

IFT 3245  
Simulation et modèles  
Introduction au quasi-Monte Carlo

Fabian Bastin  
DIRO  
Université de Montréal

Automne 2016

# Méthodes quasi-Monte Carlo (QMC)

(Tiré de Pierre L'Ecuyer)

On veut estimer

$$\mu = \int_{[0,1]^t} f(\mathbf{u}) d\mathbf{u}$$

par la moyenne de  $f$  sur l'ensemble  $P_n = \{\mathbf{u}_1, \dots, \mathbf{u}_n\} \subset [0, 1]^t$ :

$$\bar{\mu}_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{u}_i).$$

Idée de QMC: Choisir  $P_n = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$  plus uniformément distribué qu'un ensemble de points au hasard.

Ensembles (et suites) de points à faible discrédance, ou hautement uniformes.

MC: erreur  $E_n = \bar{\mu}_n - \mu$  aléatoire.

QMC classique:  $E_n$  est déterministe.

# Questions

Est-ce que  $E_n \rightarrow 0$  plus vite avec QMC que MC quand  $n \rightarrow \infty$ ?

Comment construire  $P_n$  et mesurer son uniformité?

Facile en 1 dimension. Mais en plusieurs dimensions?

Pour  $t$  grand, il faut trop de points pour vraiment bien remplir l'espace.

Est-ce que QMC peut quand même fonctionner? Pourquoi?

# Dimension effective

Dans plusieurs cas,  $f$  peut s'approximer assez bien par une somme de fonctions  $f_I$  dont chacune ne dépend que de peu de coordonnées de  $\mathbf{u}$ :

$$\begin{aligned} f(\mathbf{u}) &= f(u_1, \dots, u_t) \\ &= \sum_{I \subseteq \{1, \dots, t\}} f_I(\mathbf{u}) = \mu + \sum_{i=1}^t \dot{f}_{\{i\}}(u_i) + \sum_{i,j=1}^t \dot{f}_{\{i,j\}}(u_i, u_j) + \dots \end{aligned}$$

où  $f_I$  ne dépend que de  $\{u_i, i \in I\}$ .

Il suffit alors que les projections correspondantes de  $P_n$  soient bien uniformes.

# Cas simple: une dimension ( $t = 1$ )

Solutions évidentes:

$$P_n = \mathcal{Z}_n/n = \{0, 1/n, \dots, (n-1)/n\}$$

ou encore

$$P'_n = \{1/(2n), 3/(2n), \dots, (2n-1)/(2n)\}.$$

Si on permet des poids différents pour les  $f(\mathbf{u}_i)$ , on a aussi la règle du trapèze,

$$\frac{1}{n} \left[ \frac{f(0) + f(1)}{2} + \sum_{i=1}^{n-1} f(i/n) \right],$$

pour laquelle  $|E_n| = O(n^{-2})$  si  $f''$  est bornée, ou la règle de Simpson,

$$\frac{f(0) + 4f(1/n) + 2f(2/n) + \dots + 2f((n-2)/n) + 4f((n-1)/n) + f(1)}{3n}$$

qui donne  $|E_n| = O(n^{-4})$  si  $f^{(4)}$  est bornée, etc.

## Cas simple: une dimension ( $t = 1$ )

Ici, on se restreint à des poids égaux. Plus tard, on va randomizer  $P_n$  pour mieux estimer l'erreur et dans ce contexte, les poids égaux sont optimaux.

Solution simpliste en: grille rectangulaire

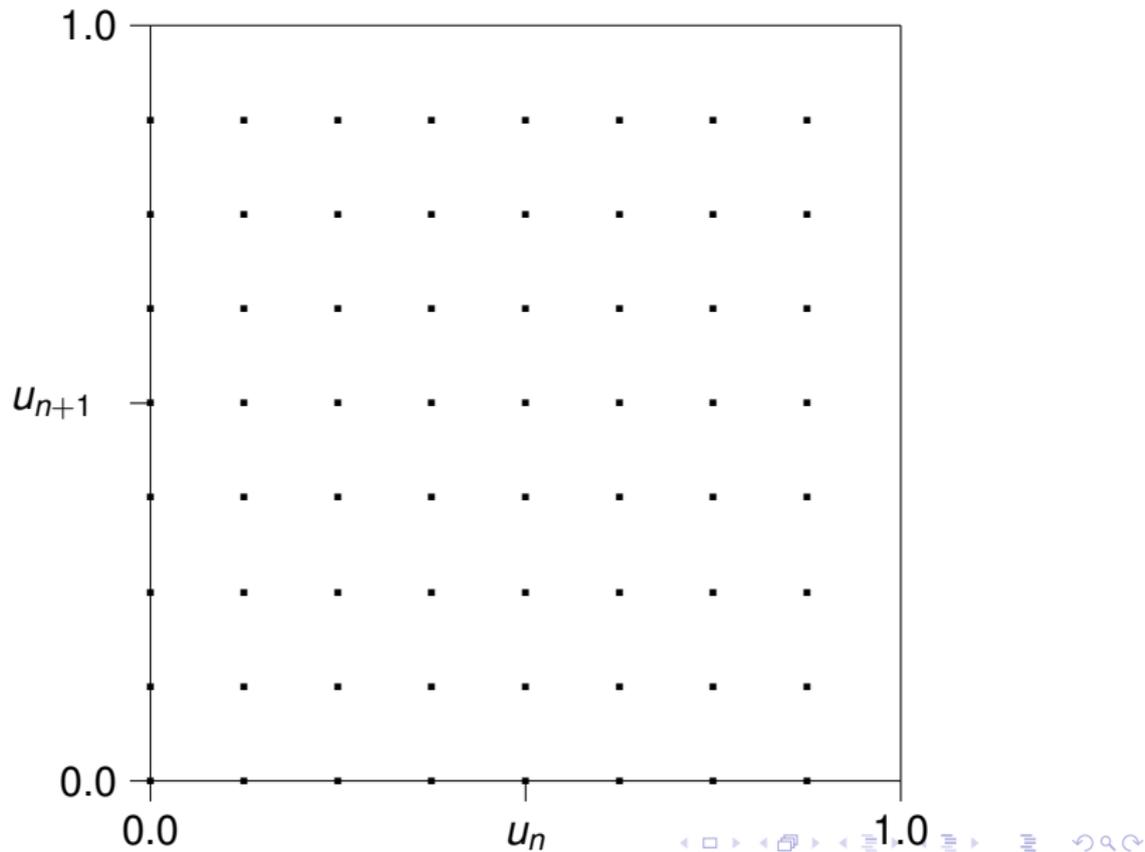
$$P_n = \{(i_1/d, \dots, i_t/d) \text{ tels que } 0 \leq i_j < d \ \forall j\},$$

où  $n = d^t$ .

Mais devient vite inutilisable quand  $t$  augmente.

Et on perd des points dans les projections: chaque projection le long d'un axe correspond à un ensemble de points.

# $t > 1$ dimensions: illustration

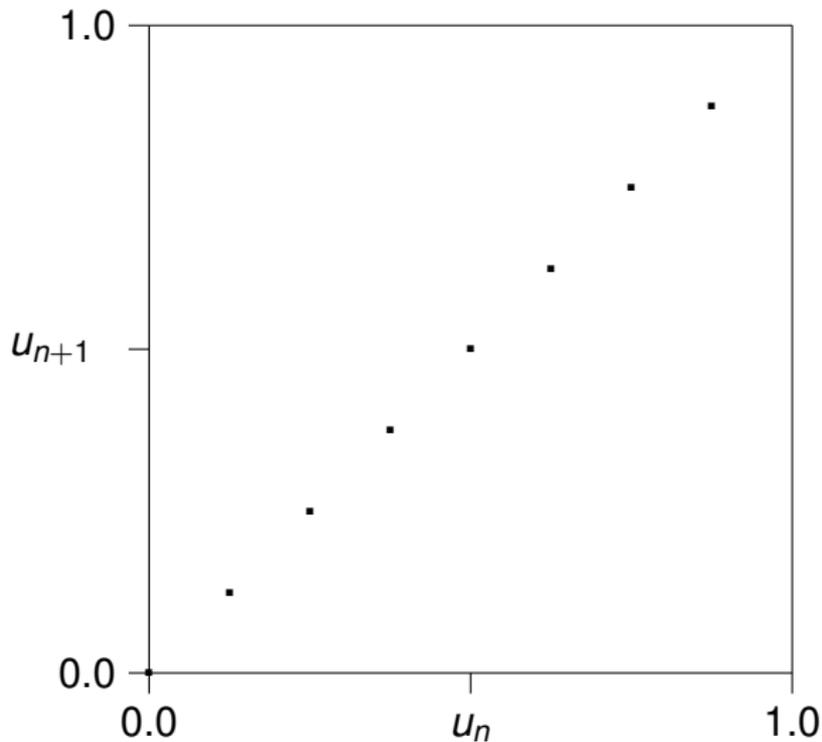


## $t > 1$ dimensions

Idée: on voudrait construire  $P_n$  de manière à ce que chaque projection unidimensionnelle soit  $\{0, 1/n, \dots, (n-1)/n\}$ .

Pour cela, on doit énumérer ces  $n$  valeurs dans un ordre différent pour chaque coordonnée. (Autrement tous les points seront sur une diagonale.)

## $t > 1$ dimensions



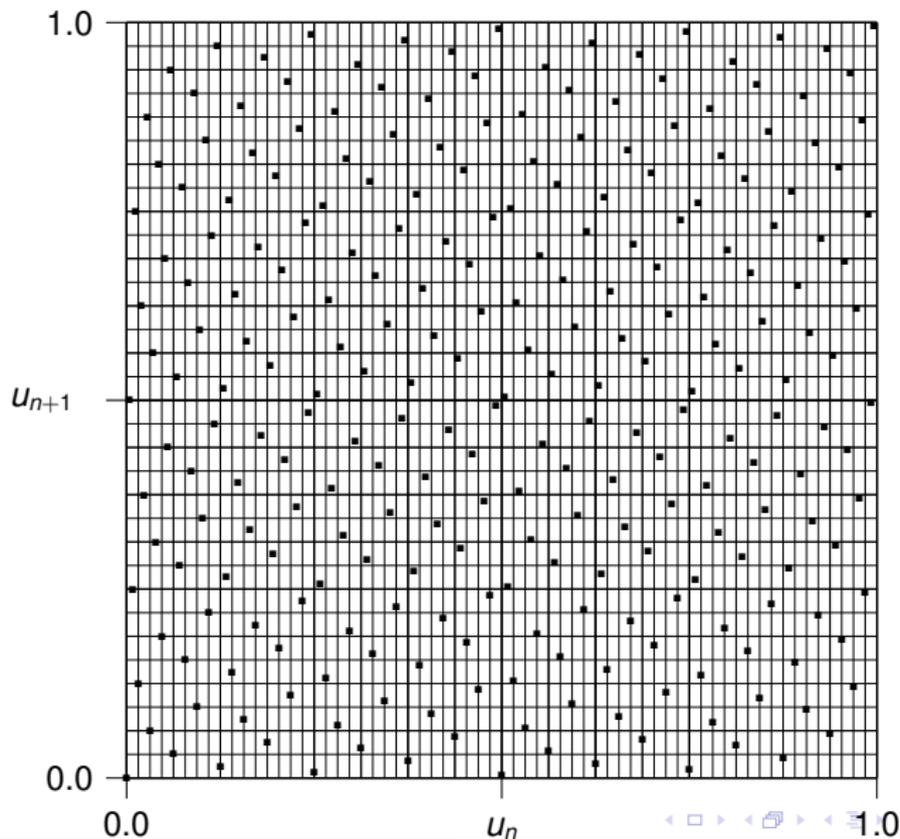
i.e., on cherche une permutation de  $\mathcal{Z}_n$  pour chacune des  $t$  coordonnées pour que  $P_n$  soit très uniforme sur  $[0, 1]^t$ .

# $t > 1$ dimensions

Exemple: Soient  $n = 2^8 = 256$  et  $t = 2$ . Prenons les points (en binaire):

$i$	$u_{1,i}$	$u_{2,i}$
0	.00000000	.0
1	.00000001	.1
2	.00000010	.01
3	.00000011	.11
4	.00000100	.001
5	.00000101	.101
6	.00000110	.011
$\vdots$	$\vdots$	$\vdots$
254	.11111110	.01111111
255	.11111111	.11111111

Example:  $n = 2^8 = 256$  and  $t = 2$



## Exemple: $n = 2^8 = 256$ and $t = 2$

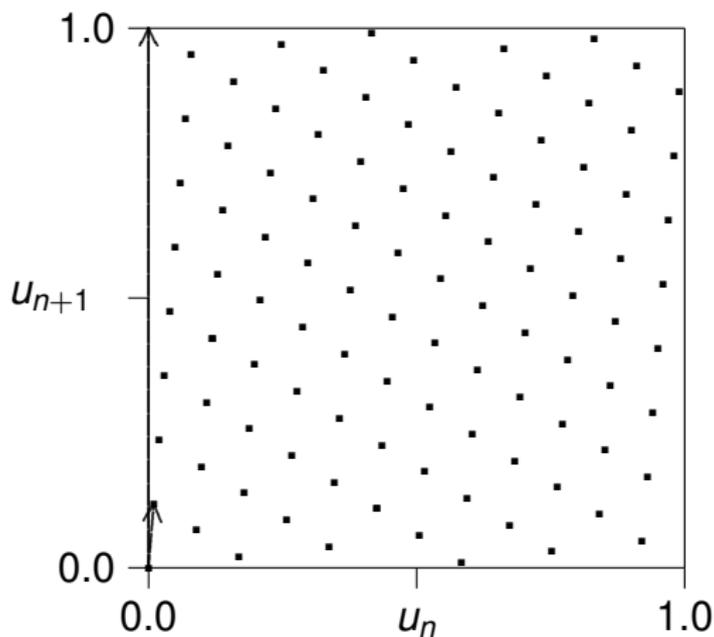
Ces points forment un  $(0, 8, 2)$ -réseau en base 2 (explication de cette appellation ultérieurement).

En général, on peut prendre  $n = 2^k$  points.

Si on partitionne  $[0, 1)^2$  en rectangles de tailles  $2^{-k_1}$  par  $2^{-k_2}$  où  $k_1 + k_2 \leq k$ , chaque rectangle contiendra exactement le même nombre de points.

Ce type de  $P_n$  est un cas particulier d'un réseau digital en base 2.

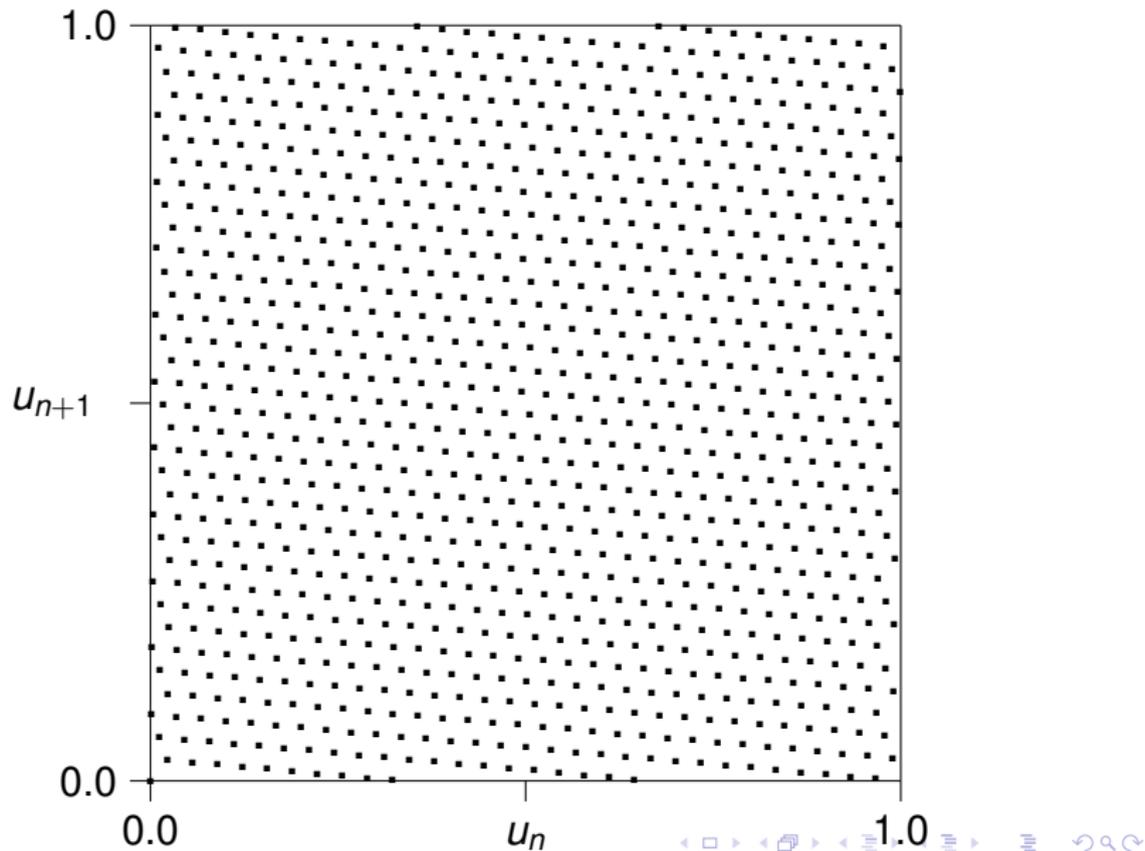
# “Lattice” (réseau) avec $n = 101$



Exemple jouet: LCG avec  $m = 101$  et  $a = 12$ .

Les  $u_i$  sont tous les multiples entiers de  $(1/101, 12/101)$ , modulo 1.

# LCG avec $t = 2$ et $n = 1021$



Les réseaux digitaux (“digital nets”) et les règles de réseaux (“lattice rules”) se généralisent à un  $t$  arbitraire.

Par exemple, on peut définir:

$$P_n = \{ \mathbf{u}_i = (i\mathbf{v} \bmod n)/n, i = 0, \dots, n-1 \}$$

pour  $\mathbf{v} = (1, v_1, \dots, v_{t-1})$  bien choisi.

Pour un réseau digital en base 2, il suffit de choisir  $t$  permutations de  $\{0, 1, \dots, 2^k - 1\}$ .

On peut aussi avoir  $t = \infty$  et/ou  $n = \infty$  (suite infinie de points).

Comment mesurer l'uniformité d'un ensemble de points?

Idée: pour  $B$  un sous-ensemble quelconque de  $[0, 1)^t$ , la fraction de  $P_n$  qui se trouve dans  $B$  devrait être à peu près égale au volume de  $B$ .

Possible pour tout  $B$ ? Non.

On peut se limiter à une famille  $\mathcal{I}$  de boîtes  $B$ , par exemple la famille  $\mathcal{I}_t^*$  de toutes les boîtes rectangulaires alignées avec les axes et ayant un coin à l'origine (les intervalles de forme  $[\mathbf{0}, \mathbf{u})$  en  $t$  dimensions).

On définit  $\Delta(B) = |(\text{fraction de } P_n \text{ qui est dans } B) - \text{volume}(B)|$   
Mesure de discrépance pour  $P_n$ :

$$D_n^*(P_n) = \sup_{B=[\mathbf{0}, \mathbf{u}) \in \mathcal{I}_t^*} \Delta(B).$$

Il y a d'autres possibilités: moyenne  $\mathcal{L}_p$  au lieu du sup, autre choix de  $\mathcal{I}$ , etc.

Erreur pire cas:

$$|\bar{\mu}_n - \mu| \leq \|f - \mu\| D_n^*(P_n)$$

où  $\|f - \mu\|$  mesure la variation de  $f$  au sens de Hardy et Krause. On connaît des suites infinies de points  $P_\infty = \{\mathbf{u}_1, \mathbf{u}_2, \dots\}$  telle que pour  $P_n = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ ,  $D_n^*(P_n) = O(n^{-1}(\ln n)^t)$ .

Asymptotiquement, c'est mieux que  $O(n^{-1/2})$ .

Mais: la borne est difficile à calculer et ne peut être pratique que lorsque  $t$  ne dépasse pas 8 ou 10.

# Estimation de l'erreur

L'idée est de randomiser  $P_n$  de manière à:

- (1) préserver la structure et l'uniformité de  $P_n$  comme ensemble;
- (2) chaque point de  $P_n$  randomisé suit la loi uniforme sur  $[0, 1)^t$ .

Exemple. Une façon de faire cela est le décalage aléatoire: générer un seul point  $\mathbf{U}$  uniformément dans  $[0, 1)^t$  et l'ajouter, modulo 1, à chacun des points de  $P_n$ .

On répète la randomisation  $m$  fois, indépendamment, et on calcule la moyenne  $\bar{X}_m$  et la variance  $S_m^2$  des  $m$  valeurs de  $\bar{\mu}_n$ .

On peut montrer que  $E[\bar{X}_m] = \mu$  et  $E[S_m^2] = \text{Var}[\bar{\mu}_n] = m\text{Var}[\bar{X}_m]$ .

Permet de calculer un intervalle de confiance pour  $\mu$ .

Choix de  $m$ ? Souvent autour de 10.