



THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Tests d'uniformité et tessellations de Voronoi

Vandooren, Julie

Award date:
2003

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Facultés Universitaires Notre-Dame de la Paix
Namur
Faculté des Sciences - Département de Mathématique

Tests d'uniformité et tessellations de Voronoi

Mémoire présenté pour l'obtention du grade
de Licencié en Sciences
mathématiques
par

Promoteur : M. Rémon

Julie Vandooren

Année académique : 2001-2002

Remerciements

Je tiens tout particulièrement à remercier mon promoteur, M. Marcel Rémon, pour sa disponibilité et ses précieux conseils.

Je remercie mes parents et ma sœur qui m'ont toujours soutenue et encouragée tout au long de ces quatre années d'étude.

Merci également à mes amis qui ont toujours su être là quand j'avais besoin d'eux.

Résumé

Dans ce mémoire, on se propose d'expérimenter une nouvelle approche pour tester l'uniformité d'un ensemble de points dans \mathbb{R}^n . Pour ce faire nous soumettons l'idée d'utiliser les tessellations de Voronoi.

Des simulations sont réalisées à l'aide d'un programme que nous avons implémenté en dimension deux, trois et quatre pour divers ensembles de données. Grâce à ces simulations, nous testons la faisabilité de cette approche.

Nous montrons également les problèmes et les limites que nous avons rencontrés.

Abstract

In this paper, we propose to experiment a new approach to test the uniformity of a set of points in \mathbb{R}^n . To do that, we submit the idea to use the tessellations of Voronoi.

Simulations are made with a program that we have implemented in 2-dimensional, 3-dimensional and 4-dimensional for various sets of data.

Thanks to these simulations, we test the feasibility of this approach.

We show also the problems and the limits that we have met.

Table des matières

Introduction	3
1 Les tessellations de Voronoi	4
1.1 Définitions d'un diagramme de Voronoi	4
1.2 Quelques propriétés des diagrammes de Voronoi	9
1.3 Les propriétés des cellules de Voronoi	12
2 La simulation de points uniformes	17
3 Quelques tests statistiques	20
3.1 Le test chi-carré (χ^2)	20
3.1.1 Test du chi-carré à une dimension	20
3.1.2 Test du chi-carré à d dimensions	21
3.2 Le test de Kolmogorov-Smirnov	22
4 Les programmes	24
4.1 Explication des programmes	24
4.2 Complexité	25
5 Test d'uniformité	27
5.1 Le problème des cellules "frontière"	27
5.2 Test d'uniformité sur $[0, 1]^2$	29
5.2.1 Analyse graphique	29
5.2.2 Tests statistiques	31
5.3 Test d'uniformité sur $[0, 1]^3$	32
5.3.1 533 cellules non-frontière sur 1000.	32
5.3.2 3413 cellules non-frontière sur 5000.	36
5.4 Test d'uniformité sur $[0, 1]^4$	39

5.4.1	201 cellules non-frontière sur 1000	39
5.4.2	898 cellules non-frontière sur 3000	41
6	Test de non-uniformité	45
6.1	En dimension 2 en enlevant un disque de points	45
6.2	En dimension 3 en enlevant une sphère de points	48
6.3	Données de type sphérique	51
6.3.1	En dimension 2 : cas d'un disque	51
6.3.2	En dimension 3 : cas de la sphère	55
6.3.3	En dimension 2 : cas de 2 disques	58
6.3.4	En dimension 3 : cas de 2 sphères	62
6.4	En dimension 2 pour des données de type couronne	66
6.5	En dimension 3 pour des données de type couronne	71
6.6	En dimension 2 pour des données de type 2 bandes horizontales	74
6.7	En dimension 3 pour des données de type 2 bandes horizontales	78
	Conclusions	81
	Annexes	83
	Bibliographie	91

Introduction

Le but de ce mémoire était de mettre au point un test d'uniformité en grande dimension (par exemple \mathbb{R}^{10}) en nous basant sur les propriétés des cellules de Voronoi. Vu la difficulté à mettre au point une théorie exacte en grande dimension, nous utiliserons des simulations pour nous faire une idée de la théorie. L'objectif du mémoire est de tester la faisabilité de cette approche.

Dans un premier temps, nous allons donc simuler des cellules de Voronoi à l'aide d'un programme informatique que nous avons implémenté. Les points générateurs des cellules seront les points dont nous désirons tester l'uniformité. Nous essayerons ensuite d'estimer les distributions de l'une ou l'autre caractéristique (par exemple le volume) de ces cellules. Nous tenterons enfin de mettre au point un test d'uniformité.

Dans le premier chapitre, nous étudierons les tessellations de Voronoi et leurs propriétés. Le chapitre suivant sera consacré à la simulation de points. Dans la troisième partie, quelques tests classiques d'analyse de données et d'uniformité seront présentés.

Nous analyserons ensuite les résultats des simulations. A partir de cette analyse, nous essayerons de dégager les propriétés nécessaires pour avoir l'uniformité d'un ensemble de données. Nous mettrons en évidence les difficultés d'une telle approche et les limites auxquelles nous avons été confrontés.

Chapitre 1

Les tessellations de Voronoi

Les tessellations de Voronoi sont expliquées avec beaucoup de détails dans le livre de Okabe et al. [1].

Une tessellation est une division de l'espace en régions polygonales convexes. Une tessellation particulière est la tessellation ou diagramme de Voronoi. Nous allons la définir et en donner les principales propriétés.

1.1 Définitions d'un diagramme de Voronoi

Le diagramme de Voronoi correspond à un pavage de l'espace en cellules tel que tout point d'une cellule soit le plus proche possible au sens d'une certaine métrique d'une unité définissant l'espace initial (point ou segment par exemple).

Supposons que nous ayons un ensemble de points (au minimum deux, mais un nombre fini) P dans le plan euclidien. Nous supposons qu'ils sont tous distincts. Nous assignons chaque point du plan au membre le plus proche de l'ensemble des points. Si un point est à égal distance de deux membres de P , nous l'assignons à ces deux membres. L'ensemble des coordonnées assignées à chaque point de P forment les *cellules de Voronoi*. Tout l'espace est recouvert car chaque coordonnée est assignée à au moins un membre de P .

L'ensemble de coordonnées assignées à deux membres ou plus de P

forment les *frontières* des régions. L'ensemble des régions forment une *tessellation* de l'espace. Cette tessellation est appelée un *diagramme de Voronoi dans le plan* et les régions de cette tessellation sont appelées les *polygones de Voronoi dans le plan*.

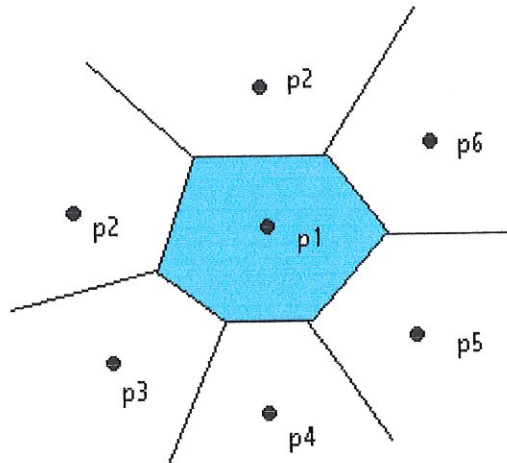


FIG. 1.1 – Diagramme de Voronoi dans le plan.

Définissons maintenant un diagramme de Voronoi formellement.

Définition 1.1.1 (*Diagramme de Voronoi dans \mathbb{R}^2*) *Etant donné un ensemble fini de points dans l'espace euclidien, on associe chaque entité dans l'espace au membre le plus proche, au sens de la distance euclidienne, de l'ensemble de points. Il en résulte une tessellation du plan en un ensemble de régions associées aux membres de l'ensemble de points. On appelle cette tessellation le diagramme de Voronoi plan généré par l'ensemble de points.*

Nous pouvons également écrire cette définition en termes mathématiques. Soient p_1, \dots, p_n , les n points de coordonnées cartésiennes $(x_{11}, x_{12}), \dots, (x_{n1}, x_{n2})$ que nous noterons $\mathbf{x}_1, \dots, \mathbf{x}_n$ vectoriellement. Soit p un point arbitraire de l'espace de coordonnées $\mathbf{x} = (x_1, x_2)$.

La distance euclidienne entre p et p_i est donnée par :

$$d(p, p_i) = \|x - x_i\| = \sqrt{(x_1 - x_{i1})^2 + (x_2 - x_{i2})^2}$$

Définition 1.1.2 Soit $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$ où $2 < n < \infty$ un ensemble de points dans le plan et $x_i \neq x_j \forall i \neq j, i, j \in I_n = \{1, \dots, n\}$. On appelle la région donnée par

$$V(p_i) = \{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\| \quad \forall j \neq i, j \in I_n \} \quad (1.1)$$

le diagramme de Voronoi plan et l'ensemble donné par

$$\mathcal{V} = \{V(p_1), \dots, V(p_n)\}$$

le diagramme de Voronoi généré par P .

Nous appelons p_i de $V(p_i)$ le point générateur du i -ème polygone de Voronoi, et l'ensemble $P = \{p_1, \dots, p_n\}$ l'ensemble générateur du diagramme de Voronoi \mathcal{V} .

Remarquons que dans le définition 1.1.2, la relation dans l'équation 1.1 est définie en terme de \leq . Il suit qu'un polygone Voronoi est un ensemble fermé. La frontière d'un polygone est notée $\partial V(p_i)$. La frontière consiste soit en un segment, soit en une droite, soit en une demi-droite que nous appellerons *arêtes de Voronoi*. Nous les noterons e_i .

Un point final d'une arête de Voronoi est appelé *un vertex de Voronoi*. Nous le noterons q_i

Lorsqu'il existe au moins un vertex de Voronoi auquel quatre arêtes de Voronoi au plus se rencontrent dans le diagramme de Voronoi \mathcal{V} , on dit que \mathcal{V} est *dégénéré*; autrement \mathcal{V} est non dégénéré.

Hypothèse de non-dégénérescence :

Chaque vertex de Voronoi dans un diagramme de Voronoi a exactement trois arêtes de Voronoi.

Les diagrammes de Voronoi peuvent également être définis en terme de demi-plans. On considère la ligne perpendiculaire qui coupe le segment $\overline{p_i p_j}$ en son milieu. Cette ligne est appelée la *ligne bissectrice entre p_i et p_j* et est notée $b(p_i, p_j)$. Comme un point sur la ligne bissectrice est équidistant des points générateurs p_i et p_j ,

$$b(p_i, p_j) = \{ x \mid \|x - x_i\| = \|x - x_j\| \} \quad j \neq i$$

La ligne bissectrice coupe le plan en deux demi-plans et donne

$$H(p_i, p_j) = \{ x \mid \|x - x_i\| \leq \|x - x_j\| \} \quad j \neq i$$

$H(p_i, p_j)$ est appelé la *région de dominance de p_i sur p_j*

Définition 1.1.3 *Le diagramme de Voronoi défini par des demi-plans. Soit $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$ avec $2 < n < \infty$ et $x_i \neq x_j \forall i \neq j, i, j \in I_n = \{1, \dots, n\}$. On appelle la région*

$$V(p_i) = \bigcap_{j \in I_n \setminus \{i\}} H(p_i, p_j)$$

le polygone de Voronoi associé à p_i et l'ensemble $\mathcal{V}(P) = \{V(p_1), \dots, V(p_n)\}$ le diagramme de Voronoi plan généré par P .

Nous pouvons maintenant généraliser cette définition à l'espace euclidien à m dimensions.

Définition 1.1.4 *Diagramme de Voronoi dans \mathbb{R}^m . Soit $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^m$ avec $2 < n < \infty$ et $x_i \neq x_j \forall i \neq j, i, j \in I_n$. Nous appelons la région*

$$\begin{aligned} V(p_i) &= \{ \mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\| \quad \forall j \neq i, j \in I_n \} \\ &= \bigcap_{j \in I_n \setminus \{i\}} H(p_i, p_j) \end{aligned}$$

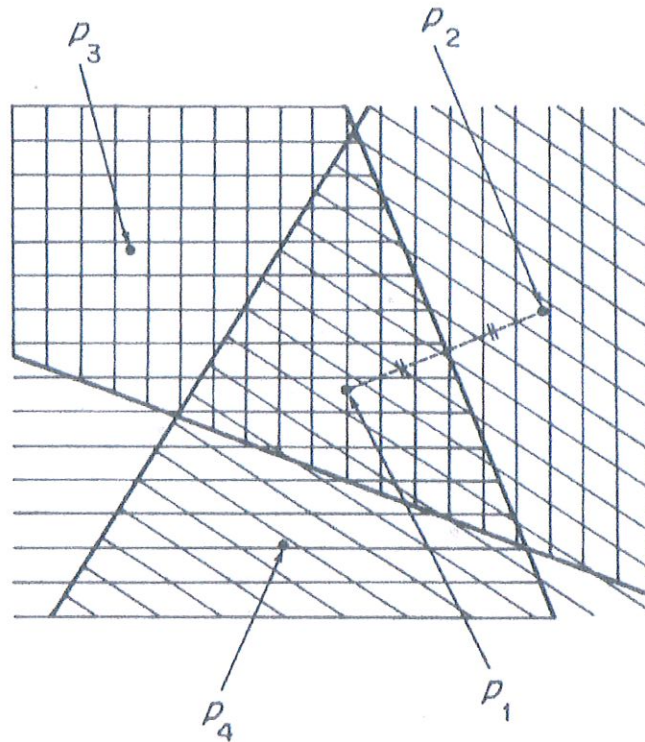


FIG. 1.2 – Cellule de Voronoi obtenue par des demi-plans.

le polyèdre à m dimensions de Voronoi associé à p_i et l'ensemble $\mathcal{V}(P) = \{V(p_1), \dots, V(p_n)\}$
 le diagramme de Voronoi dans \mathbb{R}^m généré par P où

$$H(p_i, p_j) = \{x \mid \|x - p_i\| \leq \|x - p_j\|\} \quad j \neq i$$

pour p_i et $p_j \in \mathbb{R}^m$.

Les frontières d'un polyèdre sont appelées *les faces*.

1.2 Quelques propriétés des diagrammes de Voronoi

Nous allons maintenant citer quelques propriétés des diagrammes de Voronoi. Pour les preuves de ces résultats, voir [1].

Proposition 1.2.1 Soit $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ ($2 \leq n < \infty$), un ensemble de points distincts.

L'ensemble $V(p_i)$ défini par :

$$V(p_i) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\| \quad \forall j \neq i, j \in I_n\}$$

est un polygone convexe non-vide, et $\mathcal{V}(P) = \{V(p_1), \dots, V(p_n)\}$ satisfait à

$$\bigcup_{i=1}^n V(p_i) = \mathbb{R}^2,$$

$$[V(p_i) \setminus \partial V(p_i)] \cap [V(p_j) \setminus \partial V(p_j)] = \emptyset, \quad i \neq j, j \in I_n$$

Le diagramme de Voronoi $\mathcal{V}(P)$ est donc une tessellation unique de \mathbb{R}^2 pour P .

Cette propriété est également valable dans \mathbb{R}^m , il suffit de remplacer le polygone par un polyèdre et \mathbb{R}^2 par \mathbb{R}^m .

Proposition 1.2.2 Pour un diagramme de Voronoi généré par un ensemble de points distincts $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ ($2 \leq n < \infty$), un polygone de Voronoi $V(p_i)$ est non borné si et seulement si p_i est sur la frontière de l'enveloppe convexe de P , c'est à dire si $p_i \in \partial CH(P)$

Cette propriété est également valable en dimension m .

Proposition 1.2.3 *Pour un diagramme de Voronoi g n r  par un ensemble de points distincts $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ ($2 \leq n < \infty$) :*

- *les ar tes de Voronoi sont des droites infinies si et seulement si P est collin aire*
- *une ar te de Voronoi $e(p_i, p_j)$ est une demi-droite si et seulement si P est non-collin aire et si p_i et p_j sont des points g n rateurs cons cutifs de la fronti re de l'enveloppe convexe de P .*
- *Supposons que p_i et p_j d finissent une ar te de Voronoi $e(p_i, p_j)$. Alors cette ar te est un segment si et seulement si le segment $\overline{p_i p_j}$ n'est pas une ar te de l'enveloppe convexe de P .*

Proposition 1.2.4 *Le point g n rateur le plus proche de p_i g n re une ar te de Voronoi de $V(p_i)$.*

Proposition 1.2.5 *Le g n rateur p_i est le point g n rateur le plus proche du point p si et seulement si $V(p_i)$ contient p .*

Proposition 1.2.6 *Pour chaque vertex de Voronoi, $q_i \in Q$, dans un diagramme de Voronoi, il y a un seul cercle C_i centr  en q_i qui passe par trois ou plus points g n rateurs et qui ne contient aucun point dans son int rieur. Sous l'hypoth se de non-d g n rescence, C_i passe par exactement trois points g n rateurs.*

Un cercle qui ne contient aucun point int rieur est appel  cercle vide. Le cercle de la proposition 1.2.6 est donc un cercle vide.

Proposition 1.2.7 *Le cercle C_i de la proposition 1.2.6 est le cercle vide le plus large parmi tous les cercles vides centr s en le vertex de Voronoi q_i*

Proposition 1.2.8 *Soit n , n_e , n_v , le nombre de points g n rateurs, d'ar tes de Voronoi, et de vertex de Voronoi d'un diagramme de Voronoi dans \mathbb{R}^2 respectivement ($2 \leq n < \infty$). Alors*

$$n_v - n_e + n = 1$$

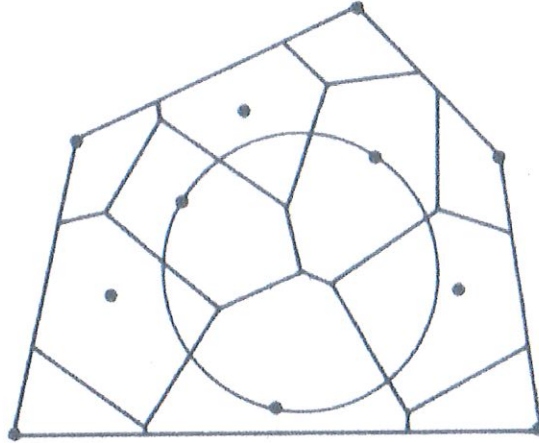


FIG. 1.3 – Le cercle vide le plus large.

Proposition 1.2.9 Soit n_k le nombre de faces à k dimensions d'un diagramme de Voronoi à m dimensions. Alors

$$\sum_{k=0}^m (-1)^k n_k = (-1)^m$$

Proposition 1.2.10 Soit n , n_e , n_v , le nombre de points générateurs, d'arêtes de Voronoi, et de vertex de Voronoi d'un diagramme de Voronoi dans \mathbb{R}^2 respectivement ($3 \leq n < \infty$). Alors

$$n_e \leq 3n - 6$$

$$n_v \leq 2n - 5$$

Proposition 1.2.11 Le nombre moyen d'arêtes par polygone de Voronoi n'excède pas six.

1.3 Les propriétés des cellules de Voronoi

Ce paragraphe est fortement inspiré de Okabe et al [1].

Dans une partie de ce travail, nous allons simuler des cellules de Voronoi pour étudier leurs caractéristiques et plus particulièrement pour estimer la distribution de probabilité de ces caractéristiques.

Plusieurs études ont déjà été réalisées à ce sujet, nous citons ci-après les principaux résultats qui nous intéressent.

Ces études considèrent une tessellation de Voronoi particulière où les points de P sont localisés dans \mathbb{R}^m suivant un processus de Poisson homogène de densité $\lambda > 0$:

$$Pr(N(A) = x) = \frac{\lambda|A|e^{-\lambda|A|x}}{x!}, \quad x = 0, 1, 2, \dots$$

Le paramètre λ est la densité de points dans une unité de volume.

Pour éviter les cas particuliers, rendant difficiles la construction des cellules de Voronoi, trois hypothèses sont rajoutées :

- P est localement fini, c'est à dire qu'il y a seulement un nombre fini de points dans une région bornée
- L'hypothèse de non-colinéarité : il n'existe pas $k + 1$ points qui se trouvent sur un hyperplan de $k - 1$ dimensions dans \mathbb{R}^m , $k = 2, \dots, m$
- L'hypothèse de non-cosphérité : il n'existe pas $m + 2$ points qui se trouvent sur la frontière d'une sphère dans \mathbb{R}^m .

Etant données les hypothèses ci-dessus, l'intersection de $k + 1$ cellules

$$F(p_0, \dots, p_k) = \bigcap_{i=0}^k V(p_i),$$

est soit vide soit de dimension $m - k$.

Définition 1.3.1 Une intersection $F(p_0, \dots, p_k)$ non-vide est appelée une $(m - k)$ face de la tessellation et une cellule une m -face.

Par exemple, dans \mathbb{R}^3 , les 0- face, 1- faces et 2- faces sont les vertex, les arêtes et les faces.

Des mathématiciens ont calculé les moments d'ordre 1 et d'ordre 2 de l'aire dans \mathbb{R}^2 et du volume dans \mathbb{R}^3 des cellules de Voronoi :

		Valeur exacte	Valeur numérique
Espérance de l'aire d'une cellule et son second moment	$E(A)$ $E(A^2)$	$1/\lambda$	$1.280\lambda^{-2}$
Espérance du volume d'une cellule et son second moment	$E(V)$ $E(V^2)$	$1/\lambda$	$1.180\lambda^{-2}$

Nous allons maintenant nous intéresser aux fonctions de distributions de ces caractéristiques :

Les auteurs de [1] font référence à une cinquantaine de chercheurs qui ont travaillé sur la distribution de l'aire et du volume des cellules de Voronoi.

Parmi ces auteurs, on trouve Hinde et Miles qui estiment l'aire des cellules par une fonction gamma généralisée à trois paramètres.

$$f_1(x) = \frac{rb^{q/r} x^{q-1} \exp(-bx^r)}{\Gamma\left(\frac{q}{r}\right)} \quad r > 0, b > 0, q > 0 \quad (1.2)$$

Les valeurs obtenues à partir de l'équation 1.2 avec $r=1.0787$, $b=3.0328\lambda^r$ et $q=3.3095$ sont dessinées à la figure suivante. Visuellement cela donne un bon ajustement, bien que ce soit rejeté par un test χ^2 .

Tanemura (voir [7]) propose l'équation 1.2 pour l'aire, le volume et l'hypervolume d'une cellule de Voronoi dans \mathbb{R}^2 , \mathbb{R}^3 et \mathbb{R}^4 . Ses valeurs estimées des paramètres sont reprises dans le tableau ci-après.



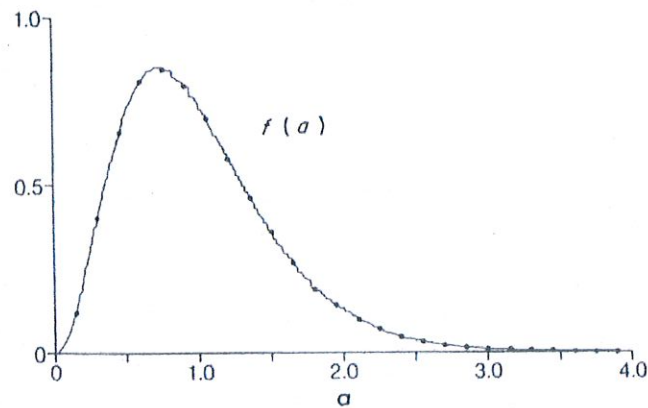


FIG. 1.4 – Histogramme normalisé de l'aire standardisée λA d'une cellule de Voronoi dans \mathbb{R}^2 montrant les valeurs pointées de la densité gamma avec les paramètres estimés de Hinde et Miles [1].

dimension	r	b	q
2	1.07805	$3.04011\lambda^r$	3.31498
3	1.16391	$4.06342\lambda^r$	4.80651
4	1.29553	$4.90493\lambda^r$	6.49707

Plusieurs scientifiques estiment que bien que les distributions lognormale et gamma soient très proches, la description de l'aire ou du volume d'une cellule de Voronoi est en général plus adéquate avec une distribution gamma qu'avec une distribution lognormale.

Sur base de ses simulations, Kiang suggère que la taille d'une cellule (aire si $m=2$ et volume d'une cellule si $m=3$) suit une distribution gamma donnée à l'équation 1.3

$$f_2(x) = \frac{b^q x^{q-1} \exp(-bx)}{\Gamma(q)} \quad b > 0, q > 0 \quad (1.3)$$

avec $q=2m$ et $b=2m\lambda$.

Beaucoup d'autres chercheurs ont également donné un excellent ajustement de l'aire A en 2 dimensions, et du volume V en 3 dimensions suivant

l'équation 1.3. Nous donnons quelques valeurs qu'ils ont trouvées dans le tableau ci-après.

Dimension	q	b	Chercheur(s)
2	3.61	3.37λ	Di Cenzo et Wertheim
	3.61	3.61λ	Marthinsen
	3.571	3.571λ	Vaz et Fortes
	3.37	3.37λ	Zaninetti
	3.718	3.718λ	Kumar et Kurtz
3	5.56	5.56λ	Andrade et Fortes
			Vaz et Fortes
			Thorvaldsen
			Marthinsen
	5.38	5.38λ	Zaninetti
	5.7869λ	Kumar et al.	

La plupart de ces estimations sont rejetées par le test de Kolmogorov-Smirnov et le test du chi-carré au niveau de signification 0.01.

On peut noter qu'excepté DiCenzo et Wertheim toutes ces estimations satisfont la condition $E(A) = E(V) = q/b = \lambda^{-1}$.

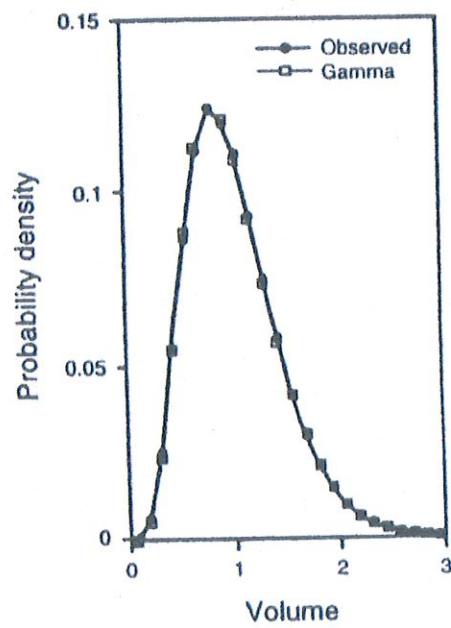


FIG. 1.5 – Histogramme normalisé du volume d'une cellule de Poisson de Voronoi dans \mathbb{R}^3 , comparé à la densité d'une gamma ayant les paramètres estimés de Kumar et Kurtz. [1]

Chapitre 2

La simulation de points uniformes

Pour plus de détails voir le livre de Brian Ripley [2]

Notre but est de tester l'uniformité d'un ensemble de points dans l'hypercube $[0, 1]^k$. Pour nos simulations, nous avons donc besoin de générer une suite de nombres uniformes. On désire générer u_1, u_2, \dots tels que les u_i aient l'air de suivre une loi uniforme sur $[0, 1]$.

Par les méthodes informatiques, les nombres générés ne sont jamais aléatoires mais pseudo-aléatoire. On génère une suite de nombres aléatoires par une formule déterministe de manière à obtenir une suite qui semble aléatoire. Idéalement, on voudrait que ce générateur soit efficace (rapide et besoin en mémoire modeste) et qu'il possède une longue période. L'objectif est que les points $(u_i, u_{i+1}, \dots, u_{i+k-1})$ soient répartis à peu près uniformément dans l'hypercube à k dimensions $[0, 1]^k$.

Plusieurs types de générateurs existent. Les générateurs congruentiels sont les mieux connus et les plus utilisés. Ils sont de la forme

$$\begin{cases} x_0 \\ x_{n+1} \end{cases} = (ax_n + c) \bmod m$$

où a, c et m sont des entiers. La suite générée est une suite $(x_n)_{n>0}$ de nombres entiers compris entre 0 et $m-1$.

Pour obtenir un nombre réel entre 0 et 1, il faut encore diviser l'entier aléatoire ainsi généré par m .

Chapitre 2

La simulation de points uniformes

Pour plus de détails voir le livre de Brian Ripley [2]

Notre but est de tester l'uniformité d'un ensemble de points dans l'hypercube $[0, 1]^k$. Pour nos simulations, nous avons donc besoin de générer une suite de nombres uniformes. On désire générer u_1, u_2, \dots tels que les u_i aient l'air de suivre une loi uniforme sur $[0, 1]$.

Par les méthodes informatiques, les nombres générés ne sont jamais aléatoires mais pseudo-aléatoires. On génère une suite de nombres aléatoires par une formule déterministe de manière à obtenir une suite qui semble aléatoire. Idéalement, on voudrait que ce générateur soit efficace (rapide et besoin en mémoire modeste) et qu'il possède une longue période. L'objectif est que les points $(u_i, u_{i+1}, \dots, u_{i+k-1})$ soient répartis à peu près uniformément dans l'hypercube à k dimensions $[0, 1]^k$.

Plusieurs types de générateurs existent. Les générateurs congruentiels sont les mieux connus et les plus utilisés. Ils sont de la forme

$$\begin{cases} x_0 \\ x_{n+1} \end{cases} = (ax_n + c) \bmod m$$

où a, c et m sont des entiers. La suite générée est une suite $(x_n)_{n>0}$ de nombres entiers compris entre 0 et $m-1$.

Pour obtenir un nombre réel entre 0 et 1, il faut encore diviser l'entier aléatoire ainsi généré par m .

Définition 2.0.2 *Un générateur linéaire congruentiel est dit multiplicatif lorsque $c=0$.*

Le nombre maximum de valeurs différentes possibles pour la suite est m . Dès qu'un nombre est répété, toute la séquence recommence. On notera par p la *période* ou le nombre de valeurs possibles de la suite.

Définition 2.0.3 *Si $p = m$, le générateur est dit de période maximale.*

Théorème 2.0.1 *Un générateur linéaire congruentiel mixte est de période maximale $p = m$ si et seulement si les trois conditions suivantes sont satisfaites :*

1. c et m n'ont pas de diviseurs communs
2. $a = 1 \pmod{4}$ quand m est un multiple de 4
3. $a = 1 \pmod{k}$ pour tous les nombres premiers k diviseurs de m

rem : si m est un nombre premier, le générateur est de période maximale seulement si $a=1$.

Pour effectuer nos simulations, nous avons choisi le générateur de Park et Miller (voir [2]) :

$$\begin{cases} x_0 \text{ impair} \\ x_{n+1} = 16807 x_n \pmod{2^{31}-1} \end{cases}$$

C'est un générateur de période maximale couramment utilisé.

Nous avons implémenté le générateur selon l'algorithme de Schrage. Cette méthode permet d'éviter les dépassements de capacité (overflows).

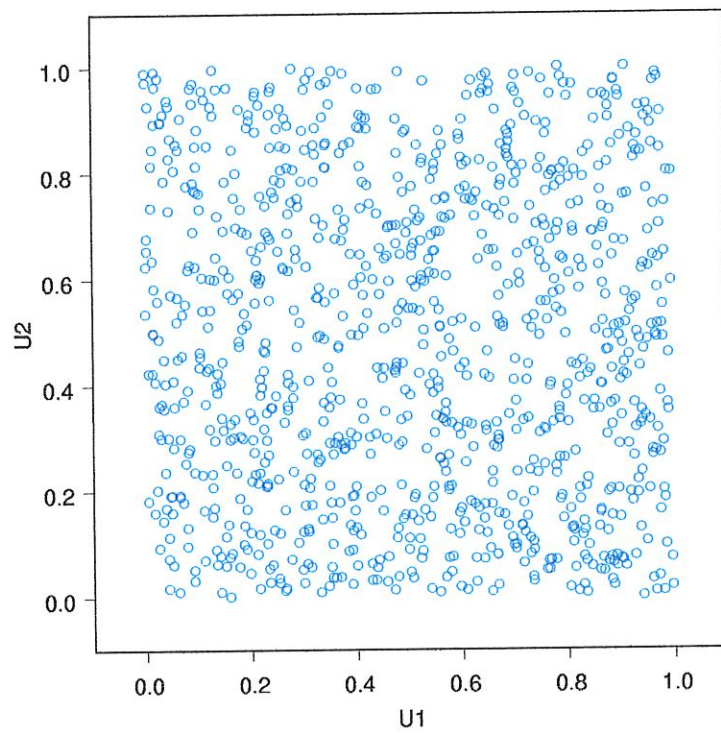


FIG. 2.1 – Points uniformes fournis par le générateur.

Chapitre 3

Quelques tests statistiques

3.1 Le test chi-carré (χ^2)

3.1.1 Test du chi-carré à une dimension

Soit x_1, \dots, x_n n valeurs aléatoires de fonction de distribution commune $F(x)$. Nous allons tester l'hypothèse :

$$H_0 : \forall x \quad F(x) = F_0(x)$$

contre l'hypothèse :

$$H_A : \exists x \quad F(x) \neq F_0(x)$$

Soit e_i les fréquences théoriques attendues et soit O_i les fréquences observées.

On calcule la statistique

$$Q = \sum_{i=1}^k \frac{(O_i - e_i)^2}{e_i}$$

Q suit une loi de χ^2 à $k-1$ degrés de liberté si n est grand ($n \geq 30$).

Règle de décision : Pour un niveau de signification α (par exemple $\alpha=0.05$), on recherche dans les tables de la loi χ^2 la valeur du quantile $\chi_{k-1,1-\alpha}^2$. Si $Q > \chi_{k-1,1-\alpha}^2$, on rejette l'hypothèse H_0 .

Ce test permet donc de tester si une suite de nombres est uniformément distribuée.

On commence par diviser l'intervalle $[0,1]$ en k intervalles de longueur $1/k$. $\pi_i = 1/k$ est la probabilité que n soit dans la catégorie i . Soit $e_i = n\pi_i$ le nombre d'occurrences attendues dans la catégorie i et soit O_i le nombre d'occurrences observées dans la catégorie i (c'est à dire que O_i est le nombre de $u_i \in [\frac{i-1}{k}, \frac{i}{k}]$). On calcule la statistique

$$Q = \sum_{i=1}^k \frac{(O_i - e_i)^2}{e_i}$$

Q suit une loi de χ^2 à $k-1$ degrés de liberté si n est grand ($n \geq 30$).

Règle de décision : Pour un niveau de signification α (par exemple $\alpha=0.05$), on recherche dans les tables de la loi χ^2 la valeur du quantile $\chi_{k-1,1-\alpha}^2$. Si $Q > \chi_{k-1,1-\alpha}^2$, on rejette l'hypothèse $H_0 : (u_1, \dots, u_n)$ est un échantillon selon la loi uniforme.

3.1.2 Test du chi-carré à d dimensions

Ce test permet de tester si des d -uples sont répartis uniformément dans $[0, 1]^d$. Les u_i sont regroupés en d -uples :

$$(u_1, \dots, u_d), (u_{d+1}, \dots, u_{2d}), \dots$$

On divise $[0,1]$ en k intervalles égaux, ce qui divise l'hypercube $[0, 1]^d$ en k^d sous-hypercubes égaux. On compte le nombre de d -uples observés dans chaque sous-hypercube (j_1, j_2, \dots, j_d) , soit f_{j_1, j_2, \dots, j_d} .

On calcule la statistique

$$Z = \frac{k^d}{n} \sum_{j_1=1,2,\dots,k} \dots \sum_{j_d=1,2,\dots,k} (f_{j_1, j_2, \dots, j_d} - \frac{n}{k^d})^2$$

Z suit approximativement une loi de $\chi_{k^d-1}^2$ sous H_0 .

Règle de décision : On rejette H_0 si $\chi_{k^d-1}^2$ est trop grand.
Si $k^d - 1$ est très grand, on peut approximer par une normale.

La puissance de ce test est faible : il faut un très grand nombre d'observations pour assurer un comptage significatif dans chaque cellule.

3.2 Le test de Kolmogorov-Smirnov

Soit x_i, \dots, x_n n valeurs aléatoires de fonction de distribution commune $F(x)$.

Nous allons tester l'hypothèse :

$$H_0 : \forall x F(x) = F_0(x)$$

contre l'hypothèse :

$$H_A : \exists x F(x) \neq F_0(x)$$

L'idée du test est de mesurer la distance entre la fonction de répartition empirique $F_n(x)$ de l'ensemble des n points générés et la fonction de distribution théorique.

$$\begin{aligned} F_n(x) &= \frac{1}{n}(\#obs \leq x) && \text{fonction empirique} \\ &&& \text{(où n est le nombre total d'observations)} \\ F_0(x) &= P(X \leq x) && \text{fonction théorique} \end{aligned}$$

Pour voir si la différence entre les deux fonctions de distribution est significative, on utilise la statistique de Kolmogorov :

$$D_n = \max_x |F_0(x) - F_n(x)|$$

Règle de décision : On rejette H_0 au niveau de signification α si D_n est grand. En d'autre terme, on rejette H_0 si $D_n > d_{n,\alpha}$ où $d_{n,\alpha}$ est le quantile d'ordre α du test statistique de Kolmogorov-Smirnov.

Ce test peut également être utilisé pour tester l'uniformité d'un ensemble de points. On mesure la distance entre la fonction de répartition empirique $F_n(x)$ de l'ensemble des n points générés et la fonction de distribution théorique de la loi uniforme sur $[0,1]$.

Chapitre 4

Les programmes

4.1 Explication des programmes

Les différents programmes calculent une approximation des cellules de Voronoi et diverses caractéristiques de ces cellules : le volume, la distance entre le point générateur de la cellule et le point appartenant à la cellule le plus éloigné, la distance entre le point générateur de la cellule et le point appartenant à la cellule le plus proche, la distance moyenne entre le point générateur et tous les points appartenant à la cellule.

Dans la suite du travail, nous nous intéresserons uniquement à l'analyse de l'aire et du volume des cellules. Les autres caractéristiques ont été calculées au cas où ce travail serait poursuivi dans le futur.

Le programme détecte également les cellules frontières (nous expliquerons pourquoi par la suite).

Pour simuler les points uniformes, points qui sont les générateurs des cellules de Voronoi, le programme fait appel à la fonction random. L'appel successif de cette fonction donne des nombres aléatoires dont la distribution tend vers une distribution uniforme sur $]0, 1[$.

Le programme approxime alors les cellules de Voronoi générées par ces points. Pour ce faire, nous générons des points que nous appelons points-grille, dans $[0, 1]^d$ à égale distance les uns des autres. Ces points sont au nombre de $(k + 1)^d$, où k est le nombre de points-grille sur un segment $[0, 1]$.

En dimension 3 par exemple, un peu plus d'un million de points sert à estimer les cellules. On peut donc dire que notre approximation est assez précise.

4.2 Complexité

Les simulations ont été réalisées sur un pentium 850 Mhz.

Dimensions	Temps d'exécution
2	1 sec
3	90 sec
4	3 heures
5	11 heures

Remarque : dans le cas de la dimension cinq, la simulation a été réalisée avec $k = 50$

Des simulations ont également été tentées en dimension 10 mais n'ont pas abouti. Après plus de 24 heures, le programme n'avait toujours pas fini de tourner.

Nous pointons donc ici une des difficultés de notre approche. Notre implémentation n'est pas assez performante et ne peut pas être utilisée en dimension supérieure à 5. Une approche est de choisir le nombre de points-grille de façon à ce que ce nombre soit une fonction linéaire de la dimension, voire même un nombre total de points-grille constant. Nous n'avons pas eu le temps d'implémenter cette approche.

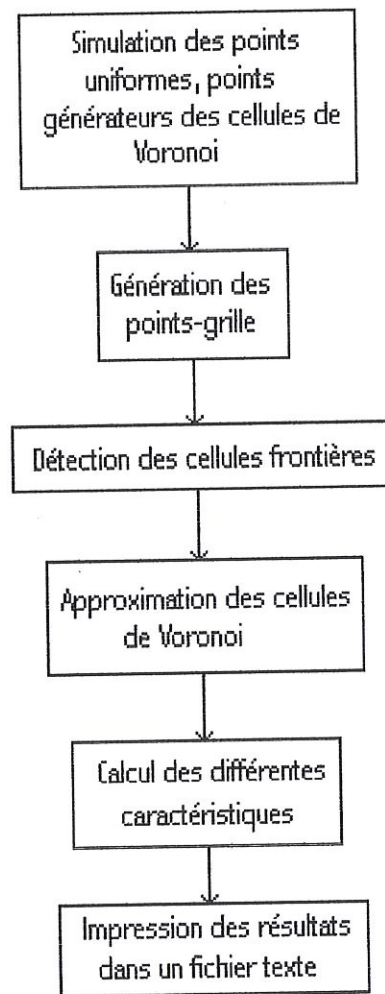


FIG. 4.1 – Organigramme des programmes.

Chapitre 5

Test d'uniformité

Nous avons réalisé des simulations pour différents types de données en dimensions deux, trois, quatre et cinq. Nous allons maintenant essayer de voir les conclusions que nous pouvons tirer de ces simulations.

Avant de débiter l'analyse, nous expliquons la deuxième difficulté que nous avons rencontrée.

5.1 Le problème des cellules "frontière"

Un autre problème rencontré lors de nos simulations est celui des cellules frontières. De telles cellules introduisent un effet de bord comme on peut le voir à la figure 5.1. Ces cellules ne sont pas "complètes". Et de ce fait biaisent le calcul du volume.

Pour éviter cet effet de bord, nous avons limité notre analyse des résultats aux cellules non-frontières. Et c'est là qu'apparait la difficulté mathématique.

Dans le tableau ci-après, on peut voir le nombre de cellules "non-frontières" pour les dimensions deux, trois, quatre et cinq. On remarque que la proportion de cellules-frontières augmente avec le nombre de dimensions. Si bien que déjà en dimension 5, il ne reste plus que 30 cellules non-frontières.

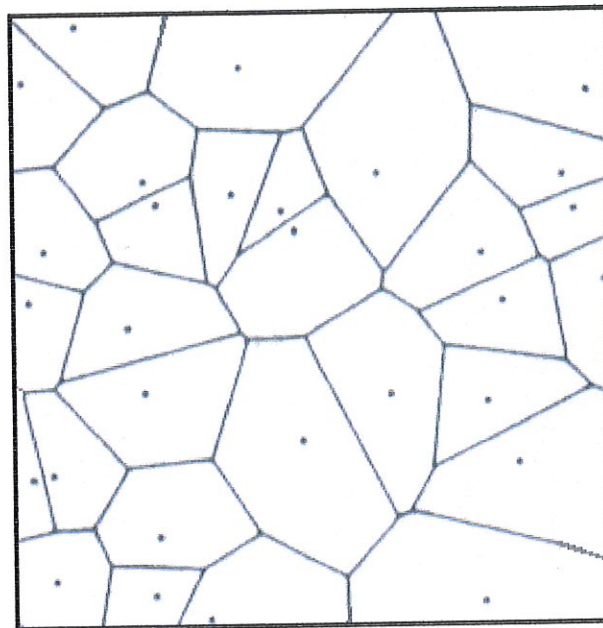


FIG. 5.1 – Le problème des cellules frontières.

Dimension	Nbre de cellules non "frontière" sur 1000 cellules
2	893
3	533
4	201
5	30

Une analyse du volume pour trente cellules sur mille n'a pas beaucoup de sens. Nous n'avons donc pu tirer profit des simulations pour des dimensions supérieures à quatre.

Passons maintenant à l'analyse des résultats.

Pour plus de facilités, nous n'analysons pas le volume mais le volume exposant(1/dimension), que nous centrons et divisons par l'écart type. Nous appelons cette variable *zvol*.

$$zvol = \frac{(\sqrt[d]{volume} - moyenne(\sqrt[d]{volume}))}{ecart\ type(\sqrt[d]{volume})}$$

Comme nous l'avons vu précédemment, plusieurs chercheurs ont trouvé que

le volume des cellules de Voronoi suivait une distribution gamma avec les paramètres proches de 2 dimensions, la variable $zvol$ que nous analysons suit approximativement une distribution normale lorsque les données sont uniformes.

Nous étudions tout d'abord cette variable graphiquement. Et dans certains cas, nous lui appliquons des tests statistiques.

5.2 Test d'uniformité sur $[0, 1]^2$

Nous avons retenu 893 cellules non-frontière sur 1000.

	Volume	Volume ^(1/2)
Moyenne	0.000961085	0.02983876
Ecart type	0.0005266044	0.008415037

5.2.1 Analyse graphique

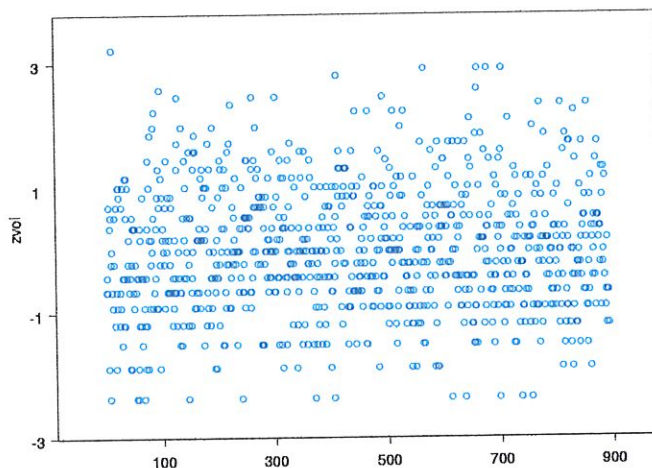


FIG. 5.2 – Diagramme de $zvol$ des cellules non-frontière en dimension 2.

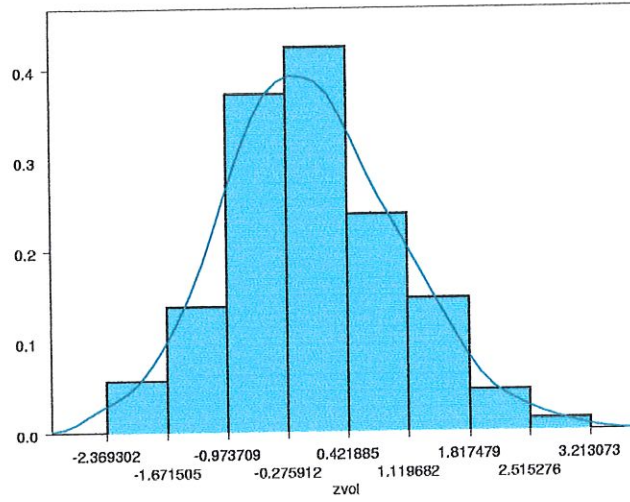


FIG. 5.3 – histogramme et densité de $zvol$ des cellules non-frontière en dimension 2.

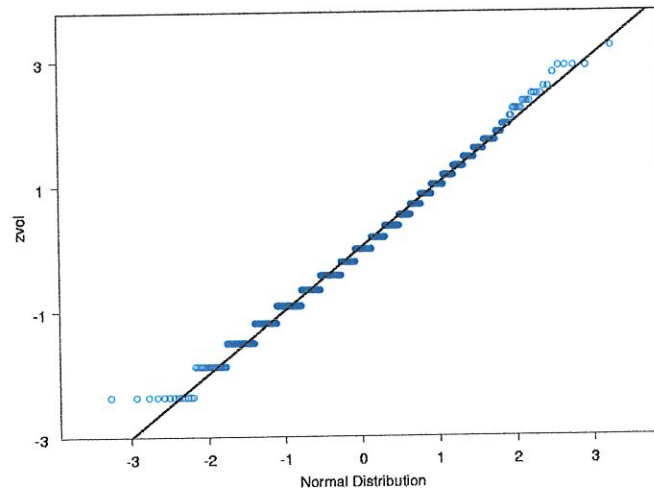


FIG. 5.4 – QQplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 2.

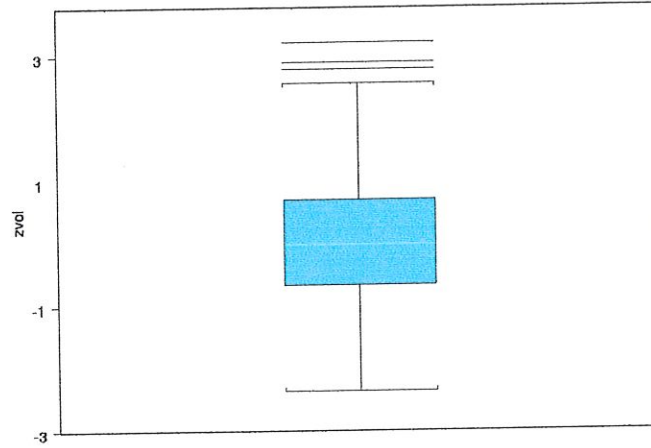


FIG. 5.5 – Boxplot de zvol des cellules non-frontière en dimension 2.

Les histogrammes et le diagramme QQplot indiquent que la variable zvol suit approximativement une loi normale centrée et réduite. L'analyse de la figure 5.5 montre que seulement 3 cellules dépassent le 95-quantiles.

5.2.2 Tests statistiques

Pour vérifier ce que semblent nous montrer les graphiques, appliquons à nos données les tests de Kolmogorov Smirnov et du chi-carré pour l'hypothèse H_0 : zvol suit une distribution $N(0,1)$.

Kolmogorov-S	χ^2
p-value=0.0018	p-value = 0.000
D = 0.0628	$\chi^2 = 856.2262$

La statistique de Kolmogorov-Smirnov est petite et tend à confirmer l'hypothèse H_0 même si le test du chi-carré rejette l'hypothèse de normalité. On sait que les tests d'ajustement de densité acceptent très difficilement l'hypothèse H_0 lorsque les données sont nombreuses.

5.3 Test d'uniformité sur $[0, 1]^3$

5.3.1 533 cellules non-frontière sur 1000.

Nous avons tout d'abord fait une analyse en retenant 533 cellules non-frontière sur 1000.

	Volume	Volume ^(1/3)
Moyenne	0.000954621	0.09649645
Ecart type	0.0004022636	0.01394921

Analyse graphique

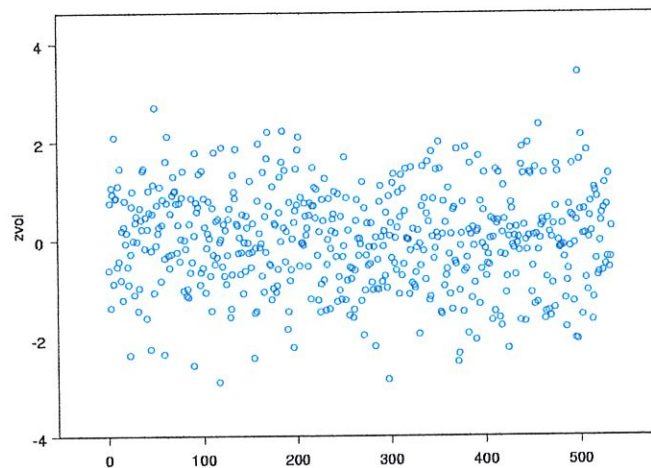


FIG. 5.6 – Diagramme de zvol des cellules non-frontière en dimension 3.

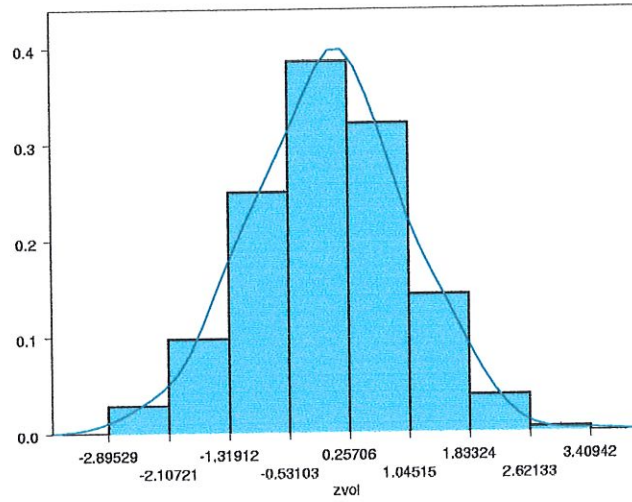


FIG. 5.7 – Histogramme et densité de $zvol$ des cellules non-frontière en dimension 3.

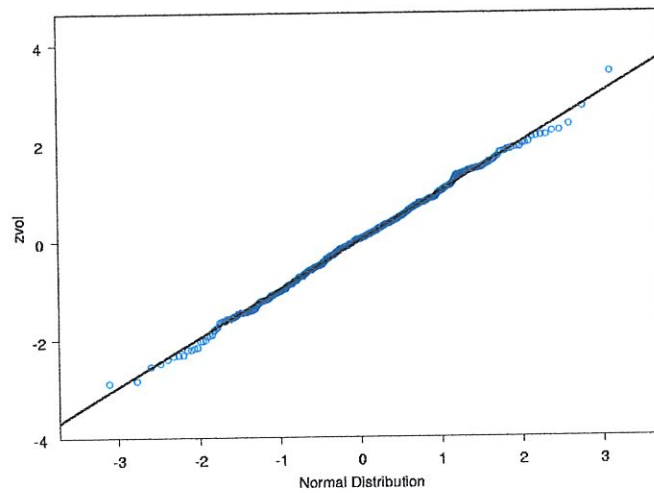


FIG. 5.8 – QQplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 3.

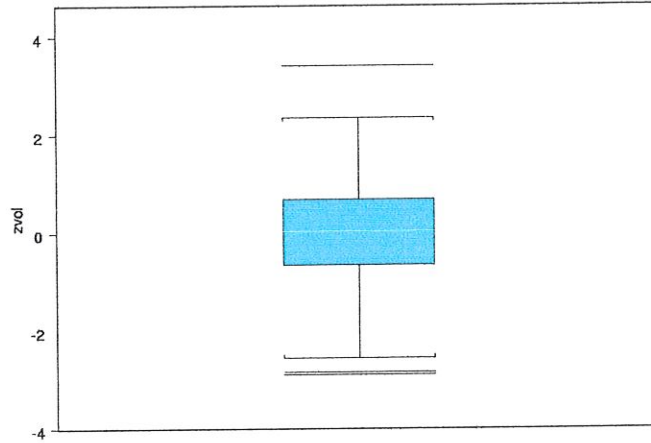


FIG. 5.9 – Boxplot de $zvol$ des cellules non-frontière en dimension 3.

Les graphes nous montrent que $zvol$ suit approximativement une loi $N(0,1)$. Nous pouvons voir 4 valeurs extrêmes sur le diagramme 5.9.

Tests statistiques

Nous vérifions les conclusions suggérées par les graphiques en appliquant les tests de Kolmogorov-Smirnov et χ^2 . Nous testons l'hypothèse H_0 : $zvol$ suit une distribution $N(0,1)$.

Kolmogorov-S	χ^2
p-value=0.7765	p-value = 0.9575
D = 0.0286	$\chi^2 = 13.4822$

Les résultats de ces tests nous poussent à valider l'hypothèse H_0 .

Nous avons également comparé la variable $zvol$ à un échantillon de nombres aléatoires de distribution $N(0,1)$ simulés par le générateur de S-plus.

A la figure 5.10, on peut voir les distributions cumulées de ces 2 échantillons.

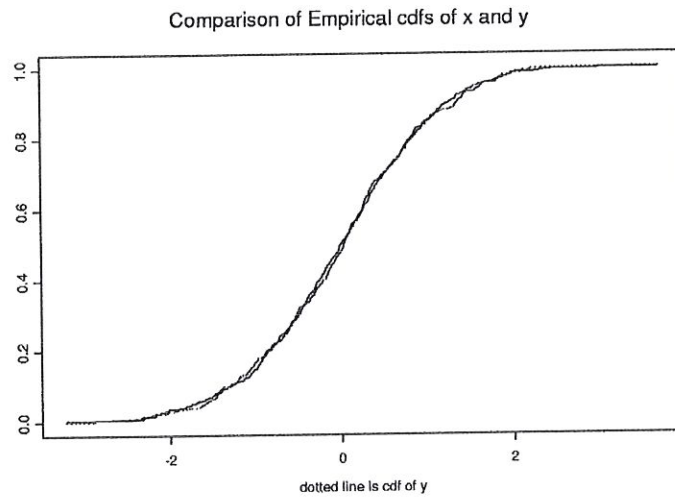


FIG. 5.10 – Distributions cumulées de zvol et d'un échantillon de $N(0,1)$ simulée par S-plus.

Kolmogorov-Smirnov appliqué aux deux échantillons donne pour l'hypothèse d'égalité de distribution :

$$p\text{-value}=0.7188$$

$$D=0.0413$$

Au vu de tous ces résultats, on ne rejette pas l'hypothèse selon laquelle la variable zvol suit une distribution $N(0,1)$.
On peut donc conclure à l'uniformité des données.

5.3.2 3413 cellules non frontière sur 5000.

Nous avons également réalisé une simulation avec 5000 points générateurs pour les cellules au lieu de 1000.

Voici les résultats :

	Volume	Volume ^(1/3)
Moyenne	0.0002053846	0.05771390
Ecart type	0.00009076776	0.0071795

Analyse graphique

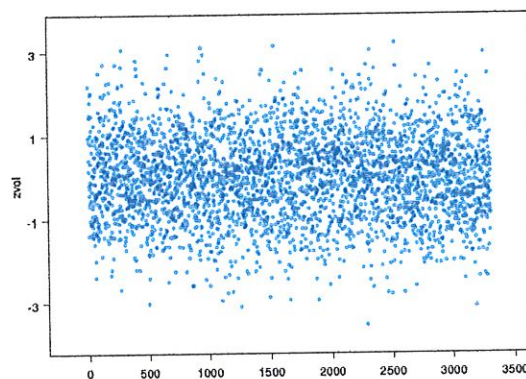


FIG. 5.11 – Scatterplot de zvol des cellules non-frontière en dimension 3.

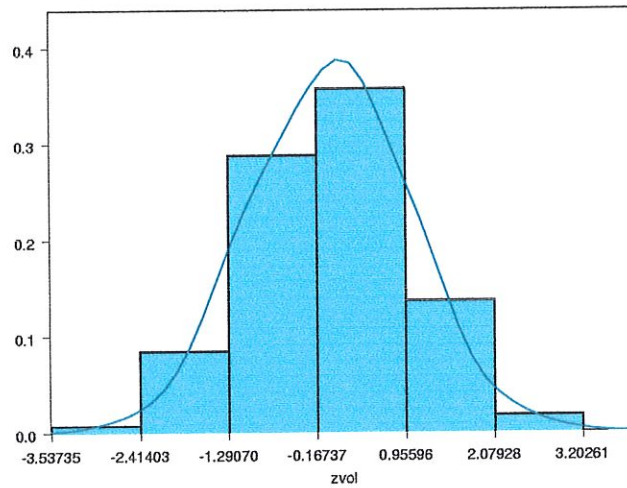


FIG. 5.12 – Histogramme et densité de $zvol$ des cellules non-frontière en dimension 3.

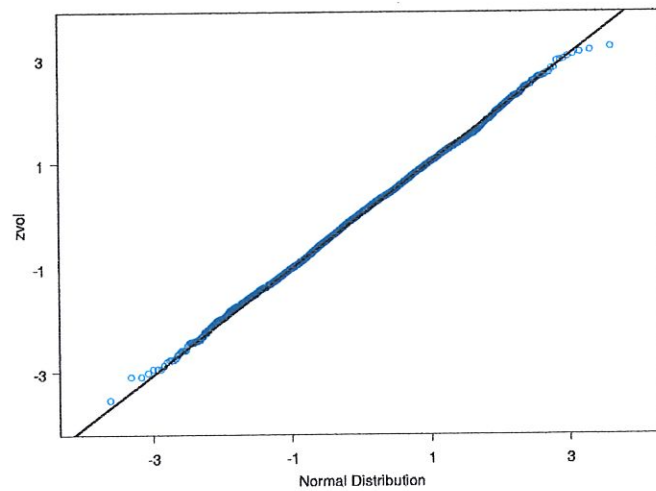


FIG. 5.13 – QQplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 3.

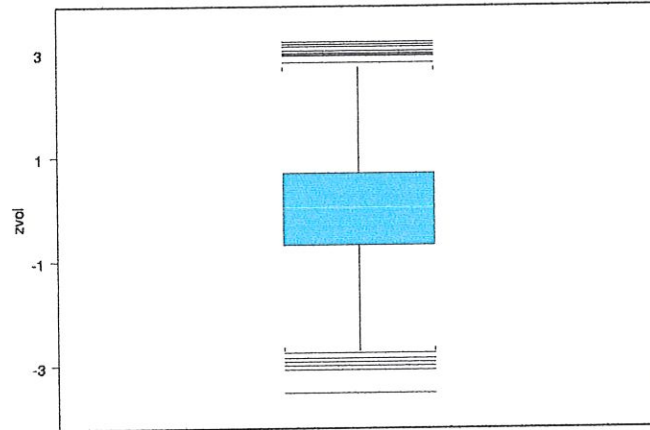


FIG. 5.14 – Boxplot de zvol des cellules non-frontière en dimension 3.

A nouveau les graphes semblent nous indiquer la quasi normalité de la variable zvol. Nous remarquons une quinzaine de valeurs hors des 95-quantiles sur le diagramme boxplot 5.14.

Tests statistiques

Voyons ce que nous apprennent les tests de Kolmogorov-Smirnov et du chi-carré. L'hypothèse H_0 est : zvol suit une distribution $N(0,1)$.

Kolmogorov-S	χ^2
p-value=0.5195	p-value = 0.0814
D = 0.0142	$\chi^2 = 65.6502$

Bien que les p-value soient moins bonnes (particulièrement celle du test chi-carré), la statistique de Kolmogorov-Smirnov est assez petite. Les graphes suggèrent une distribution normale.

On a remarqué la présence de valeurs outliers (une quinzaine) sur les graphiques 5.14 (boxplot) et 5.13 (QQplot). Ces valeurs pourraient expliquer le fait que les tests soient moins bons.

Une autre explication peut être que le test du chi-carré rejette plus facilement l'hypothèse nulle quand le nombre d'observations devient grand. Malgré tout, on peut conclure que $zvol$ suit une distribution presque normale.

5.4 Test d'uniformité sur $[0, 1]^4$

5.4.1 201 cellules non-frontière sur 1000

Nous testons maintenant l'uniformité sur $[0, 1]^4$. Nous avons réalisé 2 simulations. Commençons par analyser la première. Nous avons simulé 1000 cellules de Voronoi et nous gardons les 201 cellules non-frontière pour l'analyse.

	Volume	Volume ^(1/4)
Moyenne	0.0009482106	0.1729556
Ecart type	0.0003620716	0.01747982

Analyse graphique

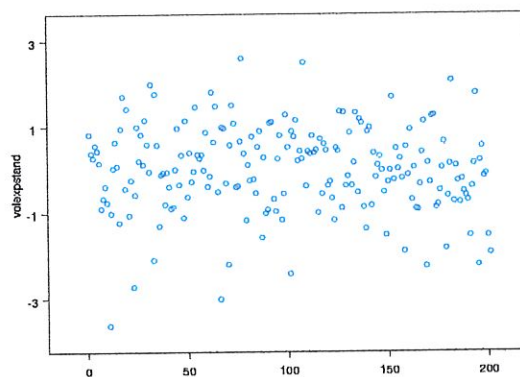


FIG. 5.15 – Scatterplot de $zvol$ des cellules non-frontière en dimension 4.

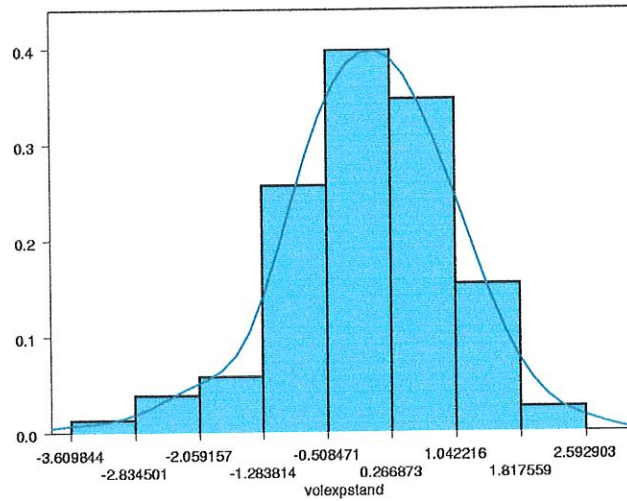


FIG. 5.16 – Histogramme et densité de zvol des cellules non-frontière en dimension 4.

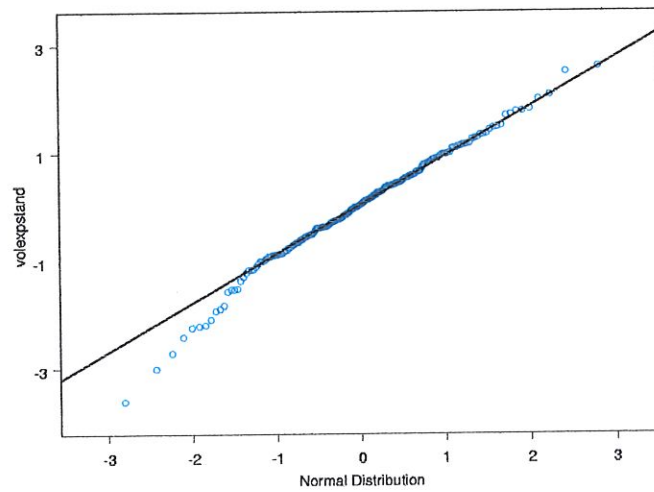


FIG. 5.17 – QQplot avec ligne de densité de zvol des cellules non-frontière en dimension 4.

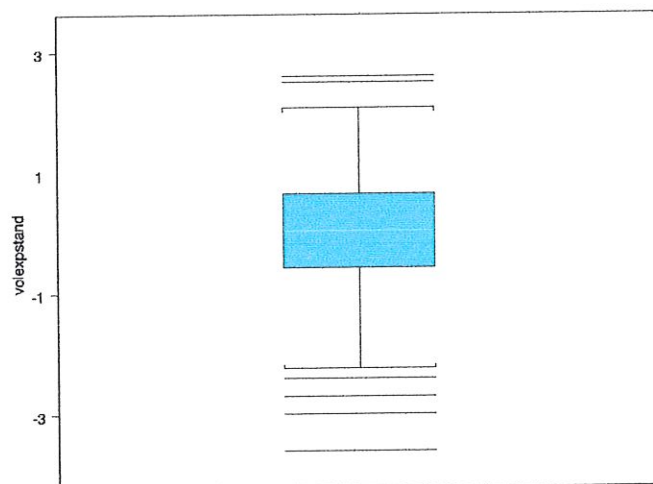


FIG. 5.18 – Boxplot de zvol des cellules non-frontière en dimension 4.

Les graphes montrent que la variable zvol suit approximativement une loi normale(0,1). Encore une fois, nous remarquons quelques valeurs extrêmes sur le diagramme 5.22.

Tests statistiques

Les résultats des tests de Kolmogorov-Smirnov et du chi-carré pour l'hypothèse H_0 : zvol suit une distribution $N(0,1)$ sont :

Kolmogorov-S	χ^2
p-value=0.7535	p-value = 0.7522
D = 0.0476	$\chi^2 = 11.8806$

Les tests tendent à confirmer l'hypothèse H_0 .

5.4.2 898 cellules non frontière sur 3000

Pour la deuxième simulation, nous avons généré 3000 cellules. Nous avons conservé les 898 cellules non-frontière pour notre analyse.

	Volume	Volume ^(1/4)
Moyenne	0.0003201356	0.1321257
Ecart type	0.0001133916	0.01219224

Analyse graphique

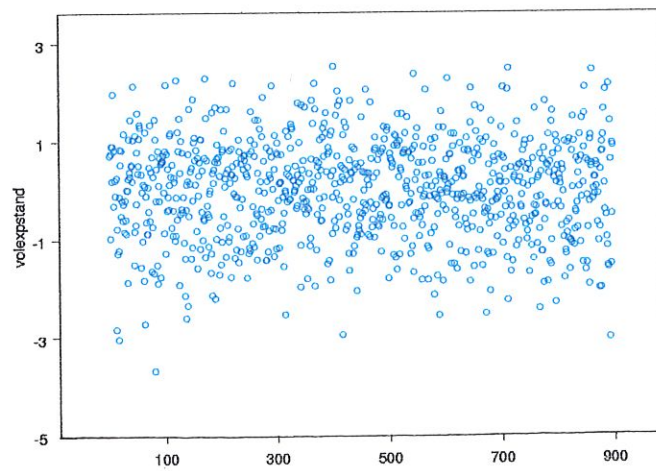


FIG. 5.19 – Scatterplot de zvol des cellules non-frontière en dimension 4.

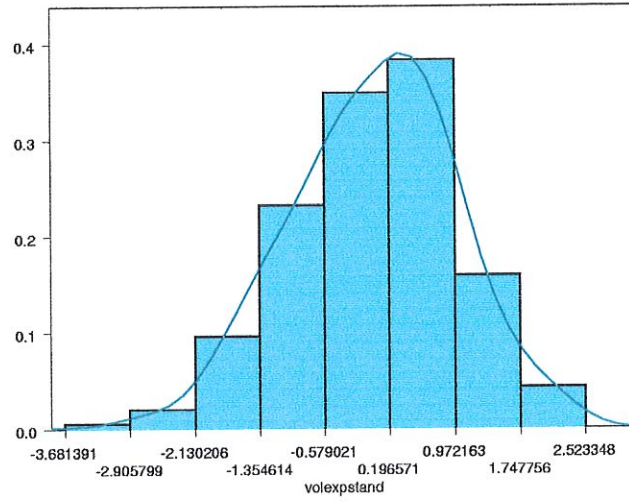


FIG. 5.20 – Histogramme et densité de zvol des cellules non-frontière en dimension 4.

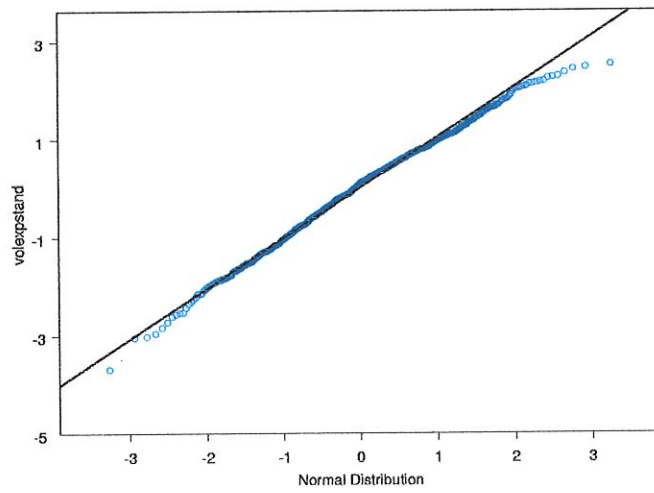


FIG. 5.21 – QQplot avec ligne de densité de zvol des cellules non-frontière en dimension 4.

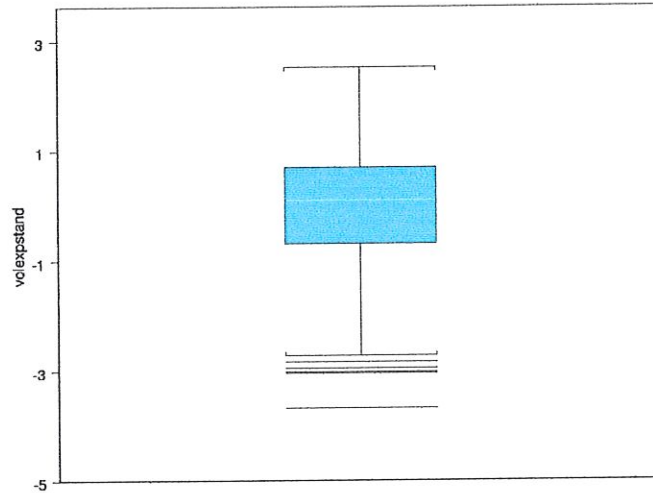


FIG. 5.22 – Boxplot de zvol des cellules non-frontière en dimension 4.

Graphiquement, nous remarquons que zvol suit une distribution approximativement normale centrée et réduite. Cinq valeurs sont hors des 95-quantiles.

Tests statistiques

Nous testons l'hypothèse H_0 : zvol suit une distribution $N(0,1)$. Les p-value et les statistiques des tests de Kolmogorov-Smirnov et du chi-carré sont données ci-après.

Kolmogorov-S	χ^2
p-value=0.0933	p-value = 0.2125
D = 0.0413	$\chi^2 = 35.8664$

Même si les p-value sont faibles, la statistique de Kolmogorov-Smirnov est petite et tend à confirmer ce que nous montrent les graphes.

Chapitre 6

Test de non-uniformité

6.1 En dimension 2 en enlevant un disque de points

Nous allons maintenant regarder ce qui se passe si on enlève une partie des points-générateurs comme à la figure 6.1. Il reste 888 points-générateurs. Sur les 888 cellules générées par ces points, 781 ne sont pas des cellules frontière.

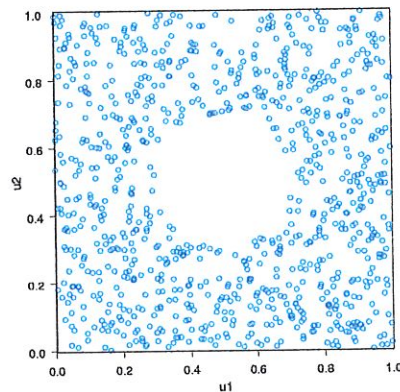


FIG. 6.1 – Points générateurs des cellules

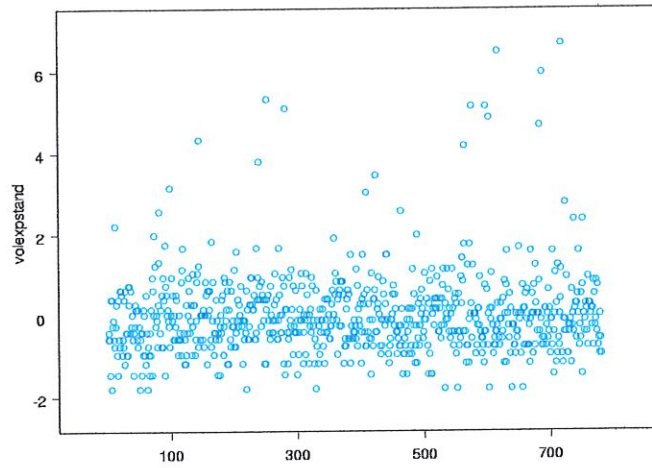


FIG. 6.2 – Scatterplot de zvol des cellules non-frontière en dimension 2 pour les données de la figure 6.1.

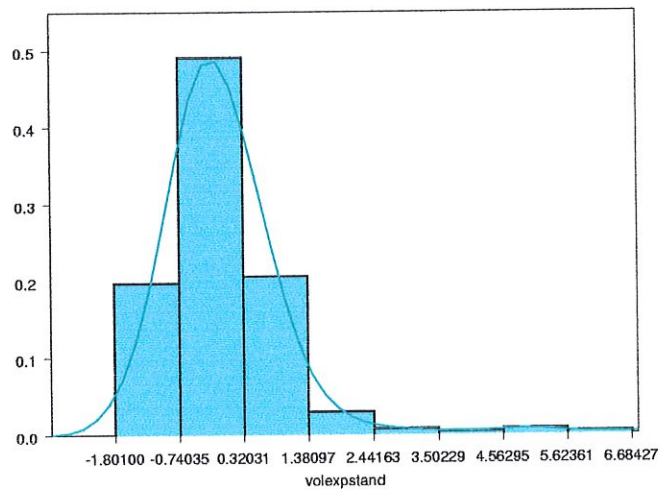


FIG. 6.3 – Histogramme et densité de zvol des cellules non-frontière en dimension 2 pour les données de la figure 6.1.

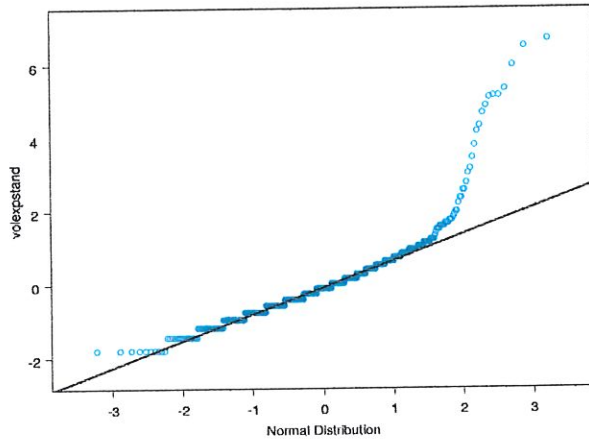


FIG. 6.4 – QQplot avec ligne de densité de zvol des cellules non-frontière en dimension 2 pour les données de la figure 6.1.

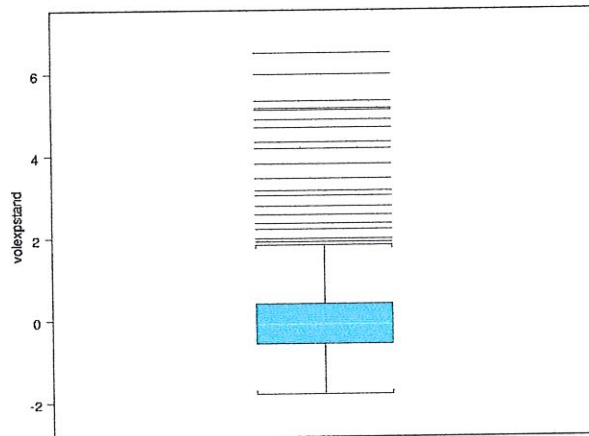


FIG. 6.5 – Boxplot de zvol des cellules non-frontière en dimension 2 pour les données de la figure 6.1.

Sur tous les graphes, on peut voir qu'un certains nombre de cellules ont un volume beaucoup plus grand. Ce qui est tout à fait normal vu la configuration des données.

Le graphique 6.5 nous montre qu'une vingtaine de cellules ont un volume plus grand. Cela pourrait nous donner une information sur l'importance du trou dans les données.

On peut également voir sur les graphiques que les autres cellules semblent suivre une distribution normale.

Nous allons maintenant appliquer les tests de Kolmogorov-Smirnov et du chi-carré à nos données. Nous les appliquons tout d'abord à l'ensemble des cellules non-frontière. Nous retirons ensuite progressivement les valeurs extrêmes et nous essayons de voir si en retirant ces cellules nous retrouvons l'uniformité des données.

	Toutes les cellules	- les 10 + grandes	- les 15 + grandes	- les 20 + grandes	- les 25 + grandes
KS	0.1218	0.0805	0.0675	0.0649	0.0644
p-value	0	0.0001	0.0019	0.0033	0.0038
χ^2	590.3175	614.3424	657.5744	660.8384	667.6852
p-value	0	0	0	0	0

Si nous retirons les 20 cellules de plus grands volumes, nous retrouvons des résultats semblables à ceux fournis par les tests du paragraphe 5.2 où nous avons l'uniformité des données. Nous pouvons donc dire que le nombre de valeurs extrêmes nous permet de tester l'uniformité des données.

6.2 En dimension 3 en enlevant une sphère de points

Nous considérons maintenant $[0, 1]^3$ où nous avons enlevé une sphère de points générateurs au centre de l'hypercube.

Il reste 504 cellules non-frontière sur 971 cellules générées.

classif

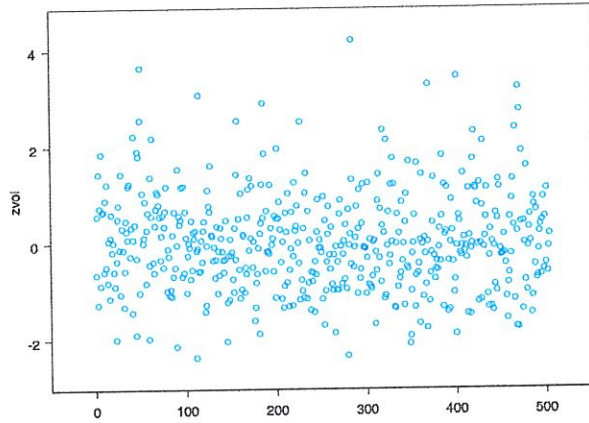


FIG. 6.6 – Scatterplot de $zvol$ pour des données sur $[0, 1]^3$ où on a enlevé une sphère de points au centre.

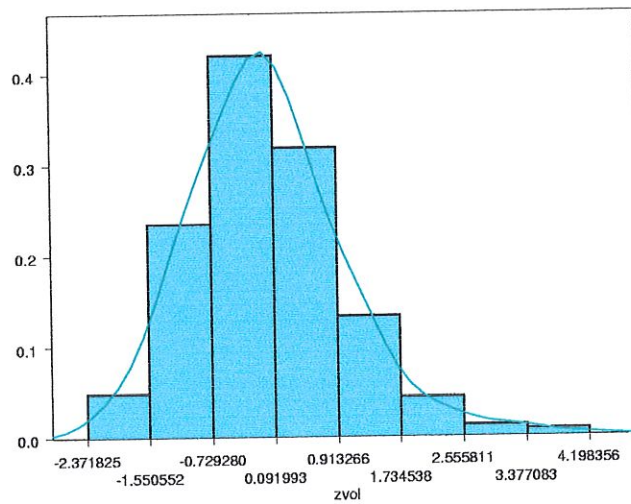


FIG. 6.7 – histogramme et densité de $zvol$ pour des données sur $[0, 1]^3$ où on a enlevé une sphère de points au centre.

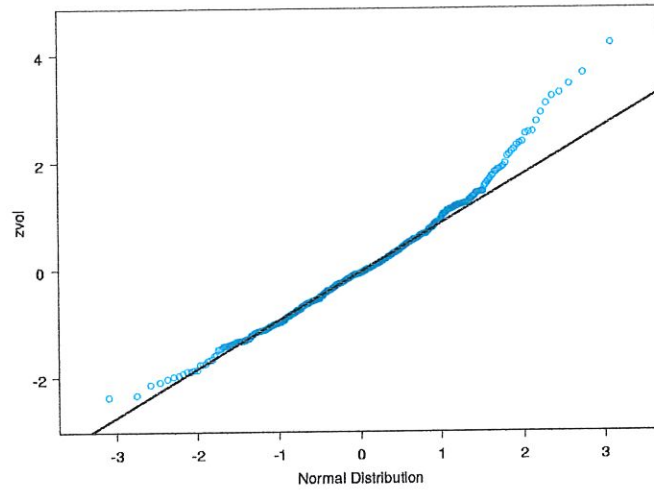


FIG. 6.8 – Q-Qplot avec ligne de densité de $zvol$ pour des données sur $[0, 1]^3$ où on a enlevé une sphère de points au centre.

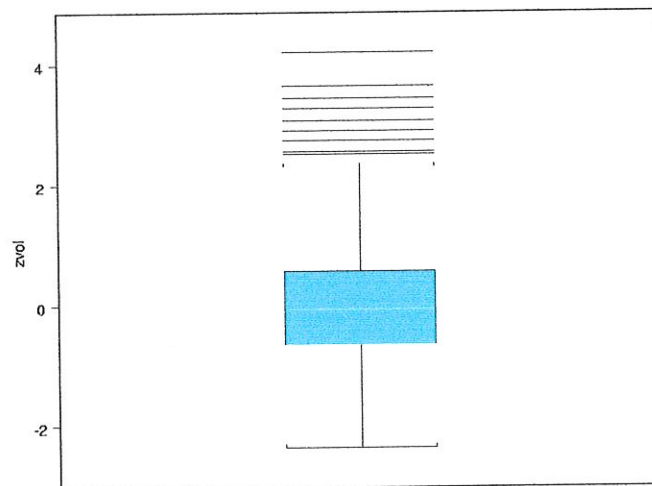


FIG. 6.9 – Boxplot de $zvol$ pour des données sur $[0, 1]^3$ où on a enlevé une sphère de points au centre.

Comme au paragraphe précédent, on peut remarquer qu'un certain nombre de cellules ont un volume plus grand.

Mis à part ces quelques valeurs que l'on peut voir sur le boxplot et le QQplot, il n'y a pas de grande différences par rapport aux données sur $[0, 1]^3$ étudiées au paragraphe 5.3.

Appliquons les tests de Kolmogorov-Smirnov et du chi-carré à notre variable z_{vol} en enlevant progressivement les cellules de plus grand volume.

	Toutes les cellules	- les 5 + grandes	- les 10 + grandes
KS	0.0495	0.0374	.0284
p-value	0.1695	0.4876	0.8211
χ^2	28.0437	23.2946	19.3279
p-value	0.2582	0.5025	0.682

En enlevant les 10 cellules qui ont un volume plus important, nous retrouvons l'uniformité des données.

6.3 Données de type sphérique

6.3.1 En dimension 2 : cas d'un disque

96 cellules non-frontières sur 112.

Les données analysées sont :

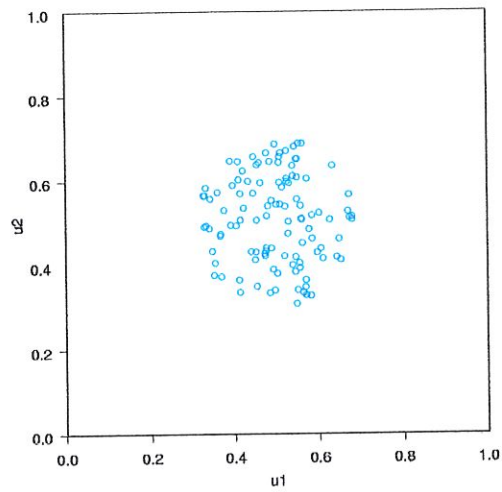


FIG. 6.10 – Points générateurs des cellules

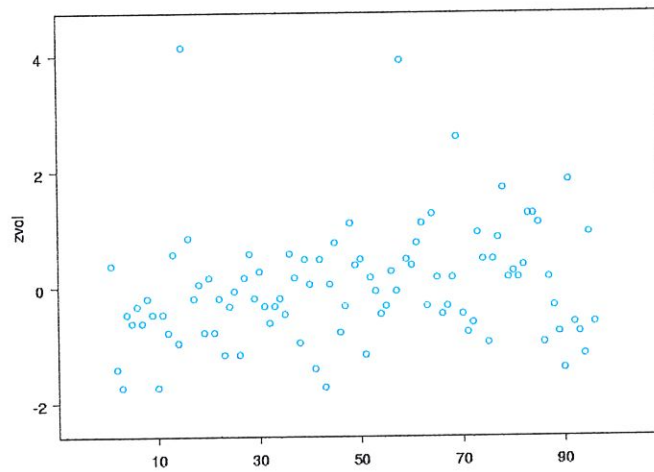


FIG. 6.11 – Scatterplot de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.10.

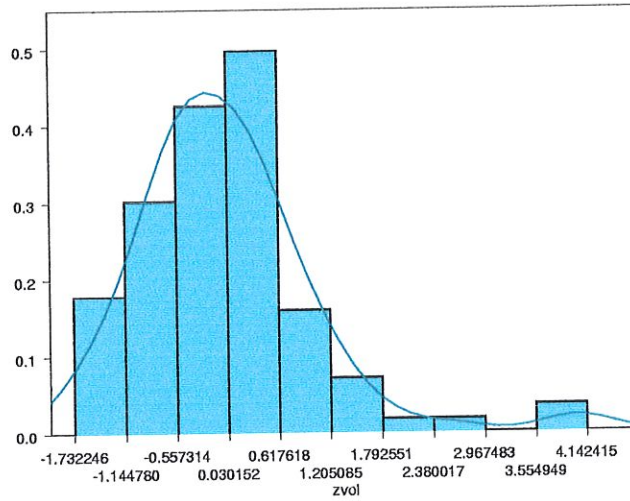


FIG. 6.12 – Histogramme et densité de z_{vol} des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.10.

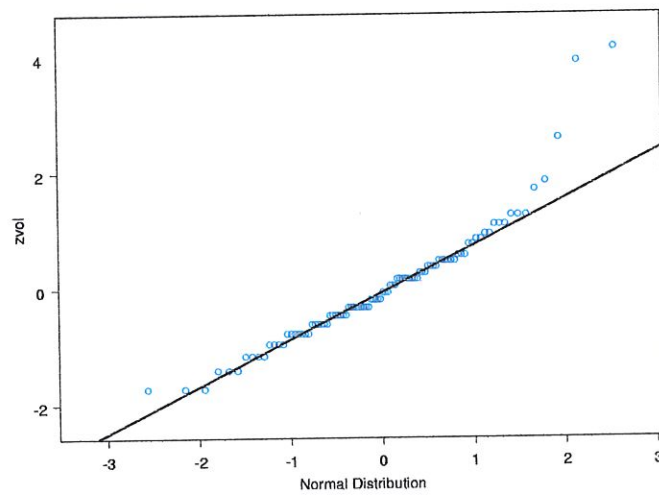


FIG. 6.13 – QQplot avec ligne de densité de z_{vol} des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.10

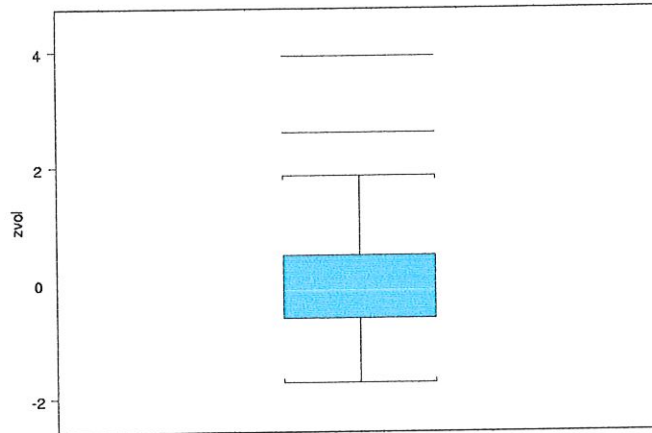


FIG. 6.14 – Boxplot de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.10.

On remarque quelques cellules de plus grand volume. Des résultats qui, si on tient compte de la différence de proportion de points (112 points générés ici 971 dans l'autre cas), ne diffèrent pas beaucoup des résultats du paragraphe 6.1.

Nous appliquons les tests de Kolmogorov-Smirnov et du chi-carré à notre variable $zvol$ en enlevant les 5 cellules de plus grand volume.

	Toutes les cellules	- les 5 + grandes
KS	0.1105	0.0738
p-value	0.1919	0.7043
χ^2	23.4375	4.2857
p-value	0.0242	0.9777

Si nous enlevons les 5 cellules de plus grand volume, nous retrouvons l'uniformité des données.

6.3.2 En dimension 3 : cas de la sphère

Nous gardons les 60 cellules non-frontières sur 106 pour l'analyse.

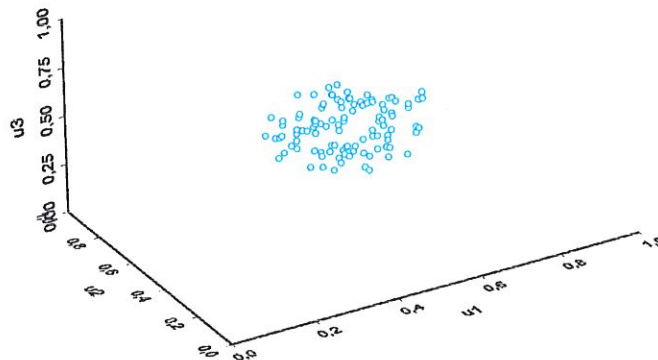


FIG. 6.15 – Points générateurs des cellules

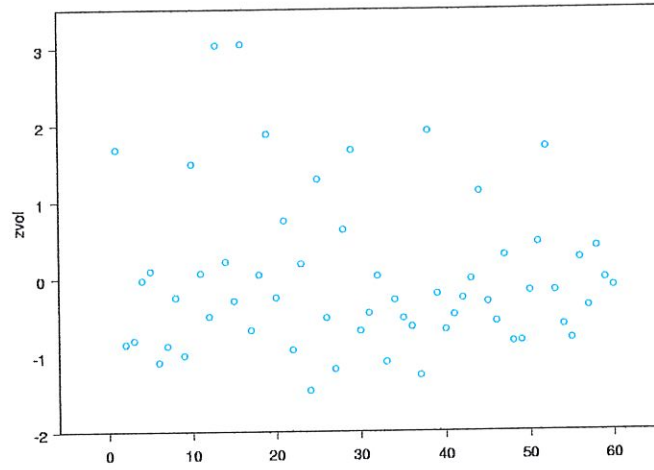


FIG. 6.16 – Diagramme de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.15.

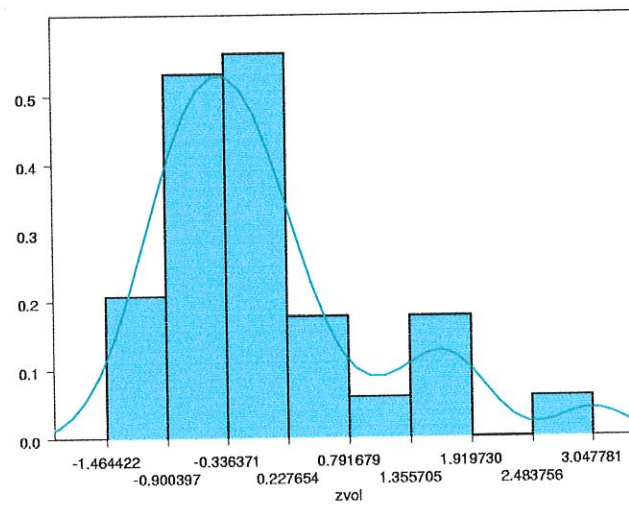


FIG. 6.17 – Histogramme et densité de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.15.

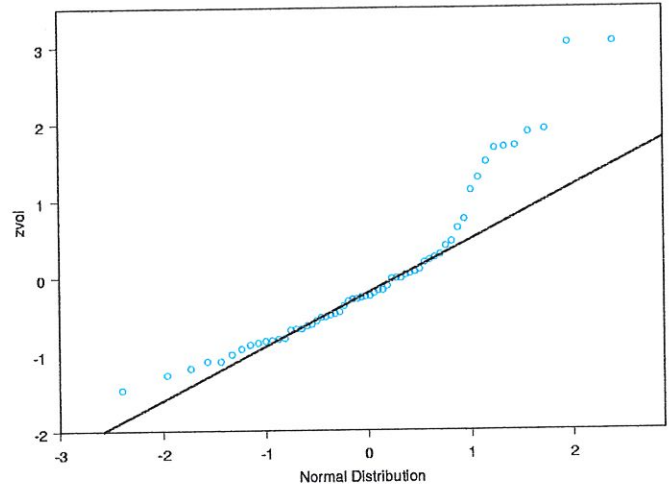


FIG. 6.18 – QQplot avec ligne de densité de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.15

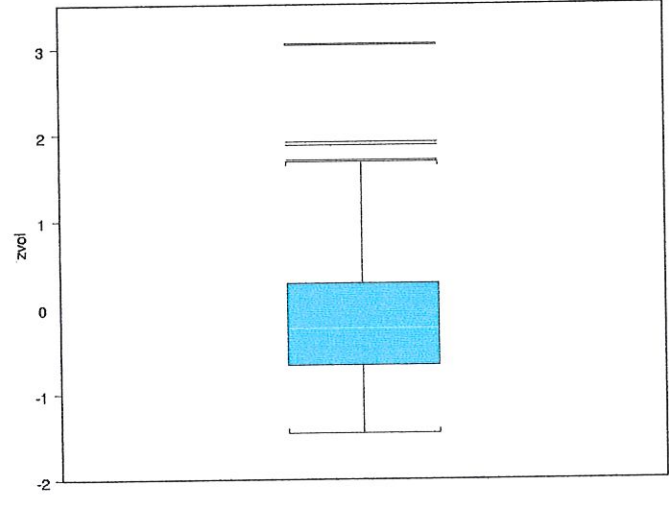


FIG. 6.19 – Boxplot de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.15.

Les résultats sont semblables à ceux fournis par les simulations en dimension 2.

Nous appliquons les tests de Kolmogorov-Smirnov et du chi-carré à notre variable z_{vol} en enlevant les 5 cellules de plus grand volume.

	Toutes les cellules	- les 5 + grandes
KS	0.1619	0.099
p-value	0.0859	0.6532
χ^2	22.1333	7
p-value	0.0144	0.6371

A nouveau, en supprimant les 5 cellules de plus grand volume nous retrouvons l'uniformité des données.

6.3.3 En dimension 2 : cas de 2 disques

Nous analysons les 234 cellules non-frontière sur 259.

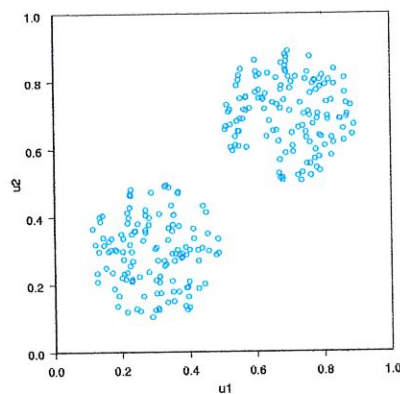


FIG. 6.20 – Points générateurs des cellules

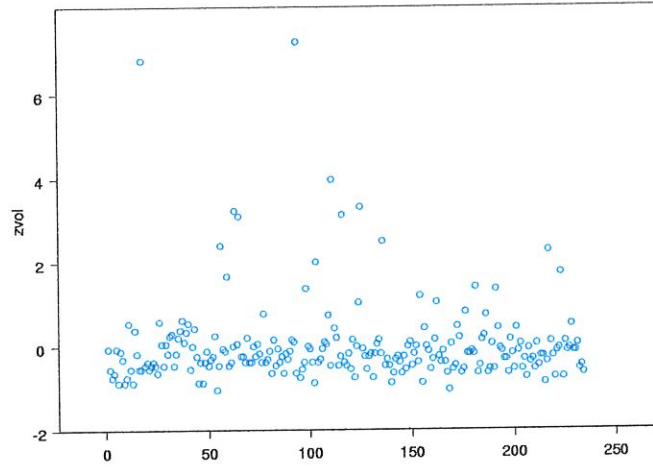


FIG. 6.21 – Diagramme de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.20.

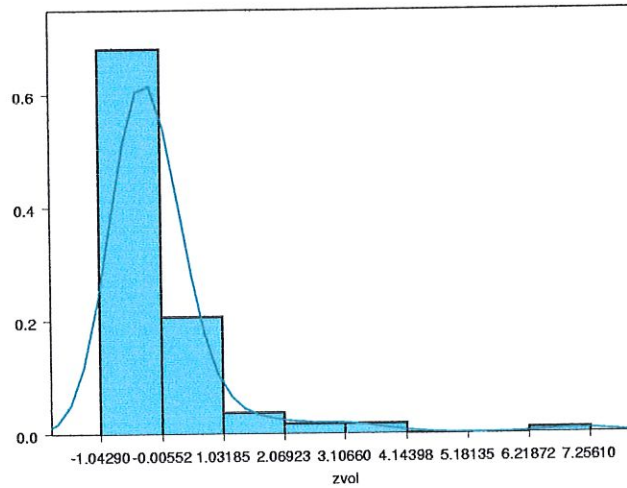


FIG. 6.22 – Histogramme et densité de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.20.

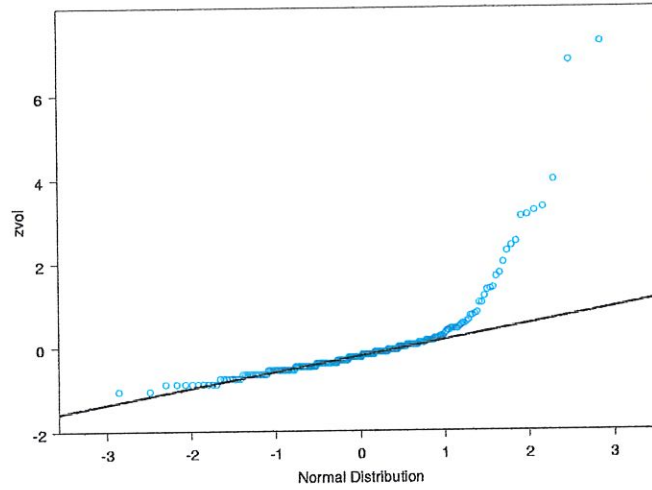


FIG. 6.23 – QQplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.20

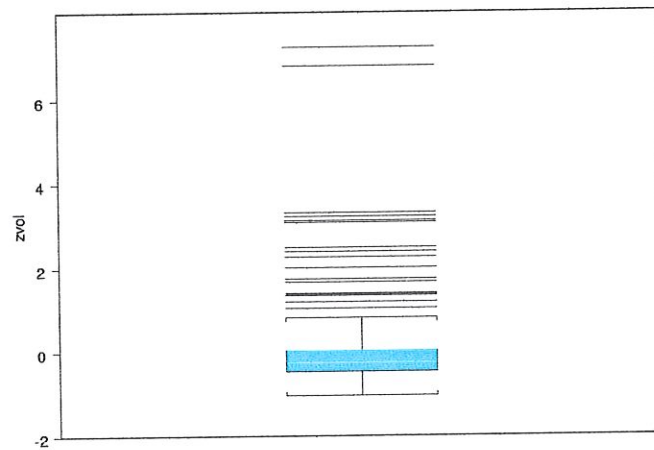


FIG. 6.24 – Boxplot de $zvol$ des cellules non-frontière en dimension 2 pour les données de type sphérique de la figure 6.20.

Encore une fois, on peut voir que l'on n'a pas l'uniformité par le fait que quelques cellules ont un volume plus grand.

On remarque particulièrement deux cellules beaucoup plus grosses que les autres. Cela ne semble pas être une caractéristique propre à ce type de donnée. Nous n'avons pas d'explications formelle, il pourrait s'agir de valeurs aberrantes liées à ce jeu de données particulier.

Nous appliquons les tests en retirant progressivement jusqu'à 30 cellules de plus grand volume.

	Toutes les cellules	- les 10 + grandes	- les 20 + grandes	- les 25 + grandes	- les 30 + grandes
KS	0.2479	0.1148	0.0828	0.0802	0.0768
p-value	0	0.0055	0.1062	0.136	0.18
χ^2	210.6154	44.875	53.3084	48.1148	45.5
p-value	0	0.0003	0	0	0.0001

Il faut ici retirer une trentaine de cellules de volume extrême pour retrouver l'uniformité des données.

6.3.4 En dimension 3 : cas de 2 sphères

Il y a 119 cellules non-frontière sur 201 cellules.

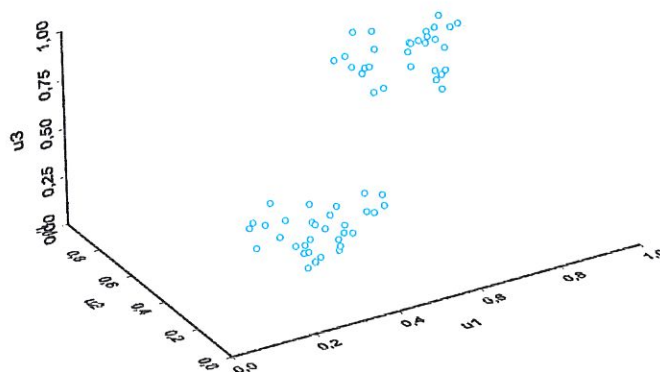


FIG. 6.25 – Points générateurs des cellules

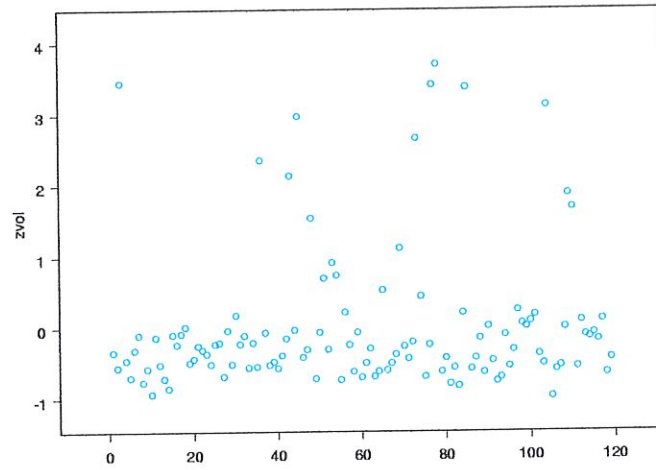


FIG. 6.26 – Diagramme de $zvol$ des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.25.

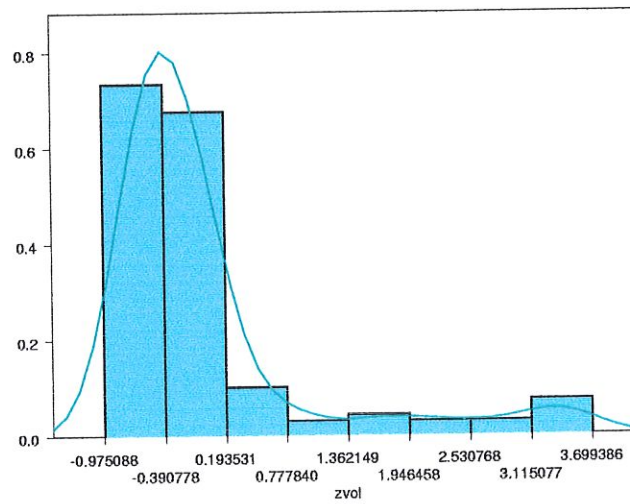


FIG. 6.27 – Histogramme et densité de $zvol$ des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.25.

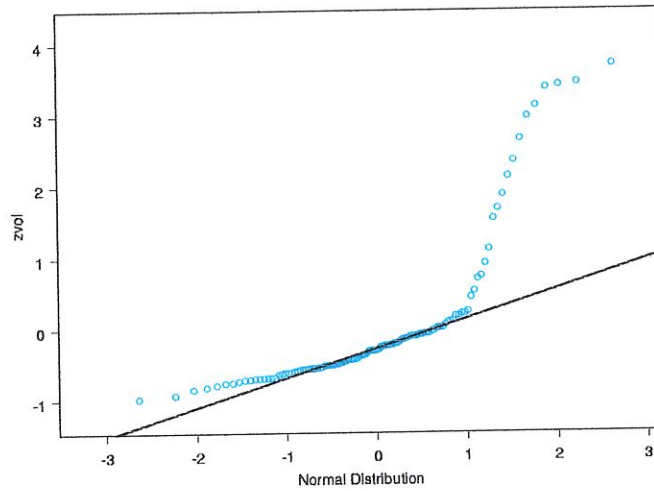


FIG. 6.28 – Q-Qplot avec ligne de densité de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.25

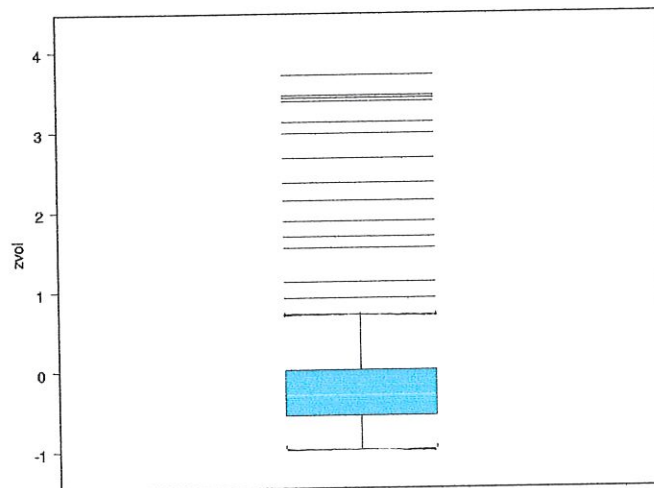


FIG. 6.29 – Boxplot de zvol des cellules non-frontière en dimension 3 pour les données de type sphérique de la figure 6.25.

Les graphes sont semblables à ceux trouvés en dimension deux. Mis à part qu'ici, les deux valeurs aberrantes ont disparu.

Si nous appliquons les tests du chi-carré et de Kolmogorov-Smirnov en enlevant progressivement les cellules de plus grand volume, nous obtenons les résultats :

	Toutes les cellules	- les 10 + grandes	- les 15 + grandes	- les 20 + grandes
KS	0.267	0.12	0.0842	0.0805
p-value	0	0.0867	0.4524	0.5422
χ^2	127.4706	27.789	10.75	13.798
p-value	0	0.0097	0.5505	0.3138

Nous constatons qu'en enlevant la vingtaine de cellules de volume plus important, nous avons à nouveau l'uniformité des données.

6.4 En dimension 2 pour des données de type couronne

Nous avons retenu les 355 cellules non-frontière pour l'analyse.

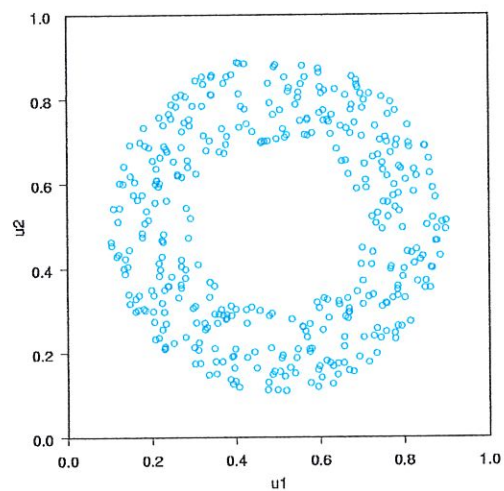


FIG. 6.30 – Points générateurs des cellules

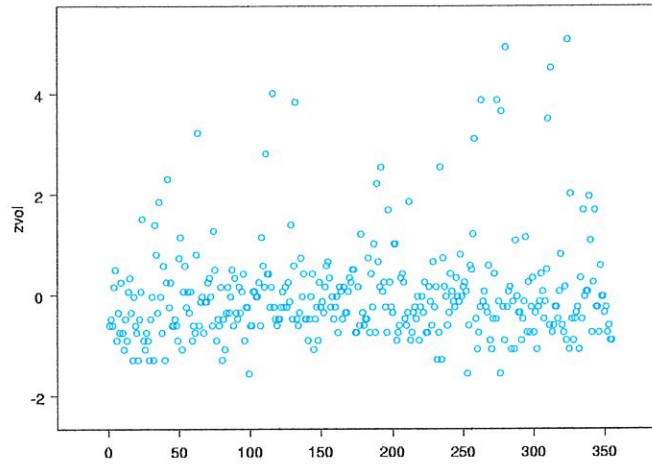


FIG. 6.31 – Diagramme de zvol des cellules non-frontière en dimension 2 pour les données de type couronne de la figure 6.30.

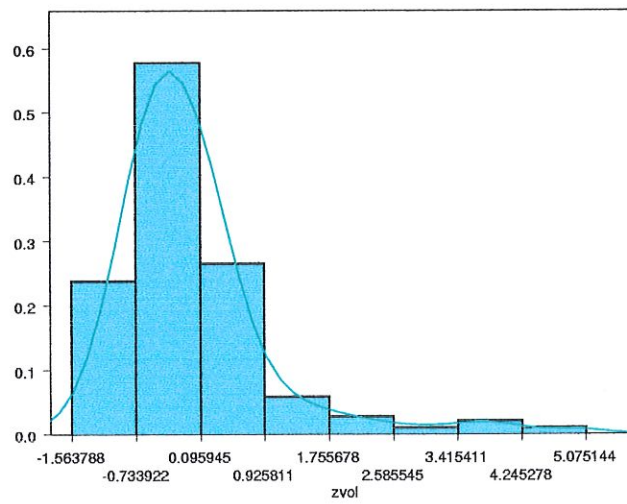


FIG. 6.32 – Histogramme et densité de zvol des cellules non-frontière en dimension 2 pour les données de type couronne de la figure 6.30.

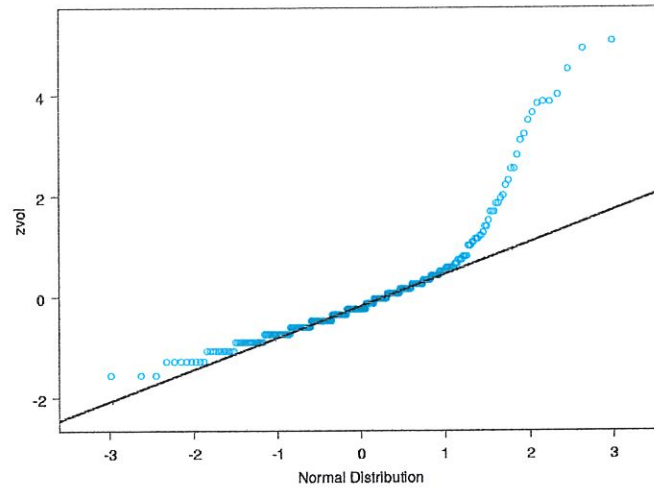


FIG. 6.33 – Q-Qplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 2 pour les données de type couronne de la figure 6.30

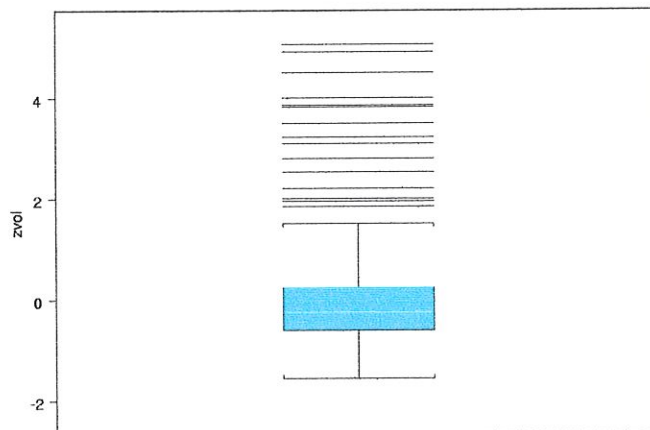


FIG. 6.34 – Boxplot de $zvol$ des cellules non-frontière en dimension 2 pour les données de type couronne de la figure 6.30.

Encore une fois, nous voyons des cellules de volume plus important. Signe que les données ne sont pas uniformes.

Voyons ce que nous apprennent les tests de Kolmogorov-Smirnov et χ^2 .

	Toutes les cellules	- les 10 + grandes	- les 20 + grandes	- les 25 + grandes	- les 30 + grandes
KS	0.171	0.1043	0.0869	0.0784	0.0711
p-value	0	0.0011	0.0127	0.0346	0.0748
χ^2	142.0789	114.0174	124.4299	129.7091	143.0092
p-value	0	0	0	0	0

Lorsque nous retirons la trentaine de cellules de plus grand volume, les p-value sont faibles mais la statistique de Kolmogorov est petite. Au paragraphe 5.2 (test d'uniformité sur $[0, 1]^2$), les valeurs étaient semblables et nous avons conclu à l'uniformité des données. Nous pouvons donc dire qu'en retirant 30 cellules de plus grand volume, nous retrouvons l'uniformité des données.

Dans les quelques cas qui précèdent, grâce aux quelques cellules de plus grand volume, nous avons à chaque fois une idée sur l'uniformité ou la non-uniformité des données. Une des caractéristiques intéressantes de la distribution de la taille des cellules de Voronoi est qu'elle indique de la même façon l'ensemble des alternatives à l'uniformité, pour autant que la densité de points soit uniforme.

Nous avons alors voulu voir si les cellules frontières ne nous donnaient pas des renseignements sur le type de données. On remarque que les graphiques ne semblent donner aucune information vraiment pertinente.

Par exemple, voici le QQplot pour les 49 cellules frontières en dimension 2 pour les données de type couronne.

6.5 En dimension 3 pour des données de type couronne

Nous avons simulé des données de type couronne en dimension 3, mais les résultats ont été surprenants. Nous avons gardé 128 cellules non-frontières sur 226.

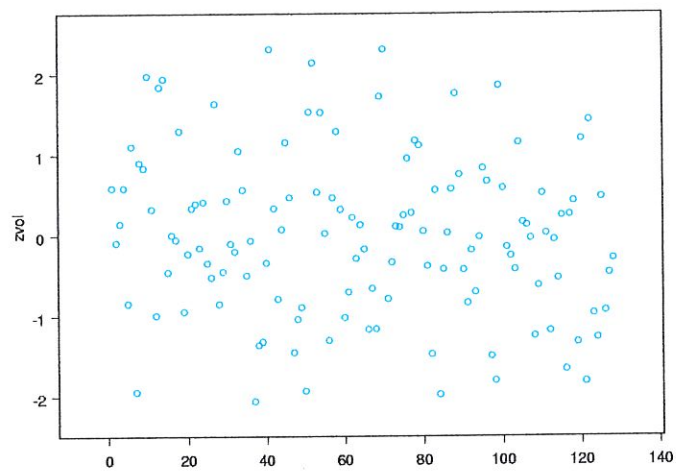


FIG. 6.36 – Diagramme de zvol des cellules non-frontière en dimension 3 pour les données de type couronne .

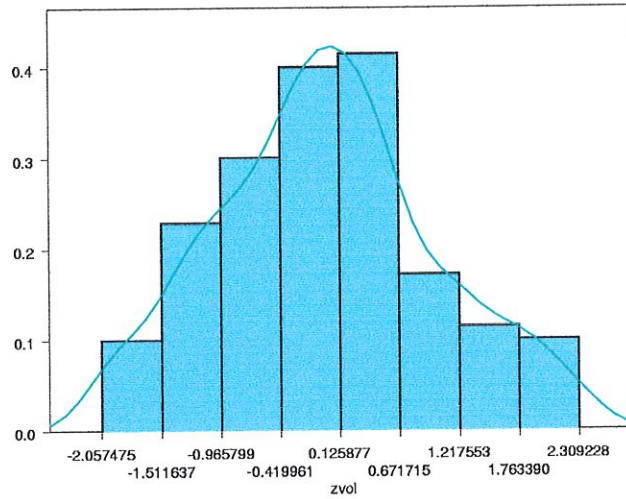


FIG. 6.37 – Histogramme et densité de z_{vol} des cellules non-frontière en dimension 3 pour les données de type couronne .

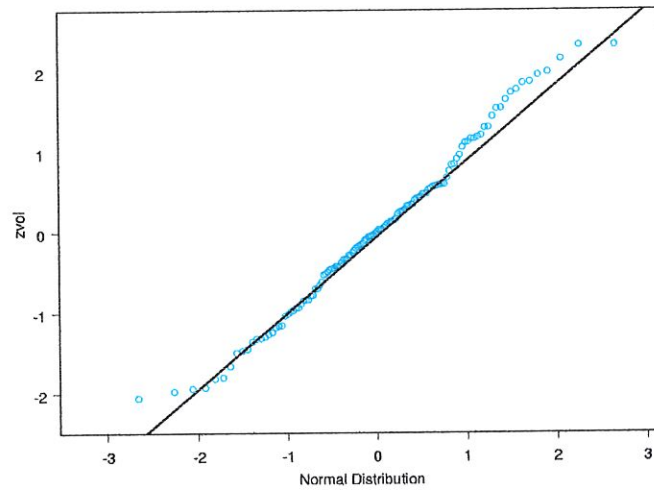


FIG. 6.38 – QQplot avec ligne de densité de z_{vol} des cellules non-frontière en dimension 3 pour les données de type couronne

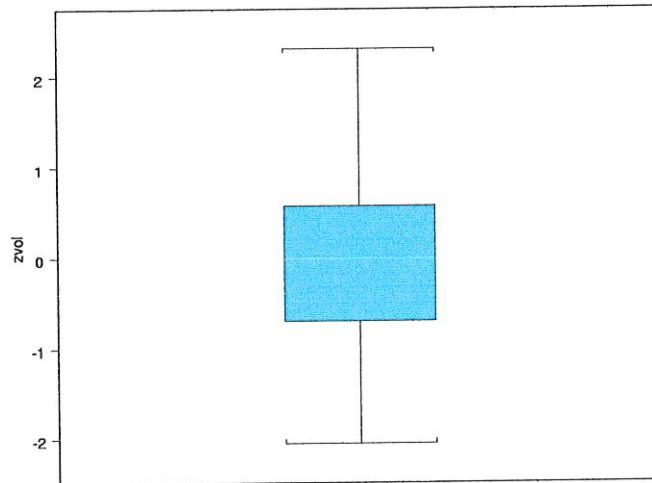


FIG. 6.39 – Boxplot de $zvol$ des cellules non-frontière en dimension 3 pour les données de type couronne .

Nous avons ici plus de mal à juger graphiquement de l'uniformité des données. Nous ne retrouvons pas de valeurs vraiment outliers comme précédemment. Le courbe de densité n'est pas normale mais a cependant une allure plutôt normale. Si nous ne savions pas que les données ne sont pas uniformes, en regardant uniquement les graphiques il nous aurait été difficile de prendre une décision sur l'uniformité des données.

Appliquer les tests de Kolmogorov-Smirnov et du chi-carré en enlevant les cellules de volume extrême n'a pas de sens ici. En gardant toutes les cellules la p-value du test de Kolmogorov est supérieure à 0.9 et celle du test chi-carré supérieure à 0.7. De plus, graphiquement, nous voyons bien qu'il n'y a pas de cellule de plus grand volume.

6.6 En dimension 2 pour des données de type 2 bandes horizontales

Nous analysons les données de la figure 6.40. Il y a 345 cellules non-frontière sur 394.

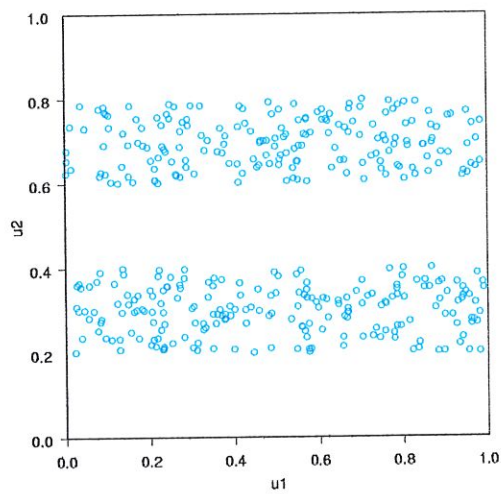


FIG. 6.40 – Points générateurs des cellules

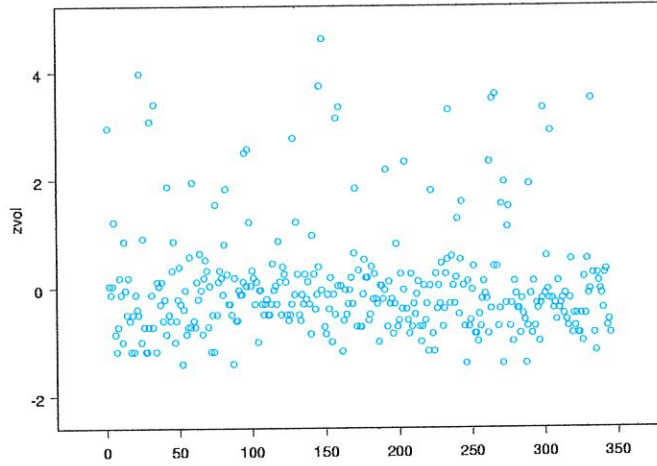


FIG. 6.41 – Diagramme de zvol des cellules non-frontière en dimension 2 pour les données de type 2 bandes horizontales de la figure 6.40.

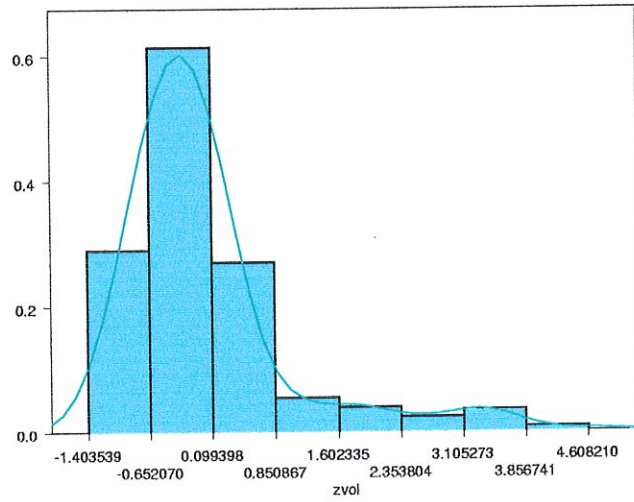


FIG. 6.42 – Histogramme et densité de zvol des cellules non-frontière en dimension 2 pour les données de type 2 bandes horizontales de la figure 6.40.

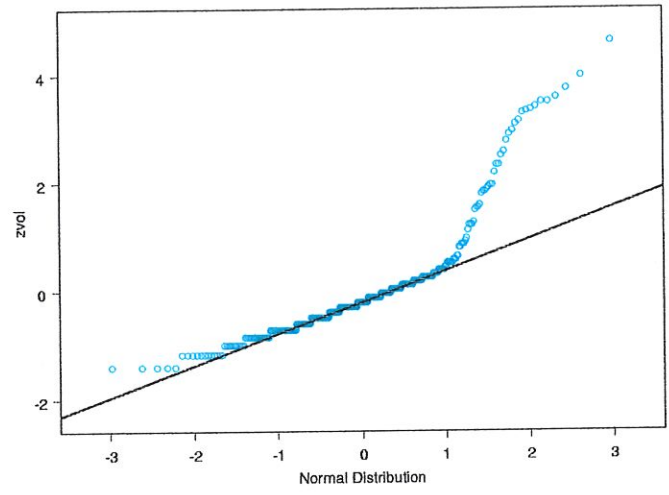


FIG. 6.43 – QQplot avec ligne de densité de *zvol* des cellules non-frontière en dimension 2 pour les données de type 2 bandes horizontales de la figure 6.40

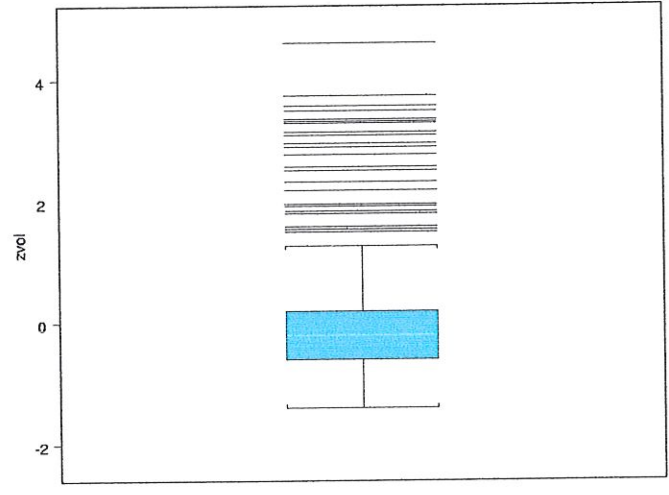


FIG. 6.44 – Boxplot de *zvol* des cellules non-frontière en dimension 2 pour les données de type 2 bandes horizontales de la figure 6.40.

A nouveau, nous pouvons dire que les données ne sont pas uniformes grâce aux valeurs extrêmes.

Les tests de Kolmogorov-Smirnov et χ^2 appliqués à la variable z_{vol} en retirant progressivement des cellules de volume plus important donnent :

	Toutes les cellules	- les 10 + grandes	- les 20 + grandes	- les 25 + grandes	- les 30 + grandes
KS	0.1994	0.1464	0.0856	0.0623	0.0557
p-value	0	0	0.017	0.1673	0.2816
χ^2	218.2261	107.003	73.4831	109.3188	85.1905
p-value	0	0	0	0	0

Lorsque nous retirons la trentaine de cellules de plus grand volume, les p-value sont faibles mais la statistique de Kolmogorov est petite. Au paragraphe 5.2 (test d'uniformité sur $[0, 1]^2$), les valeurs étaient semblables et nous avons conclu à l'uniformité des données. Nous pouvons donc dire qu'en retirant 30 cellules de plus grand volume, nous retrouvons l'uniformité des données.

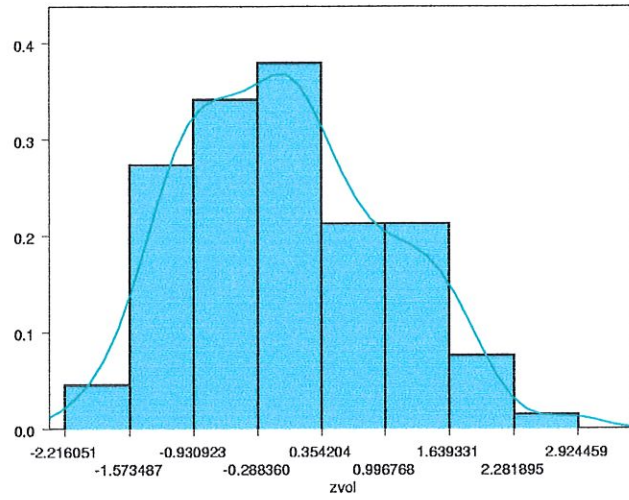


FIG. 6.46 – Histogramme et densité de $zvol$ des cellules non-frontière en dimension 3 pour les données de type 2 bandes horizontales.

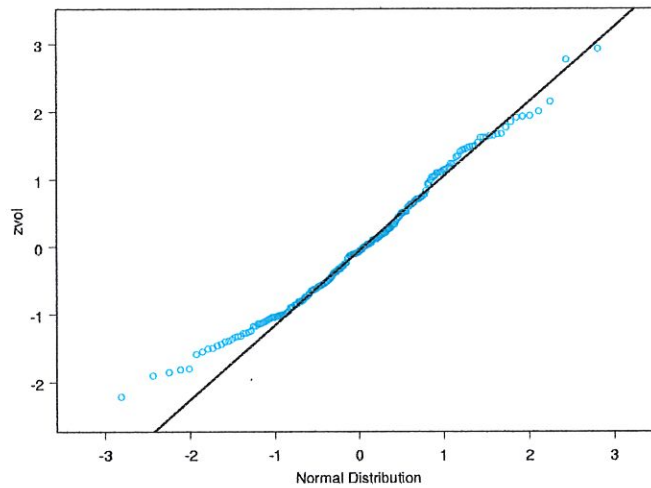


FIG. 6.47 – QQplot avec ligne de densité de $zvol$ des cellules non-frontière en dimension 3 pour les données de type 2 bandes horizontales.

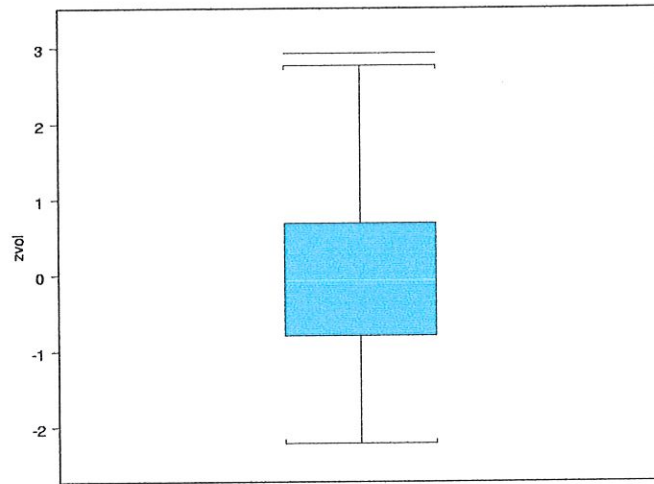


FIG. 6.48 – Boxplot de $zvol$ des cellules non-frontière en dimension 3 pour les données de type 2 bandes horizontales.

Nous ne voyons pas vraiment d'analogie avec la dimension 2.

On constate un diagramme de densité asymétrique qui indique des cellules plus volumineuses. La différence de volume est moins marquée que dans les cas précédents. Ces cellules sont en plus grand nombre que dans les situations antérieures si bien qu'on ne les remarque pas sur le boxplot.

Le fait d'être en dimension 3 diminue fortement le nombre de cellules non-frontière et donc l'intérêt du test sur les valeurs extrêmes. De plus, la différence de volume étant moins marquée, les tests en retirant des cellules de plus grand volume ne nous apprennent rien de supplémentaire sur les données.

Conclusions

Après analyse des différentes simulations, nous pouvons dire que se servir des tessellations de Voronoi pour tester l'uniformité d'un ensemble de points n'est pas sans intérêt.

Les simulations nous ont montré que si les données sont réparties uniformément, la distribution de z_{vol} est fort proche. Nous rappelons que z_{vol} est le volume exposant l'inverse de la dimension centré et réduit. En faisant le chemin inverse, nous pouvons donc conclure que le volume suit une distribution gamma, ce qui est en accord avec les résultats trouvés par différents chercheurs.

Lorsque les données ne sont pas uniformes, nous constatons des valeurs extrêmes pour le volume. Nous pouvons donc, dans la majorité des cas, détecter quand des données ne sont pas distribuées uniformément mais nous ne savons rien dire sur leur configuration. Rien dans nos résultats ne permet de différencier les données de type couronne des données de type bande horizontale par exemple.

Cependant, le nombre de valeurs extrêmes semblent un bon candidat pour un nouveau test non-paramétrique d'uniformité. Evidement, d'autres recherches plus approfondies sont nécessaires pour mieux comprendre le comportement des grandes cellules de Voronoi.

Lors de notre travail, nous avons rencontré deux difficultés. La première d'ordre mathématique. Dès que la dimension est plus grande que quatre il y a une grande proportion de cellules frontière par rapport aux cellules non-frontière. Or les cellules frontières, nous l'avons vu, empêche une analyse correcte du volume. La deuxième difficulté est d'ordre informatique. Notre implémentation n'est pas assez efficace. Quand la dimension est

supérieure à quatre, le programme est trop lent pour donner des résultats dans les quarante-huit heures. Le temps de calcul est également un problème lorsque l'on veut générer plus de trois mille cellules en dimension quatre.

Pour résoudre le premier problème, on peut envisager des espaces de type torique, où il n'y a plus le problème des cellules frontière car l'espace est clos sur lui-même : le segment de droite est remplacé par un cercle, le pavé unité par un tore, etc.

Une façon de résoudre le second problème serait d'utiliser une moins bonne approximation des cellules. Pour les approximer, nous générons des points-grille. En dimension trois, nous simulons environ un million de points-grille, en dimension quatre plus de cent millions ! L'idée serait de ne plus générer autant de points-grille. On pourrait par exemple en générer un million aléatoirement distribué dans l'hypercube.

Annexes

Plusieurs programmes ont été implémentés pour les simulations, nous donnons le code pour deux d'entre eux, représentatifs de tous les autres.

Le premier permet d'approximer les cellules de Voronoi sur $[0, 1]^3$. Les programmes pour les autres dimensions sont semblables, il suffit de rajouter des boucles pour générer les points-grilles et de changer la dimension dans les constantes.

Le second calcule les cellules dans le cas où les points générateurs sont situés dans un sous-espace de \mathbb{R}^3 (ici une sphère). Il peut facilement être généralisé dans une autre dimension comme expliqué précédemment. Pour faire varier la forme du sous-espace des points-générateurs, il suffit de modifier les dernières constantes ou d'en rajouter et d'adapter les lignes 123-158.

```

/*+++++++*/
/* Programme permettant d'approximer les cellules de Voronoi */
/* dans [0,1]^3 */
/*+++++++*/

#include <STDIO.H>
#include <STDLIB.H>
#include <MATH.H>
#include <FSTREAM.H>

#define a 16807
#define q 127773
#define r 2836
#define m 2147483647
#define xm (1.0/m)

#define k 100
#define dim 3
#define nbuni 1000
#define taille (dim*nbuni)

/*******/
/* fonction permettant de générer des nombres aléatoires : */
/* utilisation d'un générateur congruentiel */
/*******/

float random(float x)
{
    int x_mod_q;
    float x_div_q;
    int k;

    k=0;

    /* implémentation du générateur congruentiel en évitant
    les overflows */

    x_div_q=x/q;
    k=x/q;
    x_mod_q=x-(k*q);
    x=a*x_mod_q-r*x_div_q;

    if(x<0) {
        x=x+m;
    }

    return x;
}

/*******/
/* programme principal */
/*******/

main()
{
    float x_0;
    float x;
    float U[taille];
    int i;

    float i1,i2,i3;
    float G[dim];

    int F[nbuni]; /* =0 si pas cellule frontière, autre chose si oui*/

    float V[nbuni]; /* contient le nombre de points assignés à chaque
    cellule */

    float vol[nbuni];

    float moy_dis[nbuni];
    float dis_min[nbuni];
    float dis_max[nbuni];

    double d;
    double d_temp;
    int j;

```

```

int label;

int front;
int somme;

ofstream result("resultats3D+.txt");
ofstream cel("sansfront3D+.txt");

result << "Le nombre de dimensions est: " <<dim << "\n";
result << "Le nombre de points uniformes générés est : " << nbuni << "\n";

/* initialisation d'un tableau contenant 3000 nombres aléatoires */
/* uniformément distribués */

i=1;

x_0=random(123456987);

U[0]=x_0*xm;
x=x_0;

while(i!=taille){
    U[i]=random(x);
    x=U[i];
    U[i]=x*xm;
    i++;
}

/*initialisation du tableau V[i] de taille 1000. Chaque case
représente le nombre de points grille associés au ième point
générés.
*/

i=0;
for(i=0; i<nbuni; i++){
    V[i]=0;
    F[i]=0;
    moy_dis[i]=0;
    dis_min[i]=100000;
    dis_max[i]=0;
}

i1=0;
i2=0;
i3=0;

/* génération des points-grille */

for(i3=0; i3<k+1;i3++){
    for(i2=0; i2<k+1; i2++){
        for(i1=0; i1<k+1; i1++){
            front=0;
            G[0]=i1/k;
            G[1]=i2/k;
            G[2]=i3/k;

            /* Approximation des cellules de Voronoi */

            /*détection des points frontières*/

            for(i=0;i<dim;i++){
                if(G[i]==0.0 || G[i]==1.0)
                    front=1;
            }

            /* calcule de la distance avec les pts générés */
            j=0;

            d_temp=10000000;
            label=0;
            while(j!=taille){

                d=0;
                for(i=0;i<dim;i++){

```

```

        d=d+((G[i]-U[i+j])*(G[i]-U[i+j]));
    }

    if (d<d_temp){
        d_temp=d;
        label=j/3;
    }
    j=j+3;
}
moy_dis[label]=moy_dis[label]+d;
V[label]=V[label]+1;
if(front==1){
    F[label]=F[label]+1;
}
/* calcul de la distance entre le point générateur
et le point le plus proche du point générateur */
if(dis_min[label]>d){
    dis_min[label]=d;
}
/* calcul de la distance entre le point générateur
et le point le plus éloigné du point générateur */
if(dis_max[label]<d){
    dis_max[label]=d;
}

    }
    i1=0;
}
i2=0;
}

front=0;
somme=0;
for(i=0;i<nbuni;i++){
    front=front+F[i];
    somme=somme+V[i];
}

/*Calcul du volume de chaque cellule*/
for(i=0;i<nbuni;i++){
    vol[i]=(V[i]/somme);
}

/*calcul de la distance moyenne du point générateur
à tous les points de la cellule*/
for(i=0;i<nbuni;i++){
    moy_dis[i]=(moy_dis[i]/V[i]);
}

/* Impression des résultats dans le fichier resultats3D.txt*/
for(i=0; i<nbuni;i++){
    result << "V[" <<i <<"]=" <<V[i];
    result << "\t F["<<i <<"]=" << F[i];
    result << "\t vol["<< i<<"]=" <<vol[i];
    result << "\t dis_min["<<i<<"]=" <<dis_min[i];
    result << "\t dis_max["<<i<<"]=" <<dis_max[i];
    result << "\t\t moy_dis["<<i<<"]=" <<moy_dis[i] <<"\n";
    if(F[i]==0){
        cel << vol[i] <<"\n";
    }
}

result << "nombre de point-frontière :" << front << "\n";
result << "nombre de points-grille :" << somme << "\n";

result.close();
cel.close();

return(0);
}

```



```

/*****
/* Programme permettant d'approximer les cellules de Voronoi */
/* dont les points générateurs sont situés dans une sphère */
/* dans [0,1]^3 */
/*****

#include <STDIO.H>
#include <STDLIB.H>
#include <MATH.H>
#include <FSTREAM.H>

#define a 16807
#define q 127773
#define r 2836
#define m 2147483647
#define xm (1.0/m)

#define k 100
#define dim 3
#define nbuni 1000
#define taille (dim*nbuni)
#define c1 0.5
#define c2 0.5
#define c3 0.5
#define rayon 0.2

/*****
/* fonction permettant de générer des nombres aléatoires : */
/* utilisation d'un générateur congruentiel */
/*****

float random(float x)
{
    int x_mod_q;
    float x_div_q;
    int k;

    k=0;

    /* implémentation du générateur congruentiel en évitant
    les overflows */

    x_div_q=x/q;
    k=x/q;
    x_mod_q=x-(k*q);
    x=a*x_mod_q-r*x_div_q;

    if(x<0) {
        x=x+m;
    }

    return x;
}

/*****
/* programme principal */
/*****

main()
{
    float x_0;
    float x;
    float U[taille];
    float T[taille];
    int i;

    float i1,i2,i3;
    float G[dim];

    int F[nbuni]; /* =0 si pas cellule frontière, autre chose si oui*/

    float V[nbuni]; /* contient le nombre de points assignés à chaque
    cellule */

    float vol[nbuni];

    float moy_dis[nbuni];

```



```

float dis_min[nbuni];
float dis_max[nbuni];

double d;
double d_temp;
int j;
int label;

int front;
int somme;

int compteur;
int nbuni2;

double Ddisque;

ofstream result("resultat3Ddisque.txt");
ofstream cel("sansfront3Ddisque.txt");

result << "Le nombre de dimensions est: " <<dim << "\n";
result << "Le nombre de points uniformes générés est : " << nbuni << "\n";

/*****
/* initialisation d'un tableau contenant 3000 nombres aléatoires */
/* uniformément distribués */
*****/

i=1;

x_0=random(123456987);

U[0]=x_0*xm;
x=x_0;

while(i!=taille){
    U[i]=random(x);
    x=U[i];
    U[i]=x*xm;
    i++;
}

/*enlever des points*/
i=0;
compteur=taille;
Ddisque=0;
for(i=0;i<taille;){
    Ddisque=sqrt(((U[i]-c1)*(U[i]-c1))+((U[i+1]-c2)*(U[i+1]-c2))+((U[i+2]-c3)*(
U[i+2]-c3)));
    if(Ddisque>rayon){
        U[i]=10;
        U[i+1]=10;
        U[i+2]=10;
        compteur=compteur-3;
    }
    i=i+3;
    Ddisque=0;
}

i=0;

for(i=0;i<taille;){
    if(U[i]!=10){
        uni << U[i] << "\t " << U[i+1] << "\t" <<U[i+2] << "\n";
    }
    i=i+3;
}

nbuni2=compteur/3;

j=0;
for(i=0;i<taille;i++){
    T[i]=0;
    if(U[i]!=10){
        T[j]=U[i];

```

```

        j++;
    }
}

/*initialisation du tableau V[i] de taille 1000. Chaque case
répresente le nombre de points grille associés au ième point
générés.
*/

i=0;
for(i=0; i<nbuni; i++){
    V[i]=0;
    F[i]=0;
    moy_dis[i]=0;
    dis_min[i]=100000;
    dis_max[i]=0;
}

i1=0;
i2=0;
i3=0;

for(i3=0; i3<k+1;i3++){
    for(i2=0; i2<k+1; i2++){
        for(i1=0; i1<k+1; i1++){
            front=0;
            G[0]=i1/k;
            G[1]=i2/k;
            G[2]=i3/k;

            /* Approximation des cellules */

            /*détection des points frontières*/

            for(i=0;i<dim;i++){
                if(G[i]==0.0 || G[i]==1.0)
                    front=1;
            }

            /* calcule de la distance avec les pts générés */
            j=0;

            d_temp=10000000;
            label=0;
            while(j!=taille){

                d=0;
                for(i=0;i<dim;i++){
                    d=d+((G[i]-T[i+j])*(G[i]-T[i+j]));
                }

                if (d<d_temp){
                    d_temp=d;
                    label=j/3;
                }
                j=j+3;
            }
            moy_dis[label]=moy_dis[label]+d;
            V[label]=V[label]+1;
            if(front==1){
                F[label]=F[label]+1;
            }
            /* calcul de la distance entre le point générateur
            et le point le plus proche du point générateur */
            if(dis_min[label]>d){
                dis_min[label]=d;
            }
            /* calcul de la distance entre le point générateur
            et le point le plus éloigné du point générateur */
            if(dis_max[label]<d){
                dis_max[label]=d;
            }
        }
    }
}

```

```

        i1=0;
    }
    i2=0;
}

front=0;
somme=0;
for(i=0;i<nbuni;i++){
    front=front+F[i];
    somme=somme+V[i];
}

/*Calcul du volume de chaque cellule*/
for(i=0;i<nbuni;i++){
    vol[i]=(V[i]/somme);
}

/*calcul de la distance moyenne du point générateur
à tous les points de la cellule*/
for(i=0;i<nbuni;i++){
    moy_dis[i]=(moy_dis[i]/V[i]);
}

/* Impression des résultats dans les fichiers resultats3Ddisque.txt
et sansfront3Ddisque.txt */

for(i=0; i<nbuni2;i++){
    result << "V[" <<i <<"]=" <<V[i];
    result << "\t F["<<i <<"] = " << F[i];
    result << "\t vol["<< i<<"] = " <<vol[i];
    result << "\t dis_min["<<i<<"] = "<<dis_min[i];
    result << "\t dis_max["<<i<<"] = "<<dis_max[i];
    result << "\t\t moy_dis["<<i<<"] = "<<moy_dis[i] <<"\n";
    if(F[i]==0){
        cel << vol[i] <<"\n";
    }
}

result << "nombre de point-frontière :" << front << "\n";
result << "nombre de points-grille :" << somme << "\n";

result.close();
cel.close();

return(0);
}

```

Bibliographie

- [1] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu, *SPATIAL TESSELLATIONS, Concepts and Applications of Voronoi Diagrams*, Wiley series in probability and statistics, 2000, deuxième édition.
- [2] Brian D. Ripley, *Stochastic simulation*, Wiley series in probability and statistics, 1987.
- [3] A. Hardy, *Statistiques, syllabus de seconde candidature en sciences mathématiques FUNDP*, librairie des sciences, 2000.
- [4] J.P. Rasson, *Statistiques non paramétriques, syllabus de seconde licence en sciences mathématiques*, 1996.
- [5] *S-plus 2000 User's guide*, Data Analysis Products Division, MathSoft, Seattle, WA, 1999.
- [6] *S-plus 2000 guide to statistics vol.1 et 2*, Data Analysis Products Division, MathSoft, Seattle, WA, 1999.
- [7] Masahura Tanemura, *Random Voronoi cells of higher order*.
- [8] www.cours.polymtl.ca/if505/sim030.pdf, L. Granger.
- [9] www.itl.nist.gov/div898/eda/section3, anonymous.