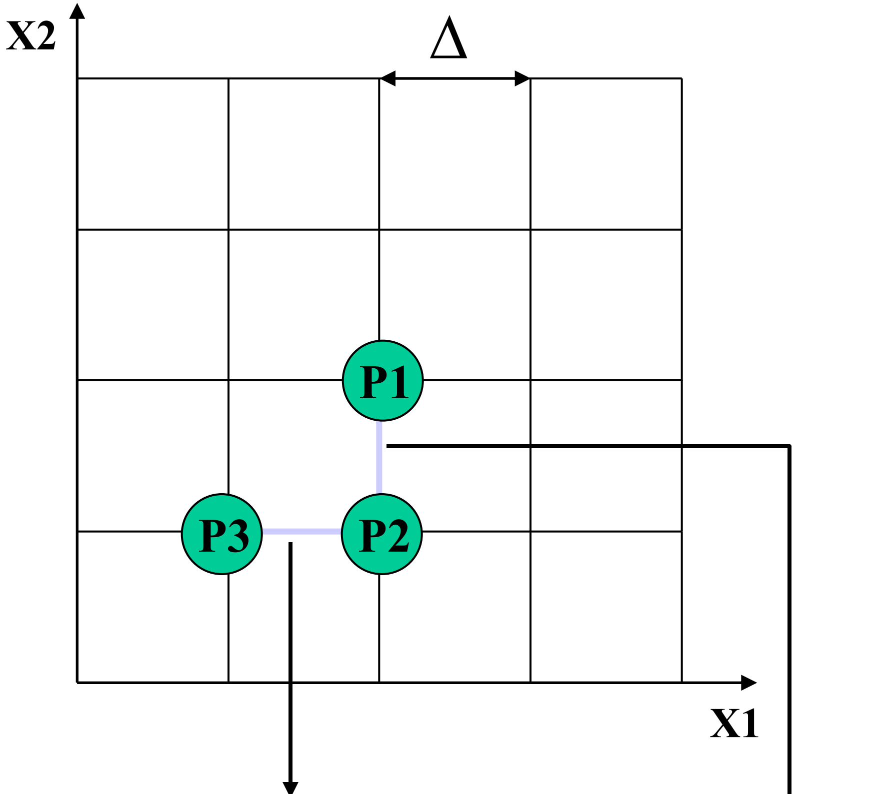
TD C

Space filling designs in R

CHANGER L'ÉNERGIE ENSEMBLE

18/05/2011

Typical engineering practice : One-At-a-Time (OAT) design



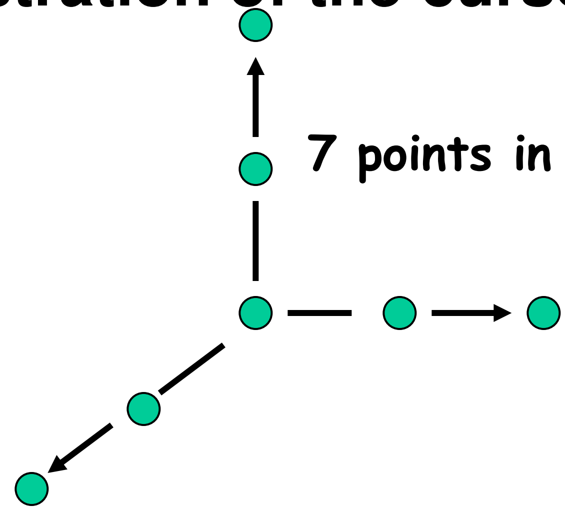
$$d_{X_1} = \frac{f(P3) - f(P2)}{\Delta} \qquad d_{X_2} = \frac{f(P2) - f(P1)}{\Delta}$$

Main remarks :

OAT brings some information, but potentially wrong

Exploration is poor : Non monotonicity ? Discontinuity ? Interaction ?

Illustration of the curse of dimensionality

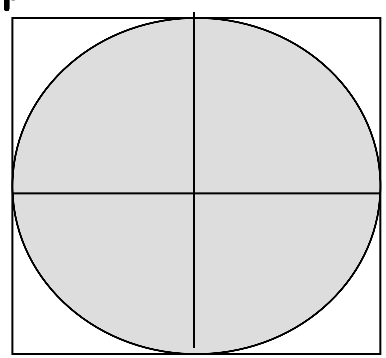


7 points in a 3D space

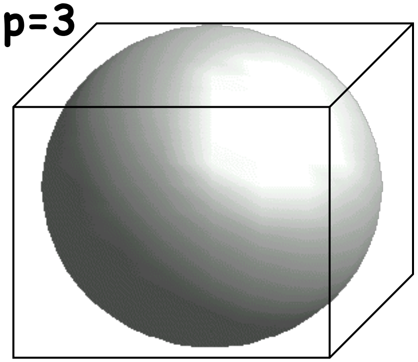
Bad space covering

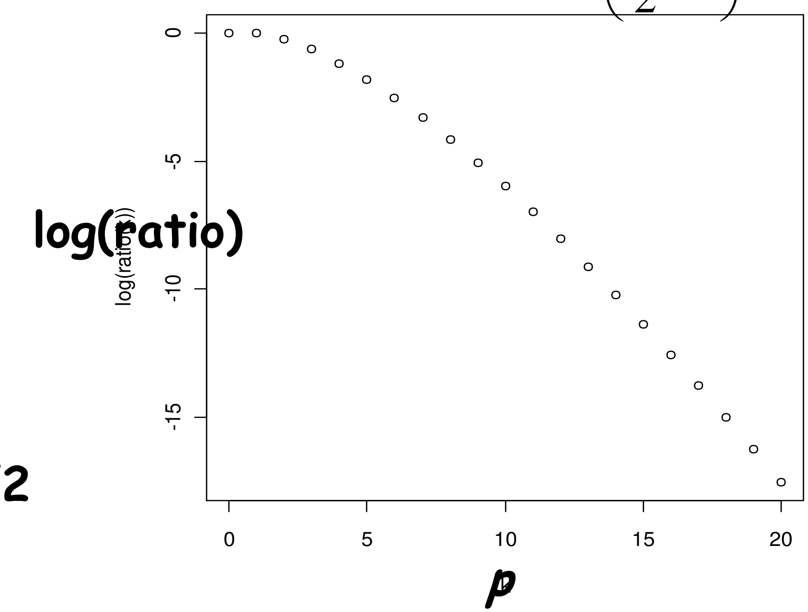
$$\text{vol. sphere}(r=0.5) = \frac{\pi^{p/2}}{\Gamma\left(\frac{p}{2}+1\right)}\left(\frac{1}{2}\right)^p$$

p=2



p=3





Surf. circle / Surf. square ~ 3/4

Vol. sphere / Vol. cube ~ 1/2

p=10 → Ratio ~ 0.0025

hypercube volume >> (included and tangent) hypersphere volume

For large dimensions, all the points will be in the corner of the hypercube

Model exploration goal

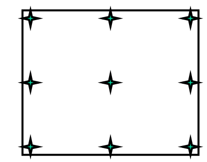
GOAL : explore as best as possible the behaviour of the code

Put some points in the whole input space in order to « maximize » the amount of information on the model output

Contrary to an uncertainty propagation step, it depends on p

Regular mesh with n levels $\longrightarrow$ $N = n^p$ simulations

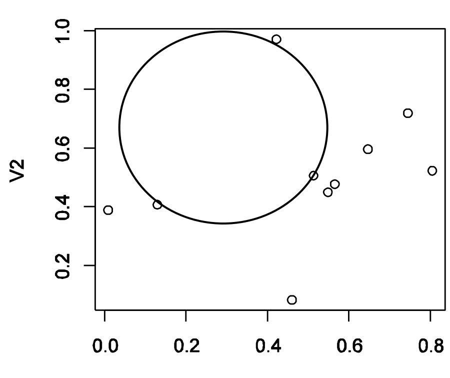
Ex: $p = 2$, $n = 3$
$\longrightarrow$ $N = 9$
$p = 10$, $n = 3$
$\longrightarrow$ $N = 59049$



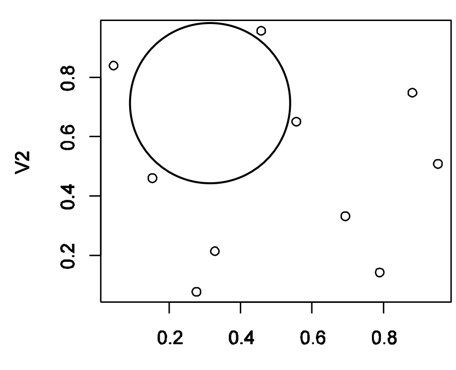
To minimize N, needs to have some techniques ensuring good « coverage » of the input space

Simple random sampling (Monte Carlo) does not ensure this

Ex: $p = 2$
$N = 10$



Monte Carlo



Optimized design

Exploration in physical experimentation

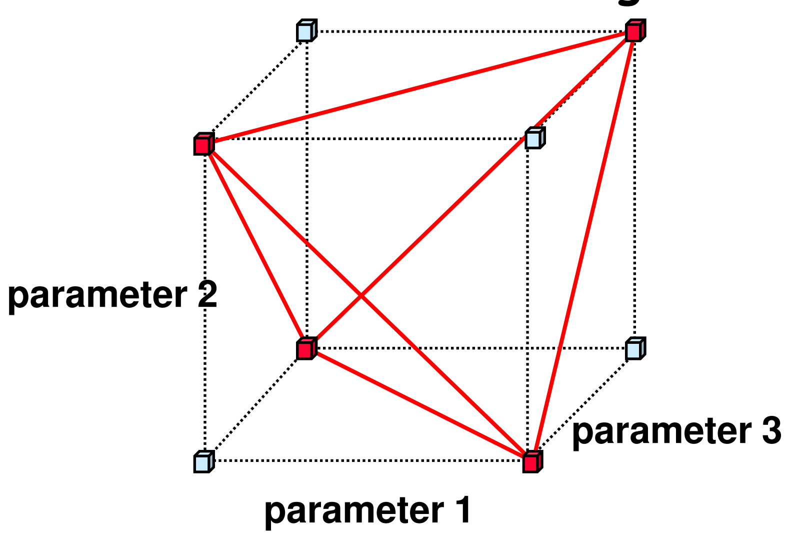
Design of experiments develops strategies to define experiments in order to obtain the required information as efficiently as possible

Designs for real experiments

Estimate parameters of linear regression with a minimal number of points

Examples :

- **Full factorial design 2^3**
- **Fractional factorial design 2^{3-1}**



Designs for numerical experiments

Characteristics

Deterministic experiments (no error),

Large number of input variables,

Large range of input variation domain,

Multiple output variables,

Strong interactions between inputs,

High non linearity in the model

→ space filling designs (uniform coverage in the input space)

Objectives

When the objectives is to discover what happens inside the model and when no model computations have been realized, we want to respect the two following constraints:

- To spread the points over the input space in order to capture non linearities of the model output,
- To ensure that this input space coverage is robust with respect to dimension reduction.

Therefore, we look some design whech insures the « best coverage » of the input space

- How to define this « best » ?
- How to choose the right number of points ?
- How to measure the representativity ?

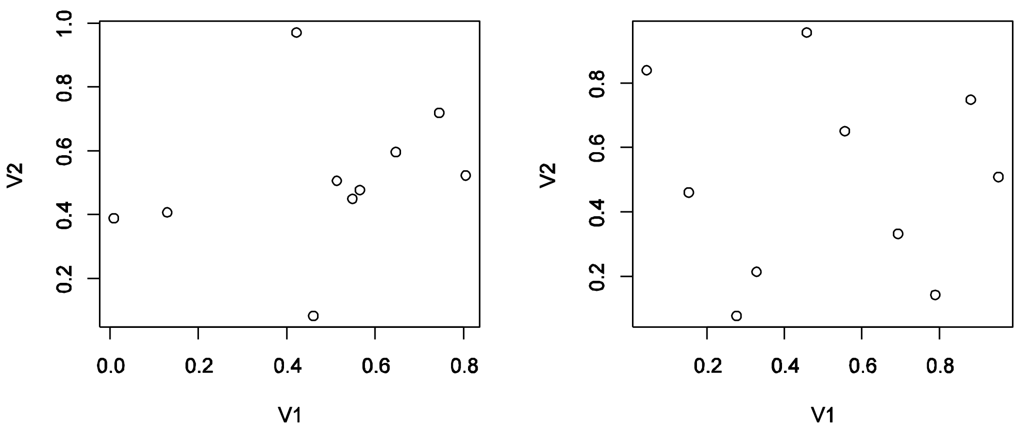
The design of numerical experiments: 1) Space filling

Sparsity of the space of the input variables in high dimension

The learning design choice is made in order to have an optimal coverage of the input domain

The space filling designs are good candidates.

Simple Random Sample (SRS)



Space Filling Design (SFD)

<u>Example:</u> Sobol sequence

<u>Two possible criteria:</u>

1. Distance criteria between the points: minimax, maximin, …
2. Uniformity criteria of the design (discrepancy measures)

Distance criteria between the points

[Johnson et al., 1990]

- Minimax design D_{MI} : Minimize the maximal distance between the points

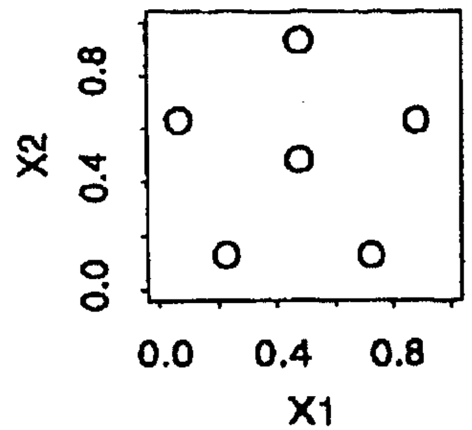
$$\min_D \max_x d(x, D) = \max_x d(x, D_{MI})$$

$$\text{where } d(x, D) = \min_{x^{(0)} \in D} d(x, x^{(0)})$$

All points in $[0,1]^p$ are not too far from a design point

=> One of the best design

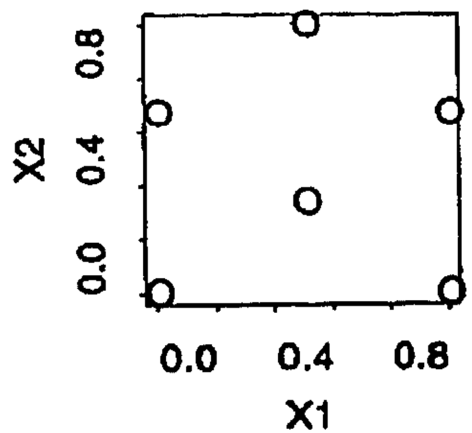
but too expensive to find D_{MI}



(a) Minimax

- Maximin design D_{MA} : Maximize the minimal distance between the points

$$\max_D \min_{x^{(1)}, x^{(2)} \in D} d(x^{(1)}, x^{(2)}) = \min_{x^{(1)}, x^{(2)} \in D_{MA}} d(x^{(1)}, x^{(2)})$$



(b) Maximin

[From: Owen, 1996]

Exercise 1

a) Build a design based on the maximin criterion

Characteristics : p = 2 variables U[0,1] ; N = 9 points

The idea is to generate a high number od random designs, then to select the best, by using the mindist() function of the DiceDesign package

b) Visualize this random maximin design with respect to a pure random design

c) Build a full factorial design, by using the factDesign() function of the DiceDesign package : 2 variables with 3 levels => 9 points

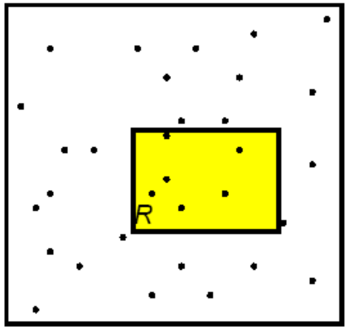
Visualize this design with respect the two others

Compare the maximin criteria of these 3 designs (random/maximin/factorial)

d) Identify the 2 problems related to the full factorial designs

Space filling measure of a design: the discrepancy

Measure of the maximal deviation between the distribution of the sample's points to an uniform distribution

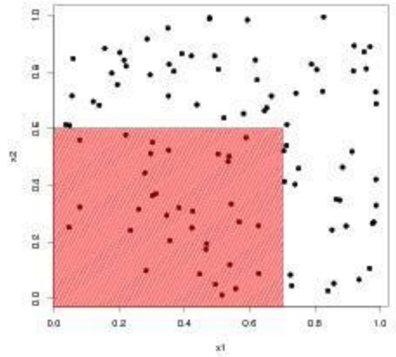


$\Rightarrow$ Measure of deviation from the uniformity

Geometrical interpretation:
Comparison between the volume of intervals and the number points within these intervals

$$Q(t) \in [0,1[^p, Q(t) = [0,t_1[\times[0,t_2[\times\ldots\times[0,t_p[$$

$$\mathrm{disc}(D) = \sup_{Q(t)\in[0,1[^p} \left| \frac{N_{Q(t)}}{N} - \prod_{i=1}^{p} t_i \right|$$



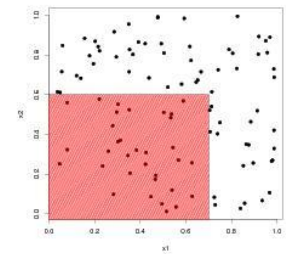
Lower the discrepancy is, the more the points of the design D fill the all space

Discrepancy computation in practice

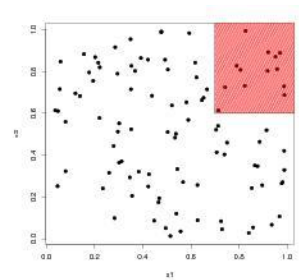
Different definitions, depending on the chosen norm & considered intervals

Classical choice (easy computations): L^2 – discrepancy

- **Modified L_2-discrepancy** (intervals with minimal boundary 0)

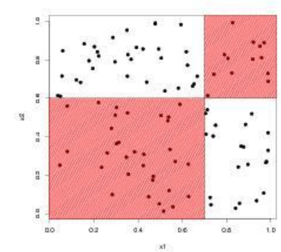


- **Centered L_2-discrepancy** (intervals with boundary one vertex of the unit cube)



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

- **Symetric L_2-discrepancy** (intervals with boundary one « even » vertex of the unit cube)



Relation with the integration problem

$$I = \int_{[0,1[^p} f(x)dx$$

$$\text{MonteCarlo} : I_N^{\text{MC}} = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)})$$

$$\text{with } \left(x^{(i)}\right)_{i=1\ldots N} \text{ a sequence of random points in } [0,1[^p$$

$$\text{E}\left(I_N^{\text{MC}}\right) = I \ ; \ \text{Var}\left(I_N^{\text{MC}}\right) = \frac{\text{Var}(N)}{N} \ \Rightarrow \varepsilon = O\left(\frac{1}{\sqrt{N}}\right)$$

General property: $\varepsilon \leq V(f) \times \text{disc}(D)$

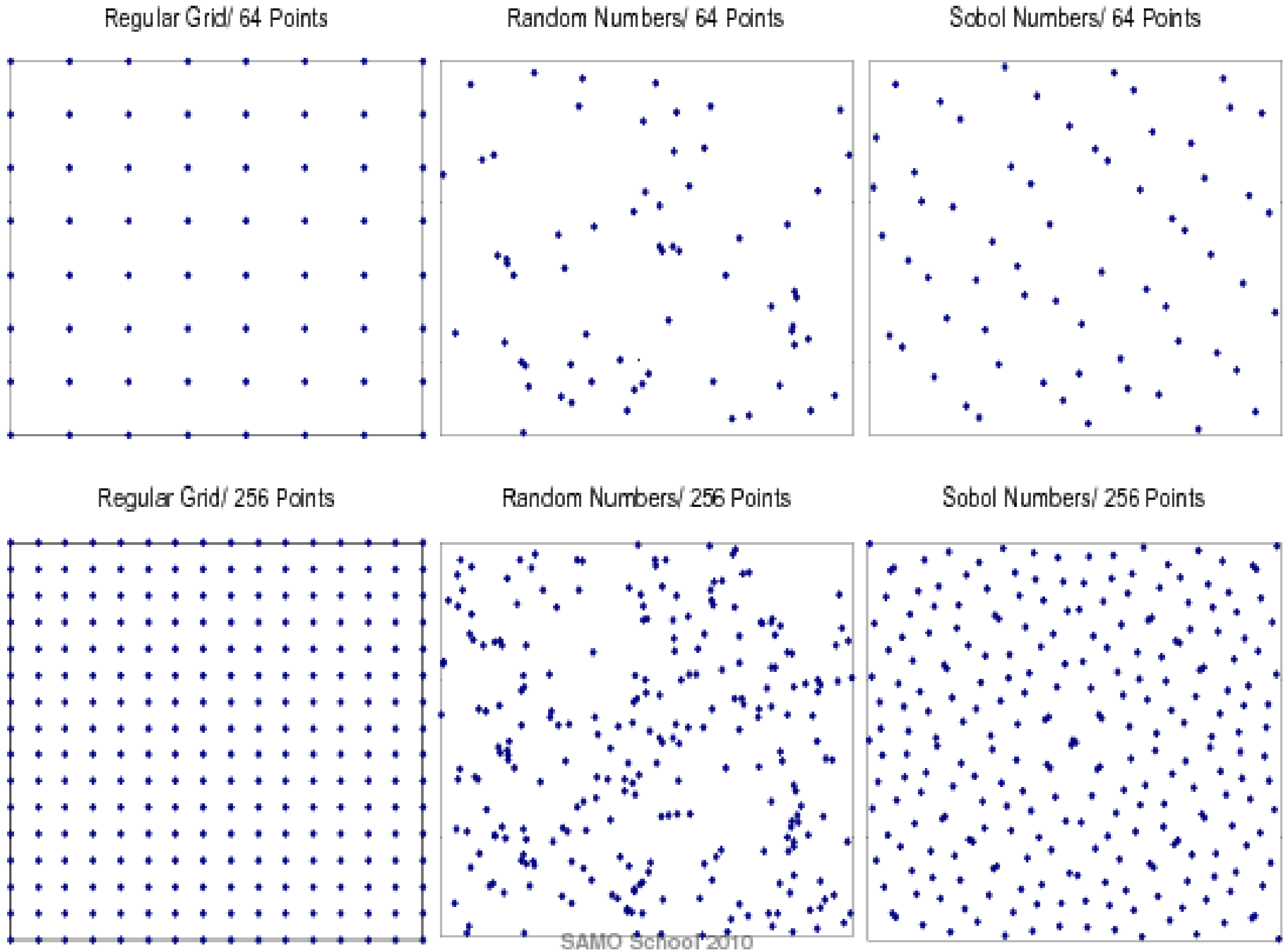
With a low discrepancy sequence *D* (quasi Monte Carlo sequence) :

$$\varepsilon = O\left(\frac{(\ln N)^p}{N}\right)$$

Well-known choice: Sobol' sequence

Sobol'sequence vs. Random sample vs. regular grid

[From: Kucherenko, 2010]



Exercise 2

a) Build a sequence of Sobol by using the sobol() function of the randtoolbox package

Characteristics : p = 2 variables U[0,1] ; N = 9 points

Visualize this design and compare its maximin criterion with those of exercise 1

b) Compute a discrepancy criterion of Sobol sequence and designs of exercise 1 by using the discrepancyCriteria() function of the DiceDesign package

c) Build a sequence of Sobol with p = 8 variables U[0,1] ; N = 200 points

Visualize all the scatterplots with the pairs() function

What kind of anomalies do you detect ?

PS: this problem is much more pregnant with Faure and Halton sequences

The design of numerical experiments: 2) LHS

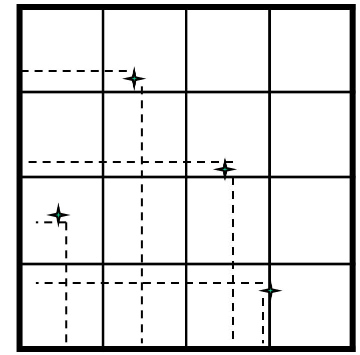
A lot of models are additive. If not, first order effects often dominate

➡ Property of *uniform projections on the margins*

It can be obtained via a Latin Hypercube Sample

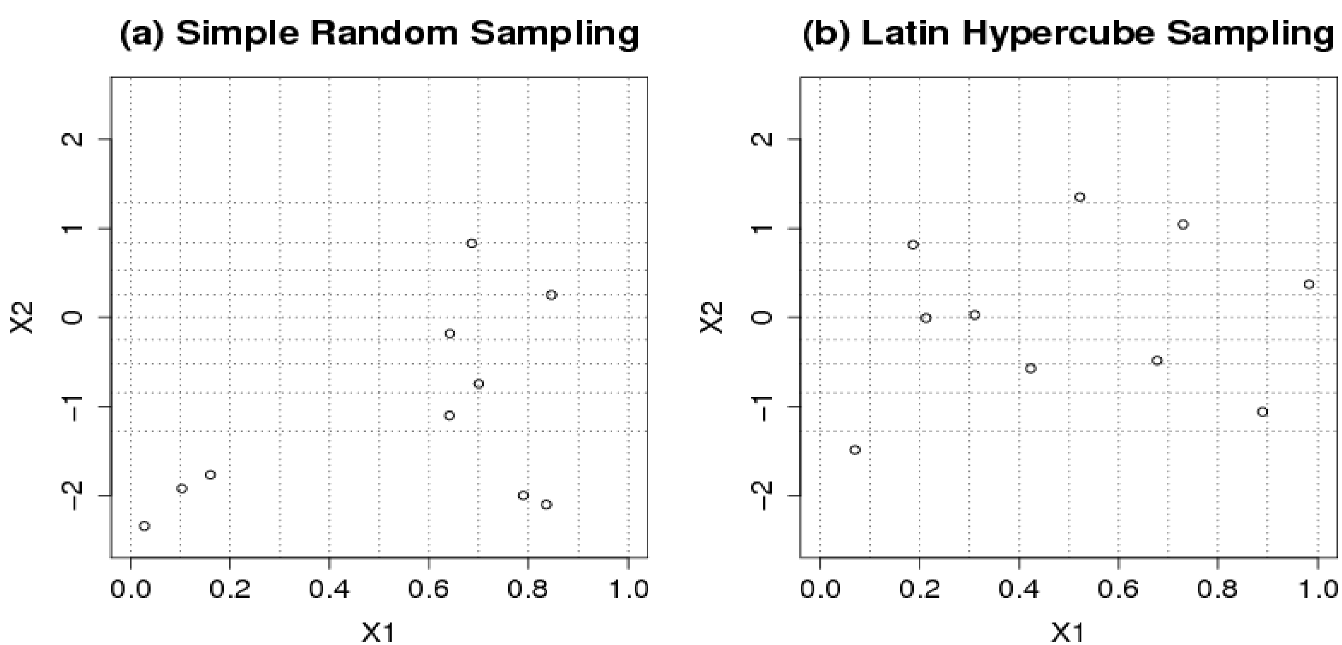
Divide each dimension in N intervals

Take one point in each stratum



Example : p =2, N =4

Example : p =2, N =10, X_1 ~ U[0,1], X_2 ~ N(0,1)



Algorithm of LHS – Stein method

Sample with N points of p inputs

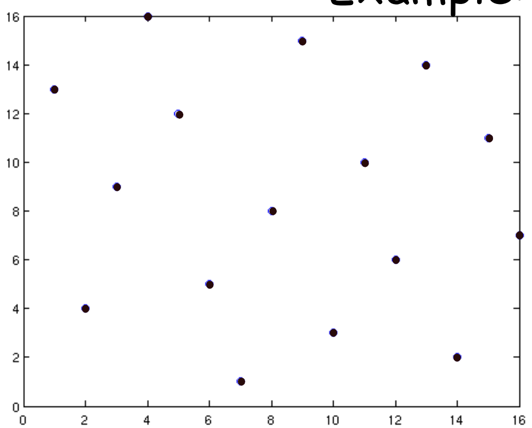
```
ran = matrix(runif(N*p),nrow=N,ncol=p)
                                    # tirage de N x p valeurs selon loi U[0,1]

for (i in 1:p)
{

     idx = sample(1:N)       # vecteur de permutations des entiers {1,2,…,N}

     P = (idx-ran[,i]) / N    # vecteur de probabilites

     x[,i] <- quantile_selon_la_loi (P)

}
```

Optimization of LHS

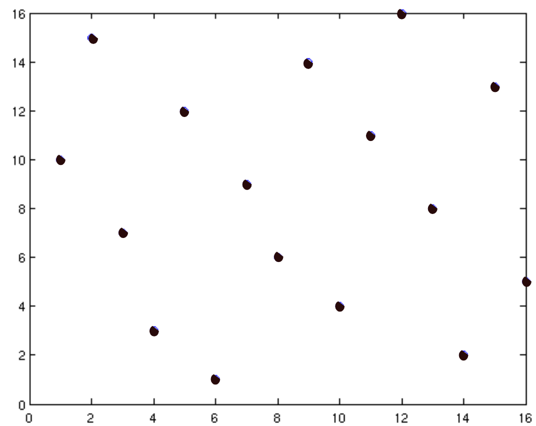
Joining the two properties (space filling and LHS)

One possibility: generate a large number (for ex: 1000) of different LHS
Then, choose the LHS which optimizes the criterion

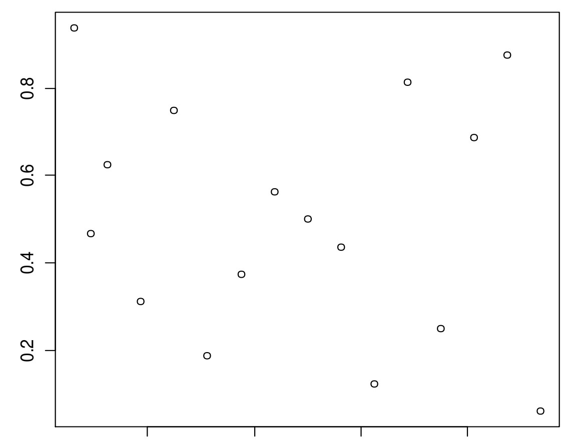
Example: $p = 2$ – $N = 16$



Maximin LHS



Low wrap-around discrepancy LHS



For comparison: Sobol sequence

Exercise 3

a) Build a random LHS and a maximin LHS by using the package lhs
Characteristics : p = 2 variables U[0,1] ; N = 20 points

Visualize these two designs and compare them with a pure random design

b) BONUS : sensitivity analysis

Look at the package sensitivity which allows to perform some global sensitivity analysis, and especially to obtain variance-based sensitivity indices

Run the example at the end of help page of sobol2002() (on Sobol *g*-function)

Replace the two initial independent Monte Carlo samples needed by sobol2002() by some independent Sobol'sequences ; then look at the results in terms of sensitivity estimates and errors

Exact 1st order indices : S1=0.716; S2=0.179; S3=0.024 ; S4=0.007 ; S5=0 ; S6=0 ; S7=0 ; S8=0
Exact total indices : ST1=0.786; ST2=0.241; ST3=0.034; ST4=0.010; ST5=0; ST6=0; ST7=0 ; ST8=0