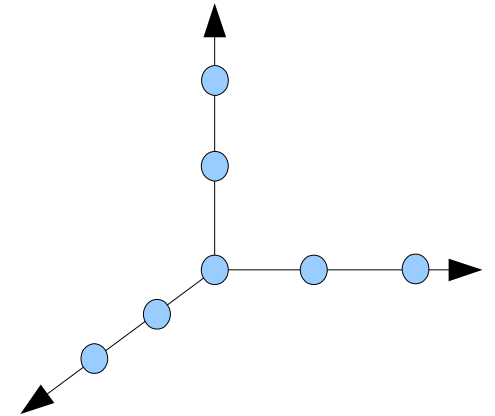


- Plan factoriel

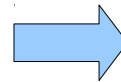
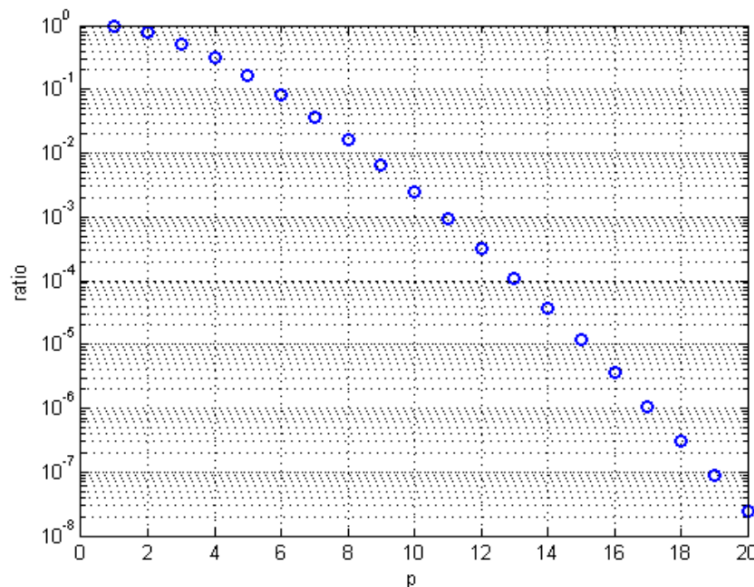


# Curiosités de la montée en dimension (1/2)

- Plan en étoile à 3 niveaux
- Inclus dans la boule de rayon 0,5
- Volume de la boule :  $V = \frac{r^d \pi^{d/2}}{\Gamma(d/2 + 1)}$



- Ratio hypersphère /hypercube :

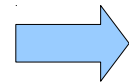
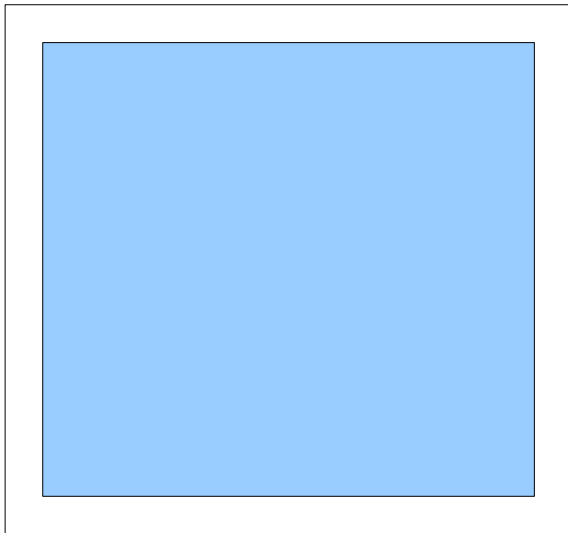


Pour d grand,  
on explore 0%  
du volume !

# Curiosités de la montée en dimension (2/2)

- Rapport entre le cube de côté 1 et celui de côté 0,9 :

$$V_1/V_2 = 0,9^p \rightarrow 0$$

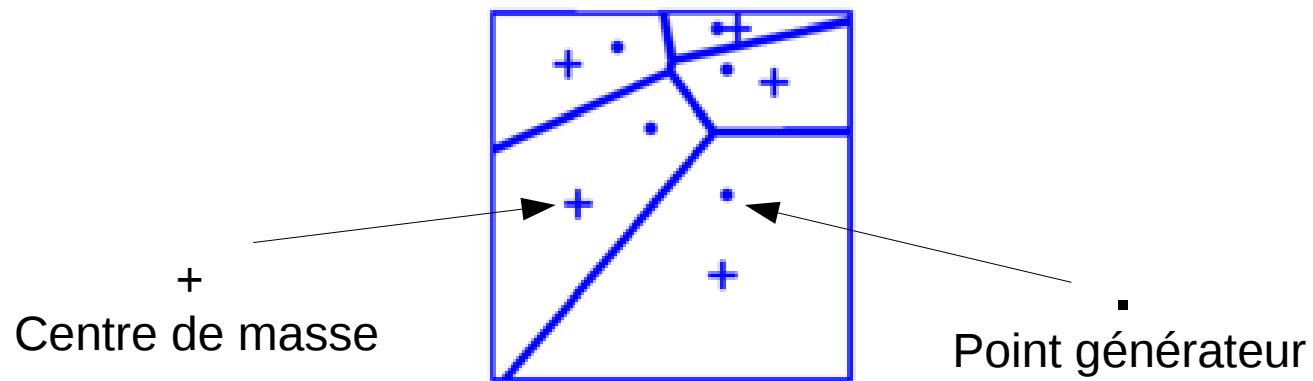


Tout le volume est contenu dans l' «écorce »

# Tessellations Centroïdales de Voronoï (CVT)

# Définitions

- Tessellation : découpage de l'espace
  - Région de Voronoi : définie par des points générateurs
  - Chaque région possède un centre de masse
  - CVT : les points générateurs sont les centres de masse
- ➡ Cas très particulier des VT !

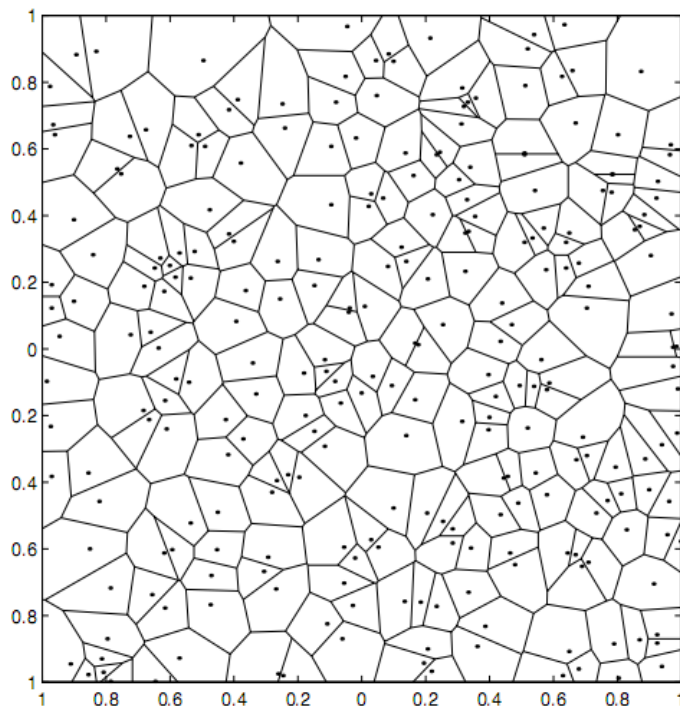




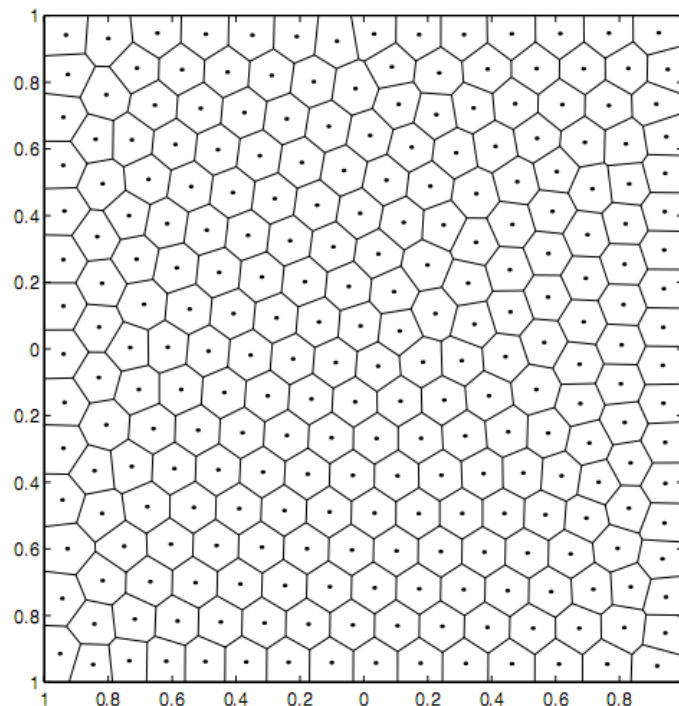
# Propriétés des CVT (1/2)

Les CVT proposent un découpage homogène de l'espace

uniform density      256 generators



Random sampling



Centroidal Voronoi

*Source :  
M. Gunzburger*

# Propriétés des CVT (2/2)

- Distortion :  $H(z_i, V_i) = \sum_{i=1}^k \int_{V_i} \|y - z_i\|^2 dy$
- Quadrature :  $\int_{\Omega} f(y) dy \approx \sum_{i=1}^k \int_{V_i} f(z_i) dy = \sum_{i=1}^k A_i f(z_i)$   
avec  $A_i = \text{volume}(V_i)$

Pour  $f$  lipschitzienne :  $Q = \left| \int_{\Omega} f(y) dy - \sum_{i=1}^k A_i f(z_i) \right| \leq L \sum_{i=1}^k \int_{V_i} \|y - z_i\| dy$

- Les CVT sont stationnaires pour :  $\sum_{i=1}^k \int_{V_i} \|y - z_i\|^p dy, \quad p \geq 1$



On a une forme d'optimalité !

# CVT et planification

- Distortion minimale : chaque point de l'espace est proche d'un point générateur
  - Bonne approximation constante par morceaux
  - Propriétés indépendantes de la dimension de l'espace
- 
- ➡ Bon remplissage d'espace
  - ➡ On utilise les **points générateurs** comme **plan d'expériences**

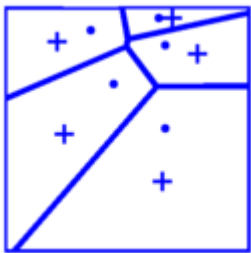
# Construction des CVT (1/2)

- Algorithme de Lloyd (point fixe) :

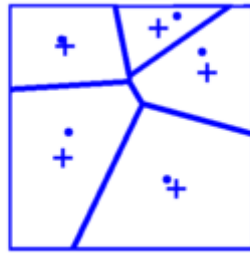
1. Générer un plan initial
2. Construire la tessellation de Voronoï correspondante
3. Déterminer les centres de masses
4. Remplacer les points générateurs par les centres

➡ Chaque itération est très coûteuse !

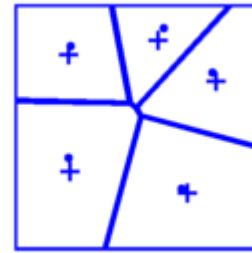
1



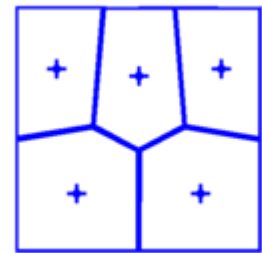
2



3



15



# Construction des CVT (2/2)

- Algorithme stochastique : MacQueen

1. Générer un plan initial, initialiser  $j_i = 1$

2. Générer aléatoirement un point  $w$

3. Chercher le point générateur  $z_k$  le plus proche

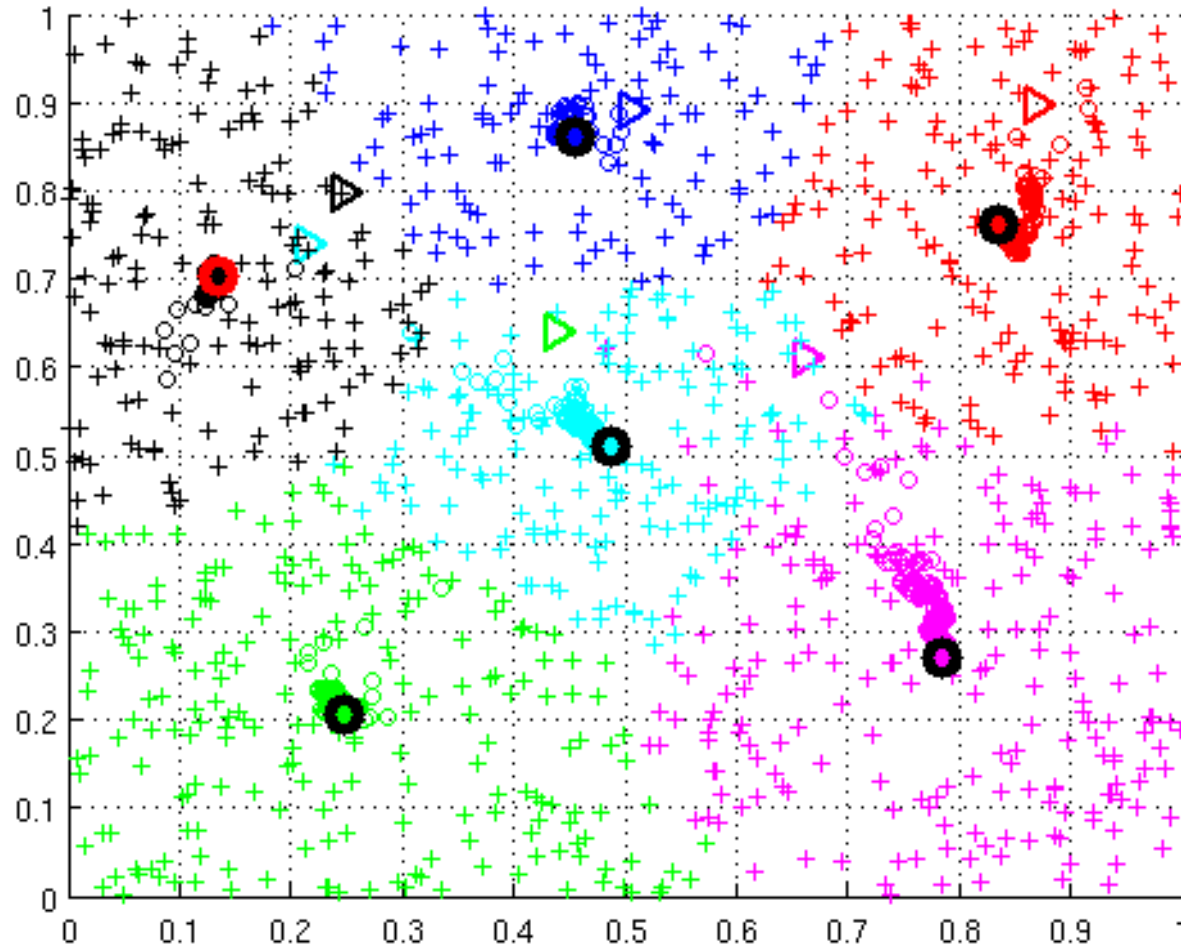
4. Mettre à jour  $z_k = \frac{j_k z_k + w}{j_k + 1}$

5.  $j_i = j_i + 1$

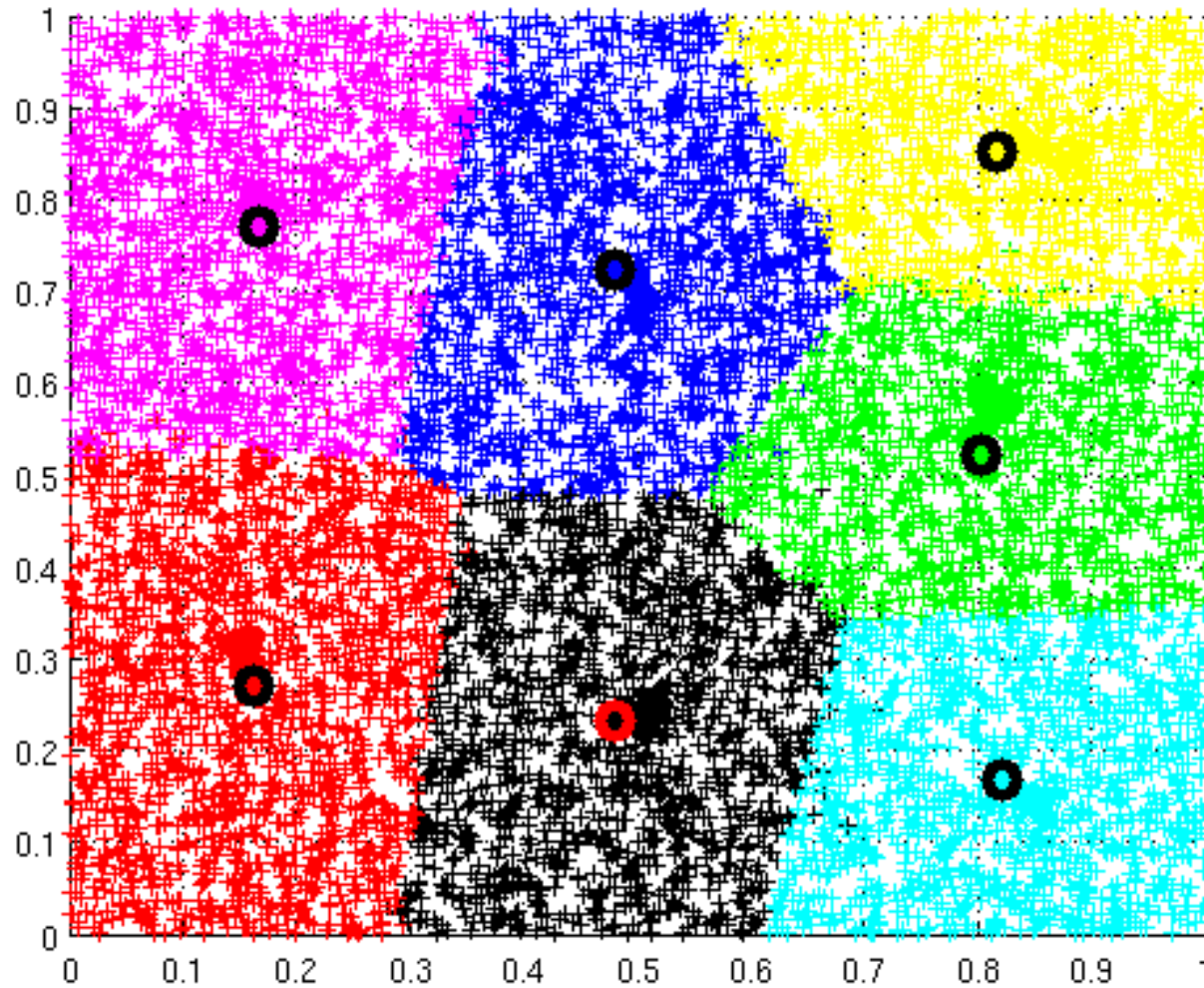
- Itération peu coûteuse
- Convergence en loi
- Liens avec l'analyse discriminante (*k-means clustering*)
- Nécessite des millions d'itérations

# Algorithme de McQueen, 6 pts, 1000 itérations

- › Triangles : points initiaux
- › Croix : points générés
- › Ronds : centres de masse



# Algorithme de McQueen, 7 pts, 10000 itérations

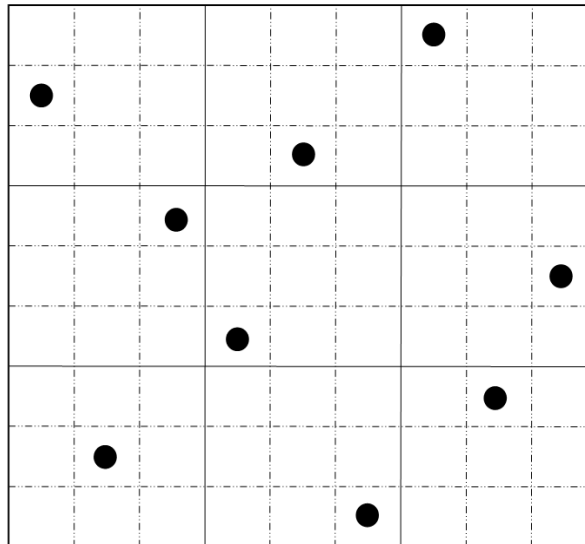


# Les Hypercubes latins (LHS)



# Hypercubes latins (LHS)

- Plans **aléatoires**
  - On découpe chaque dimension en N intervalles
  - On prend un unique point par intervalle
- ➡ Les distributions marginales sont uniformes !



Un plan LHS à 9 points

# Exercice

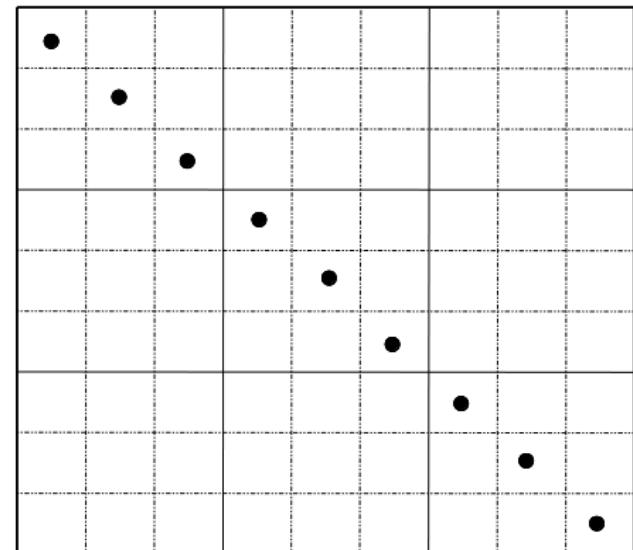
- Construire un LHS à 5 points en dimension 3

# Exercice

- Construire un LHS à 5 points en dimension 3
- Définition :  
*Un hypercube latin à  $n$  points et  $s$  variables,  $LHD(n,s)$ , est une matrice  $n \times s$  dont chaque colonne est une permutation de  $\{1, 2, \dots, n\}$*

# Avantages et inconvénients

- Très faciles à générer
- Pas de difficulté pour  $n$  et  $d$  grands
- Pas de duplication en projection
- Ne remplit pas l'espace !



Plans optimisés

# Critères basés sur les distances

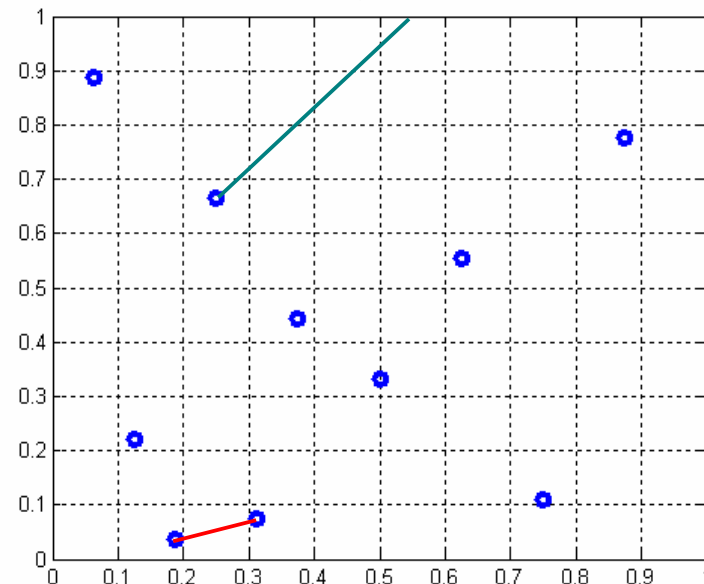
Critères de comparaison ou d'optimisation :

- **Maximin**

*(maximiser la) distance minimum entre deux points du plan*

- **Minimax**

*(minimiser la) distance maximale entre n'importe quel point du domaine et le point du plan le plus proche*



# Exercice

1. Formaliser les critères *maximin* et *minimax*
2. Ecrire les problèmes d'optimisation de plans d'expériences correspondants

# Les problèmes d'optimisation

- Maximin :  $\max_{x_1, \dots, x_N} \left[ \min_{i \neq j} (d(x_i, x_j)) \right]$

$kN$  dimensions

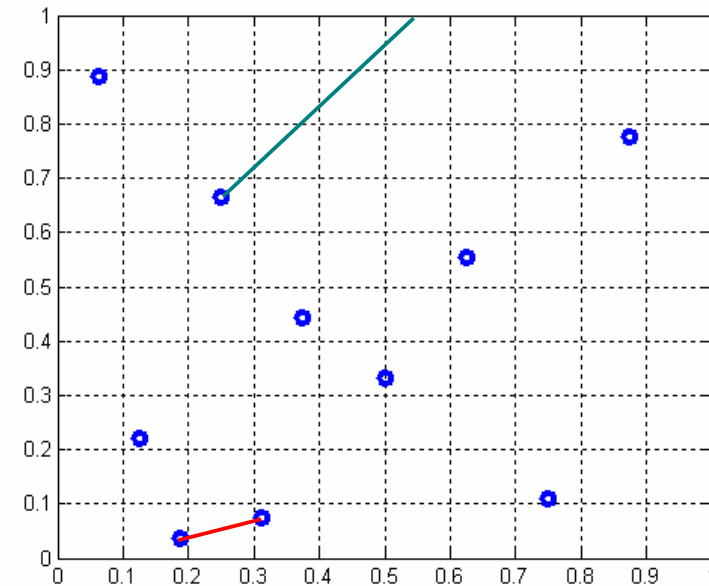
$\frac{n(n-1)}{2}$  distances

- Minimax :  $\min_{x_1, \dots, x_N} \left[ \max_x \left[ \min_i (d(x, x_j)) \right] \right]$

$dN$  dimensions

optim en dimension  $d$

$n$  distances



➡ Minimax est très coûteux

➡ Les problèmes d'optimisation sont très complexes !



# Optimisation

- Le problème général d'optimisation n'est pas soluble :
  - On recherche les  $d$  coordonnées de  $N$  points, soit un problème à  $d \times N$  paramètres !
  - Le critère *minimax* est très coûteux à évaluer
- Solution :
  - Utiliser le critère *maximin*
  - Optimiser une classe de plans, par ex. LHS

# Optimisation : algorithmes

- Problème complet (*maximin* seulement) : optimiseur global

→ voir le cours !

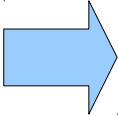
- **Algorithme d'échange :**

1. On construit (aléatoirement ou non) un plan initial
2. On choisit (simultanément ou non) un point à enlever au plan et un point à rajouter
3. On effectue l'échange
4. On répète 2 et 3 jusqu'à satisfaction du critère d'arrêt

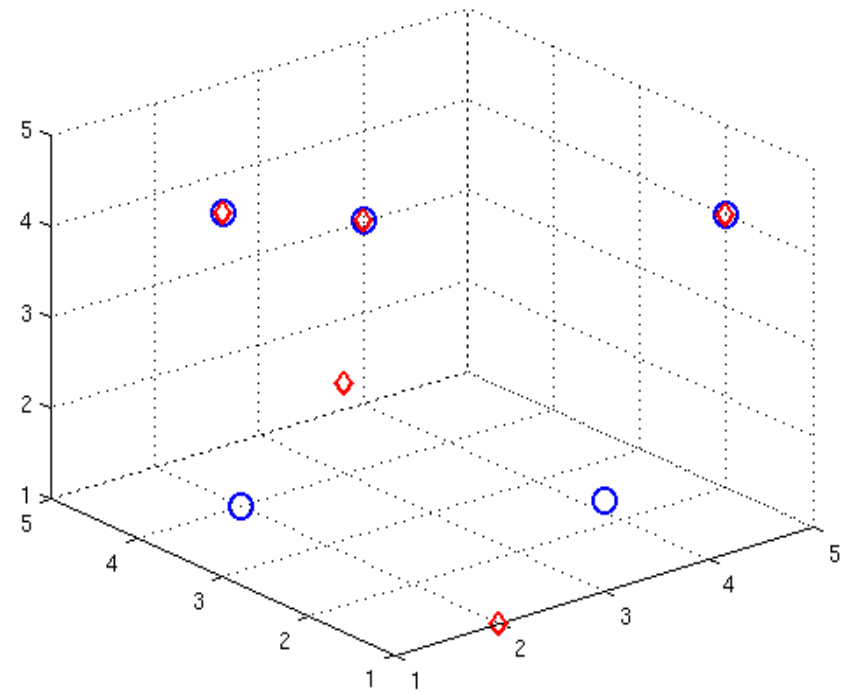
# Optimisation des LHS (1/3)

- Problème discret !
- On peut permuter 2 cellules (ou blocs) d'une même colonne

A	B	C
1	3	5
3	1	2
4	5	3
2	4	1
5	2	4



A	B	C
1	3	5
3	4	2
4	5	3
2	1	1
5	2	4



# Optimisation des LHS (2/3)

- Algorithme de recuit simulé [Morris et Mitchell, 1995]:
  1. On génère un plan initial
  2. On cherche les points critiques (au sens du critère *maximin*)
  3. On choisit aléatoirement une colonne d'un point critique
  4. On effectue un échange avec une autre cellule de la même colonne prise aléatoirement
  5. Si l'échange est bénéfique : on accepte la modification
  6. Sinon : on accepte (quand même !) avec une probabilité

$$\pi = \exp\left[-\left(\frac{\phi(PEX_{nouveau}) - \phi(PEX_{ancien})}{T}\right)\right]$$

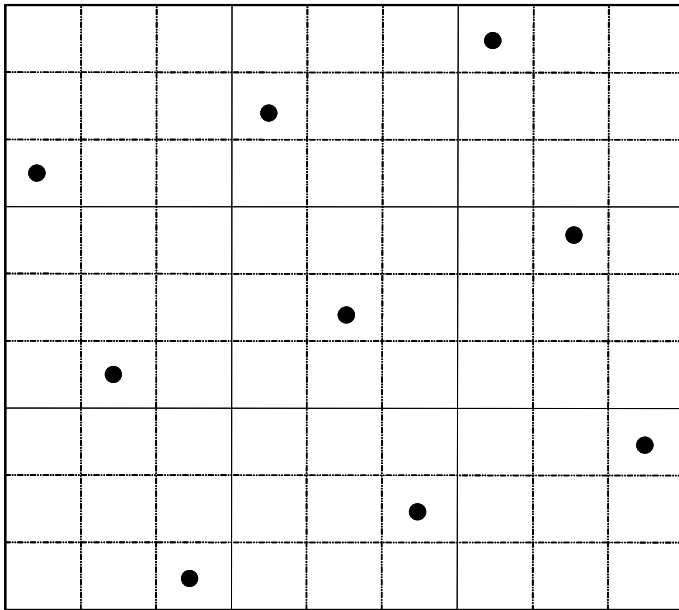
T diminue avec le temps

# Optimisation des LHS (3/3)

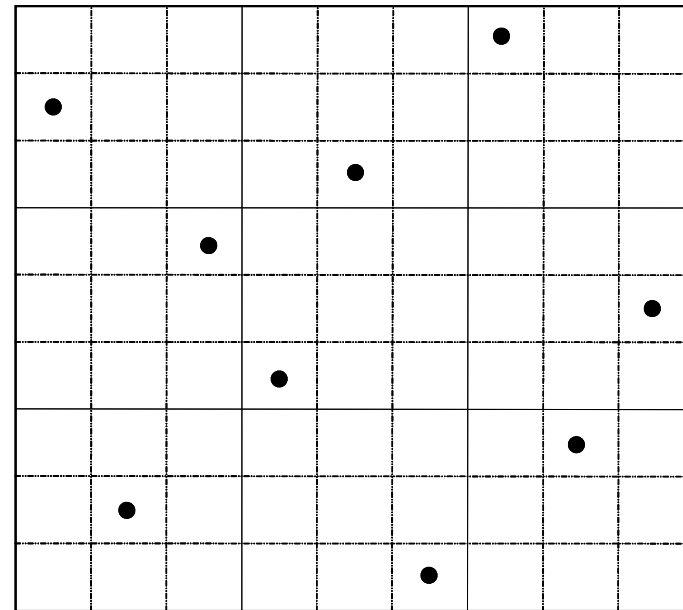
- Remarques :
  - T diminue avec le temps : on accepte moins facilement les changements
  - Acceptation d'un point moins bon : favorise l'exploration
- En pratique :
  - Difficile à régler !
  - Convergence très lente
- Nombreuses variantes / améliorations :
  - Choix des échanges moins aléatoire
  - Autres formes algorithmiques (ES)

# LHS maximin

- Le plan le plus utilisé pour les expériences numériques
- Plan à 9 points en 2D :



Plan optimisé (maximin)



Plan aléatoire

# Les suites à faible discr pance

# Une mesure de remplissage : la discrédance

- Discrédance : mesure de **non-uniformité**

*Ecart entre la distribution du plan et la distribution uniforme*

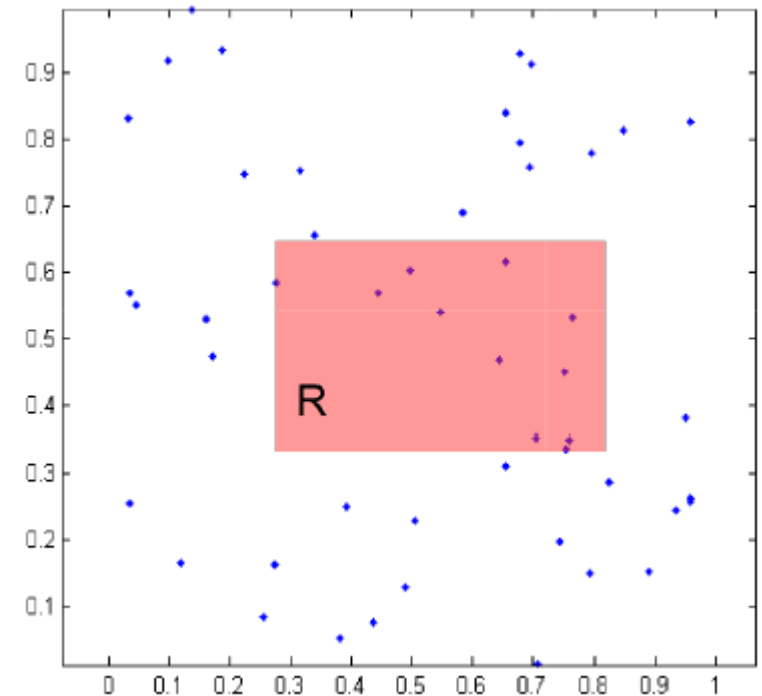
*Plus la discrédance est faible, plus on remplit l'espace*

- Pour un intervalle : comparaison entre le rapport de volumes et celui du nombre de points contenus

$$D = \max \left| \frac{N_R}{N} - \frac{V_R}{V} \right|$$

Discrédance L2 centrée :

$$\begin{aligned} \text{disc}_2(D) = & \left( \frac{13}{12} \right)^p - \frac{2}{N} \sum_{i=1}^N \prod_{k=1}^p \left( 1 + \frac{1}{2} \left| x_k^{(i)} - \frac{1}{2} \right| - \frac{1}{2} \left| x_k^{(i)} - \frac{1}{2} \right|^2 \right) \\ & + \frac{1}{N^2} \sum_{i,j=1}^N \prod_{k=1}^p \left( 1 + \frac{1}{2} \left| x_k^{(i)} - \frac{1}{2} \right| + \frac{1}{2} \left| x_k^{(j)} - \frac{1}{2} \right| - \frac{1}{2} \left| x_k^{(i)} - x_k^{(j)} \right| \right) \end{aligned}$$



Source : B. Iooss



# Les suites à faible discrédance (1/2)

- Suites équiréparties : une des trois conditions est vérifiée

- $\frac{1}{n} \sum_{i=1}^n f(x_i) \rightarrow \int f(x) dx$
- $\frac{1}{n} \sum_{i=1}^n 1_{f(x_i) \in [0, y]^d} \rightarrow \text{Vol}([0, y]^d)$
- $D_n \rightarrow 0$

- Il existe des suites équiréparties telles que :

$$\frac{1}{n} \left| \sum_i f(x_i) - \int f(x) dx \right| \leq c(f) \frac{\log(n)^d}{n}$$

 Suites à faible discrédance

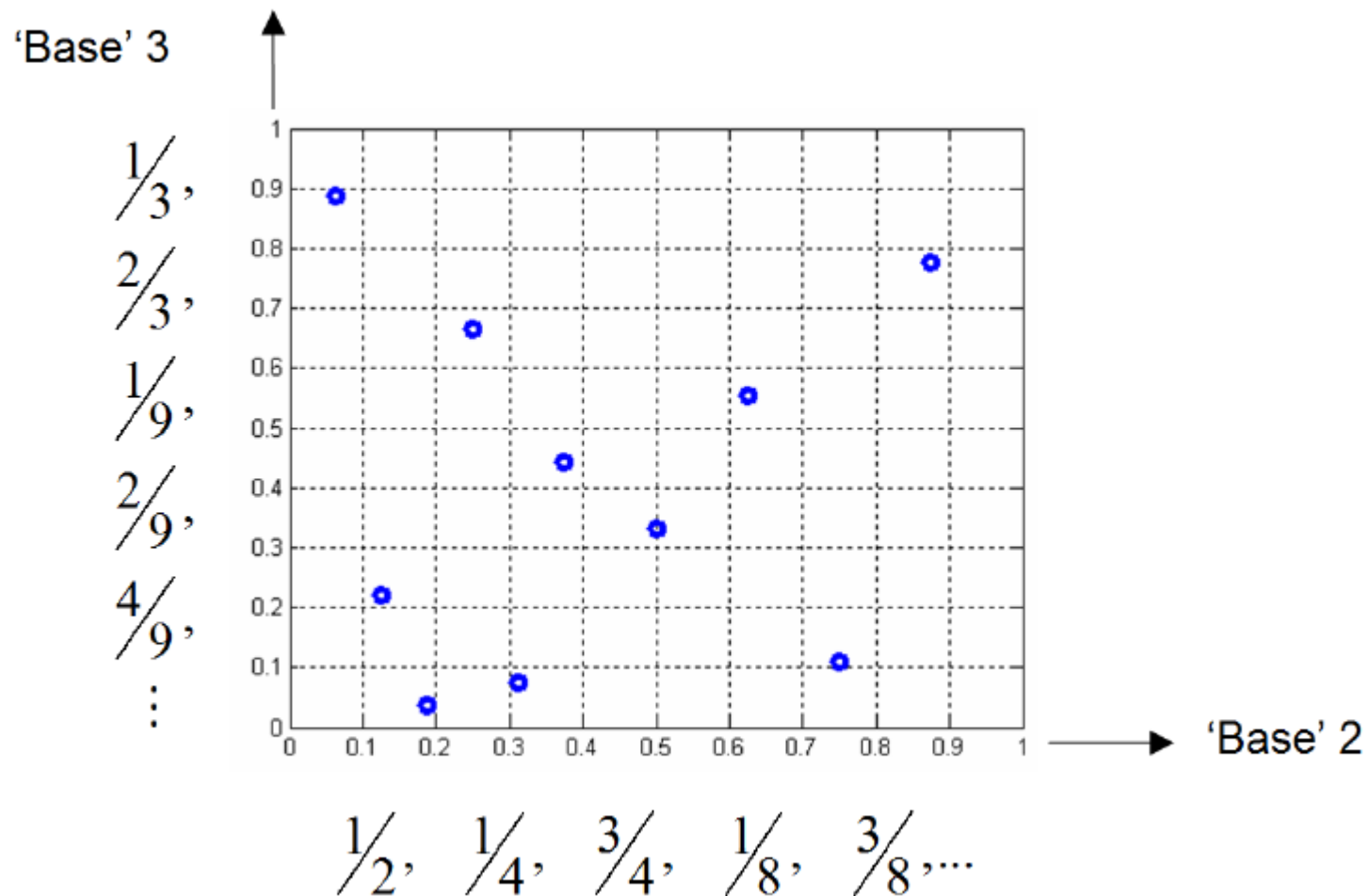
Tirage uniforme (Monte-Carlo) :  $c(f) \frac{1}{\sqrt{n}}$

Plan factoriel (grille) :  $c(f) \frac{1}{n}$

# Les suites à faible discrédance (2/2)

- Méthodes de *Quasi-Monte-Carlo*
- Faible discrédance : convergence plus rapide que la suite aléatoire uniforme
- De nombreuses suites existent :
  - Halton, Hammerley, Sobol', Faure, etc.
  - En général : suites déterministes
- Intérêts :
  - Bon remplissage d'espace
  - Rapides à construire
  - Flexibles (ajout séquentiel)

# Exemple : suite de Halton

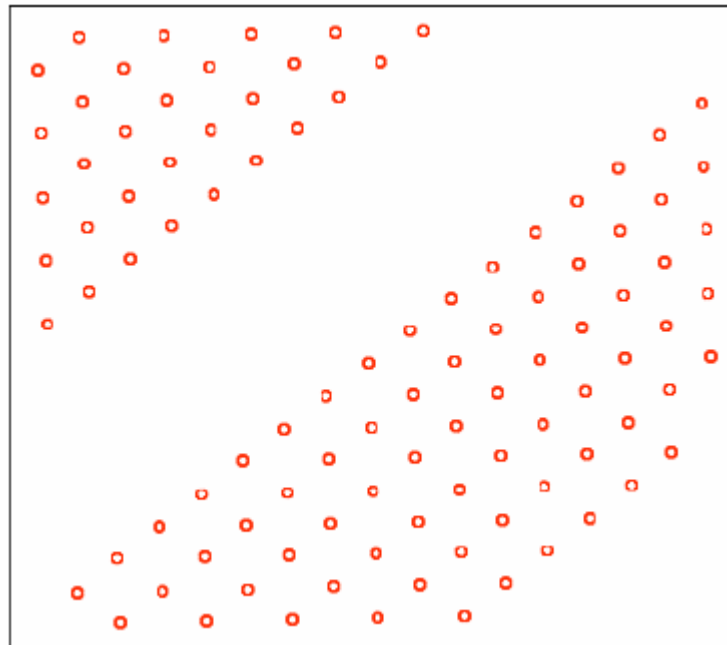


# Limites des suites à faible discrédance

Défauts en grande dimension :

- Alignements en projections
- 'Trous' dans les sous-espaces
- Déterminisme

Halton en 8 dimensions, 80 points : (X7, X8)



*Source :  
J. Franco, 2008*

# Récapitulatif : remplissage d'espace

- Objectif : exploration du maximum de l'espace à moindre coût
- Plusieurs familles de plans :
  - Suites à faible discrépance
  - CVT
  - LHS
- Plusieurs critères de « qualité » :
  - Maximin
  - Minimax
  - Discrépance
  - Projection
- Pas de solution universelle