

INTRODUCTION AUX TECHNIQUES DE SIMULATION

LES GENERATEURS DE NOMBRES ALEATOIRES

G. Burillon

CTFT-CIRAD

février 1989

INTRODUCTION AUX TECHNIQUES DE SIMULATION

LES GENERATEURS DE NOMBRES ALEATOIRES

G. Burillon

CTFT-CIRAD

février 1989

SOMMAIRE:

- I) Introduction: intérêts de la simulation.
- II) Qu'entend - on par simulation d'une variable aléatoire?
- III) Nature des problèmes rencontrés lors de la génération de variables aléatoires.
- IV) Lois uniformes: leur importance.
- V) Exemples: quelques méthodes de génération de lois particulières.
- VI) Différentes techniques de génération de lois uniformément réparties sur $[0,1]$: générateurs de nombres aléatoires.
- VII) Tests de qualité des générateurs.
- VIII) Que faire lorsque ces propriétés ne sont pas respectées?
- IX) Conclusion.

- Annexe 1: Notre propre expérience.
- Annexe 2: Exemples issus de l'ouvrage de Morgan.
- Annexe 3: Listings de deux générateurs (URAND et GGUBFS).
- Annexe 4: Tables de D statistique du test de Kolmogorov-Smirnov.

BIBLIOGRAPHIE.

La génération de nombres aléatoires en vue de simulation.

I) Introduction: intérêts de la simulation.

Outil relativement récent, à base d'opérations numériques, la simulation a vu son rôle grandir depuis le développement et la démocratisation des moyens de calcul. C'est une méthode d'investigation rapide qui passe par différentes étapes (modélisation, contrôle,...), et qui a pour but d'étudier des hypothèses de travail lorsqu'il est difficile de le faire sur le plan strictement théorique, ceci de manière non pas à tenter de les démontrer, mais à essayer de savoir si elles sont au moins vraisemblables ou pertinentes. Pour employer une image, c'est en quelque sorte une maquette.

Exemple d'utilisation: incidence de la violation de certaines hypothèses relatives aux techniques de l'analyse de variance, (non indépendance des résidus par exemple), sur le comportement de la statistique de Fisher classiquement utilisée dans ce cadre.

Nota: la qualité des simulations dépend de la modélisation que l'on utilise et de la faculté que l'on a de représenter le hasard.

Exemple: si l'on étudie les questions de compétition ou d'émulation entre plants voisins (arbres dans une plantation par exemple), les variables mesurées que nous qualifierons de rendements ne sont pas indépendantes, et l'on pressent que la relation de dépendance est fonction de la distance entre les individus. La façon dont on fera intervenir la distance dans la relation de dépendance constituera l'étape de modélisation et interviendra dans le conditionnement des événements que l'on générera, alors que la façon dont on représente le hasard est plutôt tributaire de l'algorithme choisi pour simuler des nombres aléatoires.

Dans toute la suite, nous supposons que la modélisation n'est pas en cause et nous intéresserons à la partie de simulation proprement dite.

II) Qu'entend on par simulation d'une variable aléatoire ?

Rappel: La loi d'une variable aléatoire X peut être caractérisée par sa fonction de répartition $F(x)$.

Définition de la fonction de répartition: $F(x)$, fonction de répartition de la variable X a les propriétés suivantes:

- a) F est croissante.
- b) $\lim_{x \rightarrow -\infty} F(x) = 0$, $\lim_{x \rightarrow \infty} F(x) = 1$.
- c) En tout point x , $F(x)$ est continue à gauche et admet une limite à droite. En termes de probabilités, si l'on désigne par $P(X < b)$ la probabilité que la variable aléatoire X soit de valeur inférieure à b , on a les propriétés suivantes:

$$P(a \leq X < b) = F(b) - F(a)$$

$$P(X < b) = F(b)$$

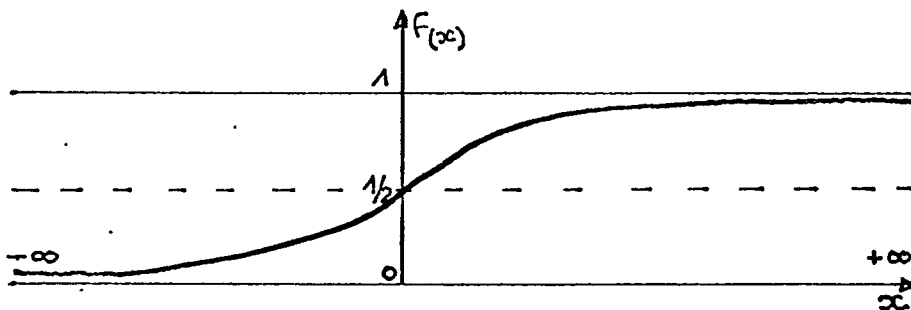
$$P(X = a) = F(a^+) - F(a)$$

Exemples:

- a) Loi de Cauchy:

$$F(x) = \frac{1}{\pi} \left(\text{Arctg}(x) + \frac{\pi}{2} \right)$$

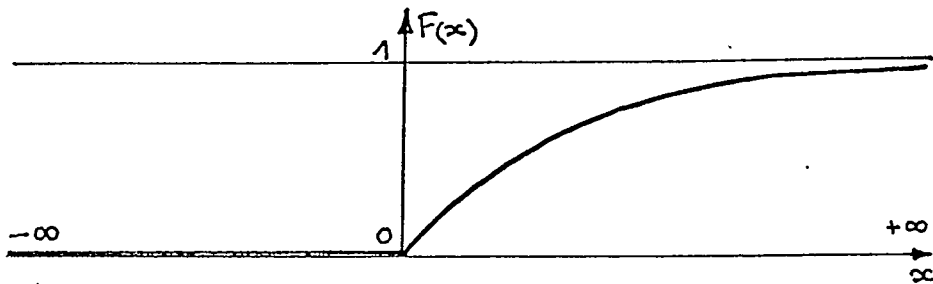
Graphe de $F(x)$.



b) Loi de Laplace:

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 - e^{-x} & \text{si } x > 0 \end{cases}$$

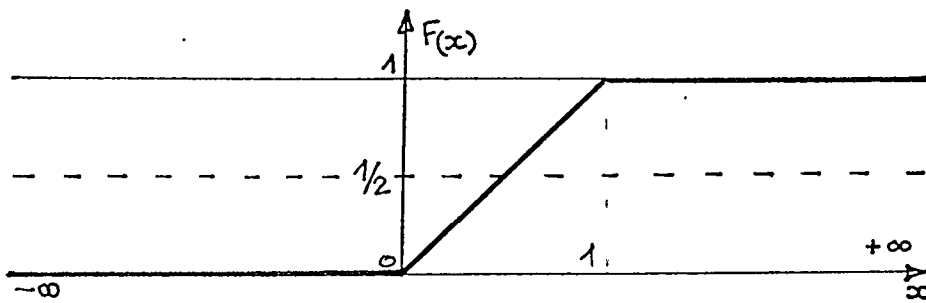
Graphe de $F(x)$.



c) Loi uniforme sur $[0,1]$:

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x < 1 \\ 1 & \text{si } x > 1 \end{cases}$$

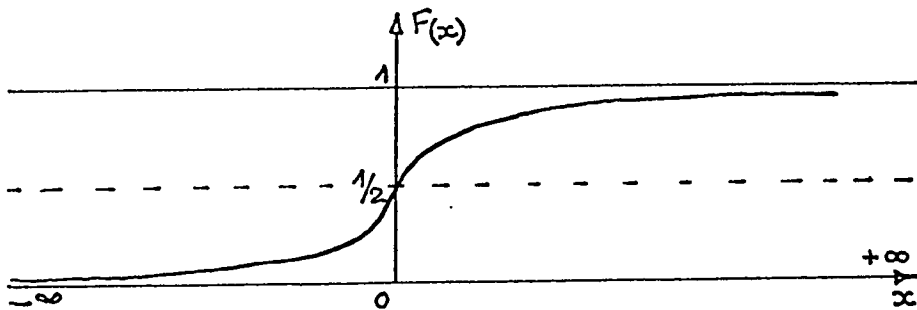
Graphe de $F(x)$.



d) Loi normale $\mathcal{N}(0,1)$:

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt$$

Graphe de $F(x)$.



III) Nature des problèmes rencontrés lors de la génération de variables aléatoires:

Lorsque l'on cherche à simuler une variable aléatoire X , on a pour objectif de créer un outil qui engendre, on dira génère, des valeurs qui pourraient être prises par X . Autrement dit, on a pour référence une suite de n variables aléatoires, généralement considérées comme indépendantes, $X_1, X_2, \dots, X_i, \dots, X_n$, (n quelconque), ayant toutes même loi que X . La question posée est de trouver un "générateur" tel que la suite des n réalisations x_1, x_2, \dots, x_n puisse être considérée comme une réalisation de X_1, X_2, \dots, X_n .

Mais cela reste assez imprécis. Aussi semble t'il plus raisonnable de circonscrire le problème autour d'une réflexion portant sur les opérations de contrôle qui peuvent être envisagées suite à une simulation particulière.

Soit une suite S , $S: \{x_1, x_2, \dots, x_n\}$, ($n = 10000$ par exemple), générée par simulation et sensée représenter les valeurs prises par une variable aléatoire X de fonction de répartition $F(x)$.

Comment contrôler que la simulation est de bonne qualité?

a) en première attente, on peut espérer que la distribution des x_i , (l'histogramme), "reproduise" $F(x)$ avec une assez bonne fidélité. Soit,

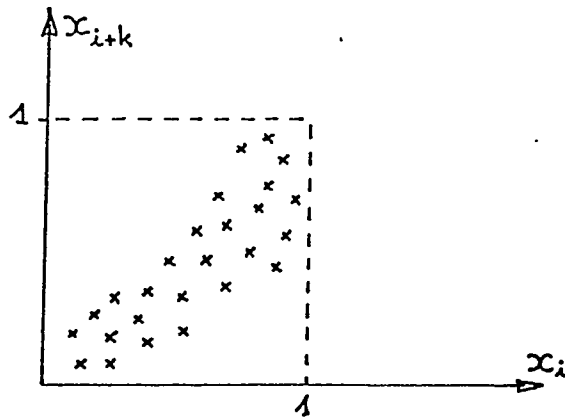
$$\forall x, \frac{\text{nombre des } x_i \text{ inférieurs à } x}{n} \simeq F(x).$$

b) Mais cette approche globale est insuffisante, car pour l'utilisateur il faut que la qualité de l'ajustement soit acceptable pour tout sous-ensemble de S . Ceci sous entend que la propriété a) puisse être vérifiée pour tout sous-ensemble S_i tel que $S_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}\}$, ($p \simeq 1000$ par exemple).

Exemple où cette propriété ne serait manifestement pas respectée:

X_i distribuée suivant une loi normale $N(0,1)$;
 S pourrait s'ajuster correctement, mais les 5000
 premières valeurs pourraient être toutes négatives
 et les 5000 suivantes toutes positives.

c) On a avancé l'hypothèse que les X_i devaient être indépendants;
 si par exemple en effectuant une représentation cartésienne des
 couples (x_i, x_{i+k}) , (k entier donné), en observant un nuage du
 type ci-dessous,



on pourrait conclure à la non indépendance des éléments de la
 suite.

IV) Lois uniformes: leur importance.

Sans entrer dans des considérations trop compliquées, on retiendra
 que la majorité des simulations peut se ramener à la génération de
 variables uniformément réparties entre 0 et 1.

Dans ce paragraphe, on suppose que l'on sait générer une variable
 Y de loi uniforme sur $[0,1]$.

a) Comment générer une variable X qui prend deux valeurs 0 et
 1 avec les probabilités p et $1-p$? X suit une loi de Bernouilli de
 paramètre p .

$$\begin{cases} P(X = 0) = p \\ P(X = 1) = 1-p \end{cases}$$

On génère n valeurs de Y

$$y_1, y_2, \dots, y_i, \dots, y_n \quad (0 \leq y_i \leq 1)$$

On compare chaque y_i à p , ($0 \leq p \leq 1$)

Si $y_i \leq p$, on pose $x_i = 0$

Si $y_i \geq p$, on pose $x_i = 1$

On obtient une suite x_1, x_2, \dots, x_n (simulation de X).

b) Comment simuler une variable discrète X ?

X prend les valeurs u_i avec les probabilités P_i .

$$u_i \in \{x_1, x_2, \dots, x_n\},$$

On génère n valeurs de Y

$$y_1, y_2, \dots, y_i, \dots, y_n, \text{ on compare chacun des } y_i \text{ aux } \sum_{j=1}^k P_j,$$

$$\text{si } \sum_{j=1}^k P_j \leq y_i \leq \sum_{j=1}^{k+1} P_j, \text{ on remplace } y_i \text{ par } u_k.$$

On obtient ainsi une suite $x_1, x_2, \dots, x_i, \dots, x_n$ (simulation de X).

c) Comment simuler une variable aléatoire X de fonction de répartition $F(x)$?

D'après les rappels effectués au paragraphe précédent, nous pouvons écrire $0 \leq P(X < x) = F(x) \leq 1$.

Considérons $Y = F(x)$, une nouvelle variable aléatoire, $0 \leq Y \leq 1$,

(Y est une transformation de X au même titre que X^2 ou $\frac{x^5}{x-1}$,,etc).

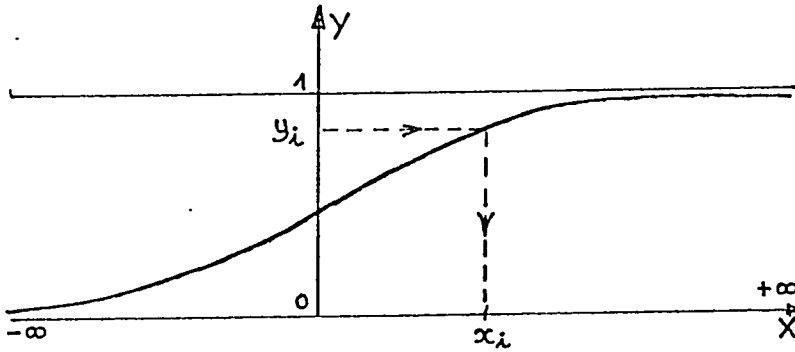
Si F est inversible, alors $X = F^{-1}(Y)$.

$P(a \leq Y \leq b) = P(a \leq F(X) \leq b) = P(F^{-1}(a) \leq X \leq F^{-1}(b))$
 $= F(F^{-1}(b)) - F(F^{-1}(a)) = b - a$, ce qui permet de conclure que Y suit une loi uniforme sur $[0,1]$.

On génère donc; n valeurs de Y

$y_1, y_2, \dots, y_i, \dots, y_n$, on calcule $x_i = F^{-1}(y_i)$,

on en déduit la suite des valeurs x_1, \dots, x_n (simulation de X).



Si $F(x)$, n'est pas inversible, on est amené à utiliser, entre autres, différentes méthodes numériques, basées par exemple pour certaines sur l'application du théorème central limite, pour d'autres sur des approximations polynomiales. (Illustration de celles-ci dans le paragraphe suivant lors de la description des méthodes de simulation d'une loi normale).

Enfin certaines variables, de par leur forme même, peuvent être simulées à partir de la génération d'autres variables; la variable χ^2 à n degrés de liberté, par exemple, sera construite comme la somme des carrés de n variables normales centrées réduites indépendantes, et l'on est alors ramené au problème de la génération de telles variables $N(0,1)$. Nota: c'est en particulier dans ce type d'applications que les considérations sur l'indépendance des réalisations, ici des variables $N(0,1)$, revêtent

toute leur importance.

V) Exemples: quelques méthodes de génération de lois particulières:

a) Cas de lois continues dont la fonction de répartition admet une inverse.

Si F est la fonction de répartition de la variable que l'on cherche à générer et que celle-ci est notée X . On peut écrire:

$$F(x) = P(X \leq x) = y, \text{ avec } y \in [0, 1]$$

Conformément à ce que nous avons vu au c) du paragraphe précédent, pour engendrer les x , il suffit de savoir générer des nombres y , réalisations d'une variable Y uniformément répartie entre 0 et 1 ($Y \sim U_{[0,1]}$), alors $x = F^{-1}(y)$.

Exemple: loi exponentielle de paramètre α ; $F(x) = 1 - e^{-\alpha x}$,

$$X = -\frac{1}{\alpha} \text{Log}(1-U) = -\frac{1}{\alpha} \text{Log}(U') \text{ avec } U' \sim U_{[0,1]}.$$

b) Cas particulier: la loi normale a une fonction de répartition continue, mais celle-ci ne peut être inversée, la transformation ne peut s'opérer que numériquement, soit par des méthodes approximatives, soit par des transformations de couples de variables uniformes.

1- Méthode de Box et Muller: c'est une méthode exacte basée sur une transformation d'un couple de variables uniformes.

Soit U_1 et U_2 deux variables indépendantes uniformément distribuées sur $0,1$ ($iid \sim U_{[0,1]}$), (nota: iid signifie indépendantes identiquement distribuées),

$$\text{alors } X_1 = (-2 \text{Log}(U_1))^{\frac{1}{2}} \cos 2\pi U_2$$

$$X_2 = (-2 \log(U_2))^{\frac{1}{2}} \sin 2\pi U_2$$

sont deux variables normales centrées réduites et indépendantes
soit $(X_1 \text{ et } X_2 \text{ iid } \sim \mathcal{N}(0,1))$

2) Variante: méthode de Box Muller et Marsaglia:

soit $U_1 \text{ et } U_2 \sim U_{[0,1]}$ iid, on pose alors:

$$V_1 = 2U_1 - 1$$

$$V_2 = 2U_2 - 1$$

$S = V_1^2 + V_2^2$, alors si $S < 1$ on obtient:

$$X_1 = V_1 \sqrt{\frac{-2\log(S)}{S}} \text{ et } X_2 = V_2 \sqrt{\frac{-2\log(S)}{S}} \text{ sont iid } \mathcal{N}(0,1)$$

3) Méthodes approximatives: applications du théorème central limite.

Soit U_1, U_2, \dots, U_n indépendantes de loi uniforme sur $[0,1]$,

$$\text{première méthode: } S_n = \sum_{i=1}^n U_i, \quad E(S_n) = \frac{n}{2}, \quad \text{var}(S_n) = \frac{n}{12}$$

$$X_n = \frac{S_n - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \xrightarrow[n \rightarrow \infty]{\text{loi}} \mathcal{N}(0,1)$$

Dans la pratique, on prend $n=12$; l'avantage de cette méthode réside dans sa rapidité, mais elle est très approximative, en particulier en ce qui concerne les queues de distribution: elle est donc à proscrire si l'on effectue sur les données simulées des tests qui portent justement sur ces queues de distribution.

variante:

on pose U_i pour i variant de 1 à 12 ,les U_i sont iid $\sim U_{[0,1]}$,
et toujours $S_n = \sum_{i=1}^n U_i$.

$$R = \frac{S_{12} - 6}{4} , R \text{ suit approximativement une distribution } \mathcal{N}\left(0, \left(\frac{1}{4}\right)^{\frac{1}{2}}\right)$$

alors l'expression $X = (((a_9 R^2 + a_7) R^2 + a_5) R^2 + a_3) R^2 + a_1) R$ suit
approximativement une distribution $\mathcal{N}(0,1)$

avec $a_1 = 3.949846138$
 $a_3 = 0.252408784$
 $a_5 = 0.076542912$
 $a_7 = 0.008355968$
 $a_9 = 0.029899776$

Cette méthode dite de Teichroew est une amélioration de la
précédente, l'amélioration est due au fait que l'expression
calculée est une approximation polynomiale de $F^{-1}(F_1(R))$
où $F_1(x)$ est la loi exacte de R .

c) Cas de variables discrètes:

Ce cas se présente par exemple dans le cas de problèmes de
dynamique de populations : (explosion, absorption, extinction ...).
Les lois les plus couramment utilisées sont les lois binomiales,
de Poisson, binomiale négative.

X entier évolue dans (i_1, i_2) et tel que $P_k = P(X = k)$ et $P(X = k) =$

$$Q_k - Q_{k-1} , \text{ soit, } Q_k = \sum_{i=i_1}^k p_k ;$$

si u est une réalisation de $U \sim U_{[0,1]}$, alors pour le plus petit k
tel que $u \leq Q_k$, on a: $X=k$.

Exemple: cas de la binomiale $Y \sim B(n,p)$; $P_k = C_n^k (1-p)^{n-k} p^k$
 $\forall k \in [0,n]$.

Y peut être considérée comme la somme de n variables de Bernouilli

X_i iid telles que $P(X_i=1)=p$; chacun des X_i sera engendré à partir d'un u_i de la façon suivante: $u_i \leq p \Rightarrow x_i=1$, sinon $x_i=0$.

On "voit bien" que la génération de variables aléatoires uniformément réparties entre 0 et 1 intervient partout, même dans ces cas simples; dès lors, indépendamment de toute considération sur la qualité de la modélisation (choix d'une loi plutôt qu'une autre par exemple), on s'intéressera aux différentes techniques de génération de variables uniformément réparties entre 0 et 1 ainsi qu'aux principaux tests permettant de juger de leur qualité.

VI) Différentes techniques de génération de variables $U_{[0,1]}$.

Générateurs de nombres aléatoires

Rappel: une variable aléatoire X uniformément répartie entre 0 et 1 est telle que $x \in [0,1]$ et telle que $P(X \leq x) = x$.

Une première méthode consiste à utiliser des réalisations de phénomènes physiques et aléatoires dont on a auparavant effectué l'analyse du point de vue fonction de répartition et un filtrage pour éliminer d'éventuels comportements aberrants. Cette méthode, certes séduisante, comporte des inconvénients dont les principaux sont:

- . Aucun contrôle sur les éventuelles modifications qui affectent le phénomène physique dans le temps.

- .. A moins de stocker les résultats, il n'y a aucun moyen de reproduire une série simulée; or cette propriété de répliquabilité des résultats est fort intéressante et importante: il peut en effet être utile de fixer certaines réalisations aléatoires alors que l'on s'autorise à en faire varier d'autres; par exemple, fixer l'attribution de traitements à des unités expérimentales alors que l'on fait varier la simulation des résidus dans le cas, pour fixer les idées, d'un dispositif en blocs complets. En effet l'observation de la variable $F_{\text{observé}}$ ne renseigne pas sur les conclusions à tirer quant à l'existence d'un effet traitement. Il

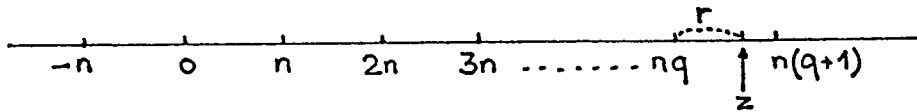
convient de répliquer des réalisations du dispositif un grand nombre de fois, d'ordonner les $F_{\text{observés}}$ obtenus et d'extraire le $F_{\text{observé}}$ correspondant au risque de première espèce α considéré.

(Si l'on dispose de 1000 simulations du même dispositif, on obtient 1000 $F_{\text{observés}}$ simulés, on les ordonne, on met en évidence le 950^{ème} et ses voisins dans le cas où le risque de première espèce α retenu vaut 5%).

Dans ces conditions, il a semblé préférable d'utiliser ce que l'on appelle des générateurs congruentiels, qui sont des moyens de construire des séries mathématiques dont le comportement s'apparente à celui du hasard, mais qui sont en fait totalement déterminées.

Rappel: soit z un entier relatif, n un entier positif, il existe toujours deux entiers q (quotient), et r (reste), tels que:

$$z = nq + r.$$



Par définition deux nombres z_1 et z_2 sont dits congrus modulo n lorsque divisés par n , ils fournissent le même reste r .

$$\text{On note : } z_1 \equiv z_2 \pmod{n}$$

(Il revient au même de dire que $z_1 - z_2$ est divisible par n).

Les générateurs congruentiels sont donc définis par des formules récursives de la forme:

$$Z_i \equiv a Z_{i-1} + c \text{ modulo } (m); \text{ ce qui se lit } Z_i \text{ est congru à } a Z_{i-1} + c, \text{ modulo } m.$$

si $c=0$, on dira que le générateur est multiplicatif,

si $c \neq 0$, on dira que le générateur est mixte.

Nota: la relation de modulo peut s'écrire algébriquement

$$Z_i = a Z_{i-1} + c - \left[\frac{a Z_{i-1} + c}{m} \right]_m$$

où $[x]$ désigne la partie entière de x .

Z_0 est l'amorce, il détermine la séquence qui sera générée.

Exemple: générateur multiplicatif $(a;m)$

$$Z_i \equiv a Z_{i-1} \pmod{m}$$

Valeur initiale Z_0

à partir de Z_0 , on calcule aZ_0 qui peut s'écrire $aZ_0 = mq_0 + r_0$

$r_0 = Z_1$, et $0 \leq Z_1 \leq m$, on en déduit donc $y_1 = \frac{Z_1}{m}$ qui est la première réalisation.

on a retenu la valeur de Z_1 ; à partir de Z_1 , on calcule de la même façon que précédemment aZ_1 qui peut se réécrire $aZ_1 = mq_1 + r_1$, on pose $r_1 = Z_2$, avec $0 \leq Z_2 \leq m$, on construit donc la deuxième réalisation y_2 de la série en calculant $y_2 = \frac{Z_2}{m}$, ainsi de suite de manière récursive.

Exemple numérique: $a=3$ $m=10$.

on prend comme amorce $Z_0 = 23$

$$aZ_0 = 69$$

$$69 = 10 \times 6 + 9, \text{ soit } Z_1 = 9, \text{ donc } y_1 = .9$$

$$aZ_1 = 27 = 10 \times 2 + 7, \text{ soit } Z_2 = 7, \text{ donc } y_2 = .7$$

$$aZ_2 = 21$$

$$21 = 10 \times 2 + 1 \text{ soit } Z_3 = 1, \text{ donc } y_3 = .1, \text{ etc}$$

Le problème se pose du choix de a, c, m . Ces nombres seront choisis sur des critères de période, de cycle du générateur, c'est-à-dire sur son aptitude à générer les chaînes les plus longues possibles

sans répétition des réalisations, ainsi que sur des critères de minimisation d'autocorrélations sérielles.

Plus généralement, les nombres générés doivent satisfaire des propriétés spatiales aussi bien que temporelles; ils doivent également contenir suffisamment de chiffres de manière à ce que la génération sur l'intervalle $[0,1]$ soit suffisamment dense.

a) Les seules valeurs de Z sont les entiers compris entre 0 et m , si l'on veut explorer le plus finement possible l'intervalle $[0,1]$, il faut que m soit grand.

b) Après m générations de nombres Z_i , on est certain de retrouver un reste Z_{m+1} qui a déjà été rencontré, donc de réitérer un cycle déjà exploré. Conclusion: si $m = 1000$, il est inutile de simuler 10000 valeurs. (m est identifié à la longueur du cycle, ceci est donc une raison supplémentaire pour choisir m grand).

c) Il faut que m et a soient premiers entre eux; dans le cas contraire $r_0 = 0$ et l'on ne génère que des y_i égaux à 0. (Pour les mêmes raisons m ne peut être quelconque).

d) On peut montrer que le coefficient de corrélation entre y_n et y_{n+1} est approximativement égal à $\frac{1}{a} \pm \frac{a}{m}$, il faut donc que a soit grand ($\frac{1}{a}$ faible) et petit par rapport à m ($\frac{a}{m}$ faible).

Exemples: générateur de Lehmer: $a = 23$, $m = 10^8 + 1$
générateur de Neave: $a = 131$, $m = 2^{35}$

Nota: à l'exception de "Urand" dont nous aurons l'occasion de reparler plus loin, la grande majorité des générateurs congruentiels actuellement utilisés sont de type multiplicatif, ce qui explique que nous n'ayons pas développé les contraintes qui président au choix de c .

VII) Tests de qualité des générateurs:

Différents outils ont été créés pour tester la validité des propriétés espérées des séries générées quant à leur adéquation à

des réalisations de nombres aléatoires uniformément répartis entre 0 et 1. Dans toute la suite de ce paragraphe nous désignerons par u_i les suites de nombres aléatoires simulés.

a) Tests d'uniformité:

- test du χ^2 :

On divise l'intervalle $[0,1]$ en k sous-intervalles égaux, on génère n nombres $\{u_i\}$ uniformes dans $[0,1]$; on note f_i la fréquence d'apparition dans le i ème sous-intervalle. Sous l'hypothèse d'uniformité, la fréquence théorique dans chaque sous-intervalle égale $f = \frac{n}{k}$.

Sous l'hypothèse d'uniformité, on a:

$$Q_1 = \frac{k}{n} \sum_{i=1}^k \left(f_i - \frac{n}{k} \right)^2 \xrightarrow[n \rightarrow \infty]{\text{loi}} \chi^2(k-1)$$

Nota: si l'on utilise ce test sur un calculateur, il est préférable de choisir les paramètres k et n de la forme: $k = 2^d$ et $n > 5k$.

- Test de Kolmogorov-Smirnov:

A partir d'un échantillon de réalisations u_1, u_2, \dots, u_n d'un générateur de nombres aléatoires, on construit la répartition empirique:

$$F_n(u) = \begin{cases} 0 & \text{si } u \leq u_{(1)} \\ \frac{j}{n} & \text{si } u_{(j)} < u \leq u_{(j+1)} \\ 1 & \text{si } u > u_{(n)} \end{cases}$$

Si $F(u)$ est la fonction de répartition théorique:

$D = \max_u |F(u) - F_n(u)|$ est distribuée indépendamment de F .

Ici $F(u)=u$; on rejette l'hypothèse d'uniformité au niveau α si $D > D_{n,\alpha}$ qui est une valeur tabulée: (voir annexe 4).

b) Tests sériels: tests d'indépendance et d'uniformité:

Le carré $[0,1] \times [0,1]$ est divisé en $k \times k$ classes égales. Soit $f_{i,j}$ le nombre de couples de valeurs successives u_{r-1}, u_r telles que u_{r-1} et u_r appartiennent respectivement au i ème et au j ème intervalles de $[0,1]$. Sous l'hypothèse d'indépendance et d'uniformité, la fréquence théorique de chaque classe i,j est $f = \frac{n}{k^2}$ où n est le nombre de couples (correspond à $n+1$ nombres).

$$\text{On pose } Q_2 = \frac{k^2}{n} \sum_{i=1}^k \sum_{j=1}^k \left(f_{i,j} - \frac{n}{k^2} \right)^2$$

$$\text{et } Q_1 = \frac{k}{n} \sum_{i=1}^k \left(f_i - \frac{n}{k} \right)^2 \quad (\text{cf: test du } \chi^2)$$

$$\text{On montre que: } Q_2 - Q_1 \underset{n \rightarrow \infty}{\rightarrow} \chi^2 (k-1)(k-2).$$

c) Tests d'indépendance:

- Test des "runs":

Ce test a pour but de vérifier que l'on n'est pas en présence de suites croissantes ou décroissantes: A partir de la série générée u_j , j variant de 1 à n , construit une suite binaire s_j , j variant de 1 à $n-1$, de la façon suivante.

$$\text{Soit } s_j = \begin{cases} 0 & \text{si } u_j \leq u_{j+1} \\ 1 & \text{si } u_j > u_{j+1} \end{cases} \quad \text{pour } j = 1, \dots, n-1$$

Nous allons à l'aide d'un exemple définir un "run".

Supposons, n la taille de la série de nombres aléatoires générée fixée à 9. Supposons également que l'on obtienne alors la série

des s_j , ($j=1,\dots,8$) suivante: 0,0,0,1,0,0,1,1.

Nous nous trouvons alors en présence de:

- . un "run zéro" de longueur 3,
- . un "run un" de longueur 1,
- . un "run zéro" de longueur de longueur 2,
- . un "run un" de longueur 2.

En fait, seuls les "runs zéro", et leur longueur nous intéressent. Ceci car cette information permet de déduire l'information analogue concernant les "runs un". Ceci nous amènera à désigner les "runs zéro" plus simplement par "run", sans qu'il n'y ait d'ambiguïté.

Soit V_k le nombre de "runs" formés de k zéros consécutifs. Sous l'hypothèse d'indépendance, le nombre moyen de tels "runs" est

$$\frac{2[(k^2+3k+1)n-(k^3+3k^2-k-4)]}{(k+3)!}$$
 ceci permet d'effectuer un test χ^2 de comparaison des fréquences.

- Test de part et d'autre de la moyenne:

Le principe est le même que dans la cas test des "runs": on construit, cette fois encore, une suite binaire, le but étant de vérifier que l'on a pas de suites ayant tendance à se trouver de part et d'autre de la moyenne.

$$s_j = \begin{cases} 0 & \text{si } u_j < .5 \\ 1 & \text{si } u_j \geq .5 \end{cases} \quad \text{pour } j = 1, \dots, n.$$

Le nombre moyen théorique de "runs" de longueur k sous l'hypothèse d'indépendance est $\frac{n-k+3}{2^k+1}$; on effectue encore un test de χ^2 .

- Test de corrélation: (autocorrélations succesives).

Sous l'hypothèse d'indépendance et d'uniformité,

$$E(U_j) = \frac{1}{2}$$

$$R_{j-k} = E\left[\left(U_j - \frac{1}{2}\right)\left(U_k - \frac{1}{2}\right)\right] = \begin{cases} \frac{1}{12} & \text{si } j=k \\ 0 & \text{si } j \neq k \end{cases},$$

Soit $\hat{R}_k = \frac{1}{n-k} \sum_{j=1}^{n-k} \left[\left(U_j - \frac{1}{2}\right)\left(U_{j+k} - \frac{1}{2}\right)\right]$ pour $k > 0$, on a:

$$E(\hat{R}_k) = 0 \text{ et } \text{var}(\hat{R}_k) = \frac{1}{144(n-k)}$$

$\sqrt{n-k} \hat{R}_k \xrightarrow[n \rightarrow \infty]{\text{loi}} N\left(0, \frac{1}{144}\right)$, on teste donc l'hypothèse $R_k = 0$.

VIII) Que faire lorsque ces propriétés ne sont pas respectées ?

Il est clair que l'idéal serait de disposer d'un générateur congruentiel parfait, malheureusement cette perle rare n'a pas encore été trouvée. Divers travaux sont menés dans ce sens de par le monde (L'INRIA, entre autres en France, travaille dans cette direction). De plus pour un même algorithme, ce qui peut faire la différence, c'est la façon de programmer; en effet, la gestion des erreurs d'arrondis peut avoir une incidence très importante sur la qualité des résultats: (on consultera utilement l'ouvrage de Morgan page 71 exercice 3.14 qui illustre parfaitement ce fait: voir annexe 2). Citant toujours Morgan, on retiendra què: "ces problèmes sont en général résolus en utilisant des programmes écrits en langage machine; dans ce cas l'opération de modulo est réalisée automatiquement sans division si le module $m = 2^r$ est la taille de la longueur de mot acceptée par l'ordinateur". Nota: les générateurs fournis avec les bibliothèques d'applications ou les logiciels satisfont en général cette condition de programmation optimale. Nous pouvons toutefois citer l'exemple du générateur prévu avec BMDP, qui ne fonctionne pas sous le système DATA GENERAL: il s'agit sans doute dans ce cas d'un programme écrit sous un autre système et qui ne s'accommode pas de la gestion des

overflows prévue par DATA GENERAL.

En règle générale, pour une application donnée, il se peut qu'une propriété particulière soit primordiale, l'utilisateur peut alors être amené (ou tenté) de modifier le générateur utilisé de manière à l'améliorer pour la propriété en question. Il ne faut toutefois pas se faire trop d'illusions, le gain sur cette propriété se fait souvent au détriment du comportement du générateur quant aux autres propriétés.

Exemple: Supposons qu'une application réclame un comportement correct par rapport à l'autocorrélation sérielle d'ordre 1, autrement dit que les réalisations successives des nombres aléatoires ne soient pas trop corrélées, et que le générateur (u_n) utilisé ne soit que médiocre par rapport à cette propriété. L'utilisateur peut espérer améliorer le comportement du générateur en ne retenant qu'une réalisation sur deux; mais le premier inconvénient apparent sera que la période du générateur ainsi créé diminuera, de plus les calculs seront plus longs, et pire on maîtrisera mal son comportement quant aux autres propriétés, à moins de refaire les tests ad hoc.

Morgan, toujours lui, cite une autre méthode d'amélioration (peut-être plus générale, et séduisante en apparence): elle consiste, pour générer une suite v_n de nombres aléatoires à partir d'un générateur noté u_n , à choisir le nombre v_n au hasard dans une séquence de taille 20 réalisée à partir du générateur original u_n ; le nombre v_n tiré, et donc manquant dans la séquence, est alors remplacé par u_{n+1} ; on choisit v_{n+1} , et ainsi de suite. Le gain obtenu en utilisant comme générateur initial, le générateur congruentiel mixte: $u_{i+1} = 781 u_i + 387$ modulo (1000) est illustré graphiquement (Morgan pages 62 et 63: voir annexe 2). Aucune allusion au comportement quant aux autres propriétés n'est faite.

De toutes façons, il serait souhaitable, dans le cas d'une modification d'un générateur en vue d'une amélioration, de lui faire subir les principaux tests cités précédemment afin de s'assurer qu'il reste acceptable pour ses autres propriétés.

IX) Conclusion:

A la lumière des paragraphes précédents, nous espérons avoir mis en évidence que la simulation est un outil puissant et séduisant, mais délicat à utiliser, son efficacité étant liée au comportement du générateur de nombres aléatoires uniformément répartis entre 0 et 1 utilisé. Ceux-ci sont nombreux et de qualité fort inégale. . On retiendra, citant Jean Pierre Vila, et concernant les générateurs congruentiels: "Après quelques essais avec les générateurs mixtes, il semble que l'on revienne actuellement aux générateurs purement multiplicatifs, plus rapides de mise en oeuvre et aux propriétés statistiques mieux contrôlées. Toutefois, un des meilleurs générateurs actuels, est un générateur mixte, URAND de G E Forsythe (voir annexe 3); il obéit aux recommandations de Knuth". Il est considéré comme adaptatif (propre à être implanté sur tous les calculateurs), mais si nous avons pu l'installer avec succès sur les ordinateurs DEC PDP 11-23 et BULL Mini 6, il n'en a pas été de même sur le matériel DATA GENERAL qui équipe le CIRAD, pas plus que sur les IBM-PC ou compatibles.

Avant toute chose, nous le répétons, il convient de tester le générateur que l'on est susceptible d'utiliser. Par ailleurs, dans toute publication utilisant une quelconque technique de simulation devrait figurer la description du ou des générateurs utilisés; c'est malheureusement rarement le cas. Citons une dernière fois Morgan: "Kennedy & Gentle (1980, p 165) rapportent qu'environ 30% des articles de l'association américaine de statistique (JASA) de 1978 ont employé la simulation. Dans un tel contexte, il est extrêmement important, dans le cas d'articles de recherche de spécifier l'algorithme utilisé et les tests employés pour les confronter au hasard. Mieux encore, des résultats produits grâce à des techniques de simulation devraient être vérifiés en utilisant un autre générateur".

Annexe 1:

Notre expérience:

C'est à travers l'étude de la méthode des plus proches voisins initiée par Papadakis que notre intérêt pour la simulation s'est révélé.

Grosso modo, nous rappellerons que la méthode de Papadakis peut être considérée comme une alternative à la mise en place de dispositifs en blocs, en utilisant pour chaque observation une covariable construite sur les observations voisines ou sur leurs résidus. L'idée en elle même est séduisante dans la mesure où les raisons qui président à la création de blocs (présence d'un gradient de fertilité par exemple), et par là même la mise en place de ceux-ci, ne sont pas toujours triviales. Cependant, la nature de la covariable construite conduit à quelques problèmes, en particulier sur la statistique de Fisher qui permet de tester la présence d'un effet traitement: "Bartlett en 1938, montra que la corrélation entre les rendements de plants voisins était si élevée, que la covariable pouvait expliquer une trop grande partie de la variation. Pour cette raison, il recommanda d'attribuer deux degrés de liberté au coefficient de régression (au lieu d'un), et ceci est resté dans la pratique." (S.C PEARCE: The agricultural field experiment: A statistical examination of theory and practice - 1983).

Cette décision d'accorder deux degrés de liberté au coefficient de régression, même si elle est en partie justifiée, semble être quelque peu empirique, et reste en tout cas sujette à caution. C'est elle, ainsi que notre curiosité à essayer de voir si des circonstances particulières pouvaient entraîner une supériorité de la méthode de Papadakis, par rapport à un dispositif en blocs, qui nous ont incités à employer la simulation en tant qu'outil d'investigation.

Au départ, le plus simple consiste naturellement à comparer la méthode de Papadakis à l'analyse de variance classique dans le cas d'un dispositif complètement randomisé en supposant qu'il n'y a

aucun effet traitement, (ce qui en pratique revient à ne simuler que des résidus). La statistique de Fisher obtenue après simulation et issue de l'analyse de variance classique prenait des valeurs que nous qualifierions pudiquement de suspectes. Force nous a été alors de conclure que nos générations étaient en cause.

Nous avons alors décidé de simuler directement des variables de Fisher en tant que rapport de sommes pondérées de carrés de lois normales centrées réduites. Les seuls éléments qui entrent alors en ligne de compte sont le générateur de nombres aléatoires uniformément répartis entre 0 et 1, et la méthode d'approximation de l'inverse de la fonction de répartition de la loi normale. Dans ce cas précis, et compte tenu des investigations limitées auxquelles nous avons pu nous livrer, il semble que le meilleur compromis soit fourni par le générateur GGUBFS (de la bibliothèque IMSL), en tant que générateur de nombres aléatoires couplé à la méthode de Box Muller et Marsaglia comme moyen de passer de la loi uniforme à la loi normale.

Pour ce qui est de la méthode de Papadakis, nous ne sommes pas aujourd'hui en mesure de fournir des résultats fiables. Il n'est d'ailleurs pas certain que nous ayons la maîtrise nécessaire dans le domaine de la simulation pour nous en servir comme outil à ce sujet.

L'enseignement majeur que nous avons tiré de cette expérience est la nécessité d'apporter un regard critique sur les méthodes de simulation. Ceci explique pourquoi, avant d'aller plus loin dans ce domaine, nous cherchions d'abord à faire un point plus complet sur la connaissance des générateurs que nous avons à disposition.

Annexe 2:

Dans cette annexe, on trouvera les exemples tirés de l'ouvrage de Morgan, qui illustrent pour le premier l'importance d'une gestion correcte des arrondis et pour le deuxième une amélioration possible d'un générateur médiocre quant à ses propriétés en matière d'autocorrélations sérielles.

Equation régissant le générateur utilisé pour illustrer les deux exemples précités.

$$x_{n+1} = 781 x_n + 387 \pmod{1000}$$

Deux types de programmation de ce générateur illustrant le rôle important de la gestion correcte des arrondis.

The following BASIC program simulates the mixed congruential generator of Equation (3.4), with $a = 781$, $b = 387$, $m = 1000$. Run this

```

10 REM MIXED CONGRUENTIAL GENERATOR
20 INPUT U0
30 LET A = 781
40 LET B = 387
50 FOR I = 1 TO 1000
60 LET U1 = (A*U0+B)/1000
70 LET U1 = (U1-INT(U1))*1000
80 LET U0 = U1
90 PRINT U1
100 NEXT I
110 END

```

program with and without the following change (from Cooke, Craven and Clarke, 1982, p. 70):

```
75 U1 = INT(U1 + 0.5)
```

Explication du mécanisme calculatoire sous-jacent.

If $A \times U_0 + B = (k \times 1000) + r$

where $0 \leq r < 1000$, $U_1 = k + r/1000 + \varepsilon$

where $\varepsilon \leq 0.001$ is the round-off error, and $\text{INT}(U_1) = r/1000 + \varepsilon$,

$$Y = (U_1 - \text{INT}(U_1)) \times 1000 = r + 1000\varepsilon$$

We require r , therefore set $U_1 = \text{INT}(r + 1000\varepsilon + \theta)$

where θ is such that $\theta + 1000\varepsilon > 0$, but $\theta + 1000\varepsilon < 1$.

$\theta = 0.5$ will do.

An example is:

```
1 168 595 82 429 435.964 874.781 ...
```

With the additional line we get:

```
1 168 595 82 429 436 903 ...
```

Graphe de u_{n+1} fonction de u_n ; une structure se détache apparemment.

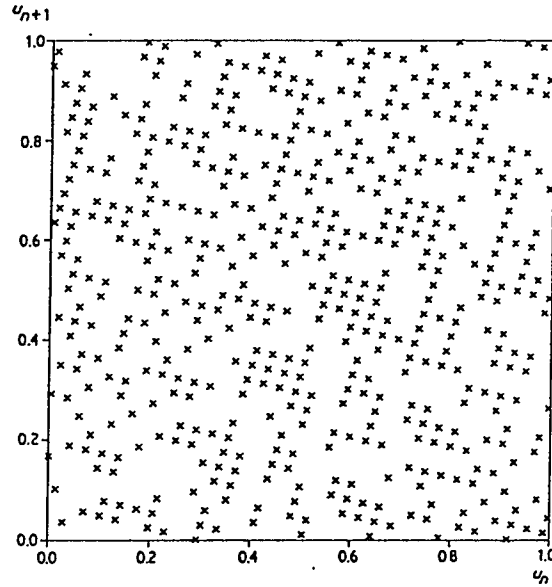


Figure 3.2 A plot of u_{n+1} vs. u_n for half the cycle of the mixed congruential generator of Equation (3.5).

Graphe de v_{n+1} fonction de v_n , où v_n est le générateur u_n modifié; la modification consistant à choisir le nombre suivant au hasard dans un ensemble de stockage de 20 nombre du générateur original.

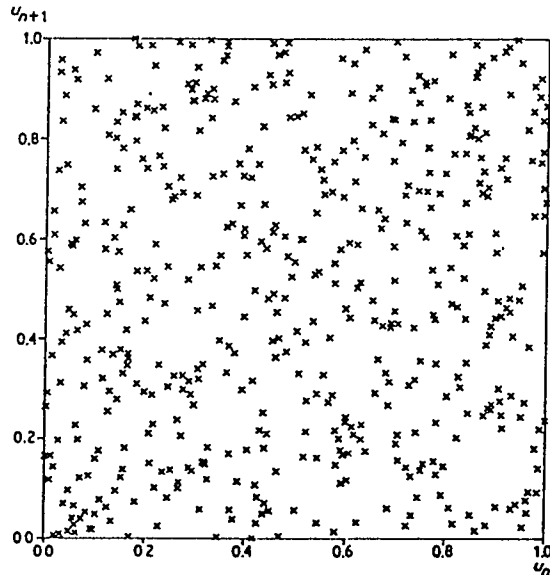


Figure 3.3 The plot resulting from modifying the same sequence that gave rise to Fig. 3.2. The modification entailed choosing the next number 'at random' from a store of length $g = 20$ of numbers from the original sequence, as explained in the text. In this example the random selection was made using Equation (3.1) and a seed of 0.5.

Annexe 3:

Listings des programmes fortran URAND.F77 et GGUBFS.F77.

Nota: URAND n'a pas pu être implanté sur le matériel DATA GENERAL ni sur les PC compatibles; Il s'agit sans doute dans les deux cas d'une mauvaise gestion des overflows, (en français "overflow" signifie dépassement de capacité), due soit aux systèmes d'exploitation soit aux compilateurs. GGUBFS lui, a pu être installé dans les deux cas.

```

real function urand(iy)
save
integer iy

c
c urand is uniform random generator based on theory and
c suggestions given in D.E.KNUTH. vol 2. The integer iy
c should be initialized to an arbitrary integer prior to the
c first call to urand. The calling program should not alter
c the value of iy between subsequent calls to urand. Values of
c urand will be returned in the interval (0,1)
c
c extrait de Forsythe, Malcolm, Moler (1977): computer methods
c for mathematical computation. prentice Hall Englewood Cliffs,
c New Jersey.
c
integer ia,ic,itwo,m2,m,mic
double precision halfm
double precision datan,dsqrt
data m2/0/,itwo/2/
if(m2.ne.0) goto 20

c
c if first entry, compute machine integer word length
c
m=1
10 m2=m
m=itwo*m2
if(m.gt.m2) goto 10
halfm=m2

c
c compute multiplier and increment for linear congruential method
c
ia=8*idint(halfm*datan(1.d0)/8.d0)+5
ic=2*idint(halfm*(0.5d0-dsqrt(3.d0)/6.d0))+1
mic=(m2-ic)+m2

c
c s is the scale for converting to floating point
c
s=0.5/halfm

c
c compute next random number
c
20 iy=iy*ia

c
c the following statement is for computers which do not allow
c integer overflow on addition
c
if(iy.gt.mic) iy=(iy-m2)-m2

c
iy=iy+ic

c
c the following statement is for computers where the word length
c for addition is greater than for multiplication
c
if(iy/2.gt.m2) iy=(iy+m2)+m2
urand=float(iy)*s
return
end

```

```
real function ggubfs(dseed)
double precision dseed
double precision d2p31m,d2p31
data          d2p31m/214783647.d0/
data          d2p31 /214783648.d0/
dseed=dmod(16807.d0*dseed,d2p31m)
ggubfs=dseed/d2p31
return
end
```

Annexe 4:

Table de la statistique D du test de Kolmogorov-Smirnov; Mode d'emploi et commentaires.

(Extrait de "Handbook of statistics 4: nonparametric methods").

Krihnaiah & Sen - North Holland. (1984).

5. Tables for the Kolmogorov-Smirnov type statistics

For n observations drawn from a continuous distribution F , defined on the real line \mathbb{R} , let $F_n(x) = n^{-1}$ (number of observations with values $\leq x$), $x \in \mathbb{R}$, be the sample (empirical) distribution function. Then, the one-sample Kolmogorov-Smirnov statistics are defined as

$$D_n^+ = \sup\{F_n(x) - F(x) : x \in \mathbb{R}\} \quad (5.1)$$

and

$$D_n = \sup\{|F_n(x) - F(x)| : x \in \mathbb{R}\}. \quad (5.2)$$

The probability distribution of D_n for small values of n , has been tabulated by Birnbaum (1952). For D_n^+ , some tabulations of the critical values are due to Millar (1956). We adopt these tables to provide the critical values for D_n^+ and D_n for some specific values of n and for significance levels close to 0.05 and 0.01. Note that for significance levels not so large (i.e., less than 0.1), we have

$$\alpha_n = P\{D_n \geq \lambda\} \approx 2P\{D_n^+ \geq \lambda\}. \quad (5.3)$$

Hence, we only provide the critical values for D_n for α_n close to 0.1, 0.05, 0.02 and 0.01. These are given in Table 5.1. In passing, we may note that for large n ,

Table 5.1
Table for the critical values of the one-sample Kolmogorov-Smirnov statistic D_n

n	λ and α_n		
5		4/5 (0.031)	5/5 (0.0006)
6	4/6 (0.066)		5/6 (0.004)
7		5/7 (0.011)	6/7 (0.0004)
8		5/8 (0.023)	6/8 (0.0015)
9		5/9 (0.039)	6/9 (0.0039)
10	5/10(0.059)		6/10(0.0078)
11	5/11(0.083)	6/11(0.014)	7/11(0.0013)
12	5/12(0.109)	6/12(0.021)	7/12(0.0027)
13		6/13(0.031)	7/13(0.0047)
14		6/14(0.042)	7/14(0.0075)
15	6/15(0.055)	7/15(0.011)	8/15(0.0016)
16	6/16(0.069)	7/16(0.016)	8/16(0.0026)
17	6/17(0.085)	7/17(0.021)	8/17(0.0040)
18	6/18(0.102)	7/18(0.028)	8/18(0.0058)
19	6/19(0.120)	7/19(0.035)	8/19(0.0080)
20		7/20(0.043)	8/20(0.0107)
21	7/21(0.052)	8/21(0.014)	9/21(0.0030)
22	7/22(0.062)	8/22(0.018)	9/22(0.0041)
23	7/23(0.072)	8/23(0.022)	9/23(0.0054)
24	7/24(0.083)	8/24(0.027)	9/24(0.0070)
25	7/25(0.094)	8/25(0.032)	9/25(0.0089)
26	7/26(0.106)	8/26(0.037)	10/26(0.0027)
27	7/27(0.119)	8/27(0.043)	10/27(0.0035)
28		8/28(0.050)	10/28(0.0045)
29		8/29(0.057)	10/29(0.0056)
30		8/30(0.064)	10/30(0.0068)

for every $t \geq 0$,

$$P\{n^{1/2}D_n^+ \geq t\} \rightarrow e^{-2t^2}, \quad (5.4)$$

$$P\{n^{1/2}D_n \geq t\} \rightarrow 2(e^{-2t^2} - e^{-8t^2} + e^{-18t^2} - \dots) \quad (5.5)$$

and, actually, the right hand sides of (5.4) and (5.5) provide upper bounds for any finite sample size. These approximations are quite good for $n \geq 31$. Hence, we provide the entries only for $n \leq 30$.

For two samples of equal sizes n , if F_n and G_n stand for the empirical distributions, then one may define the one and two-sided Kolmogorov-Smirnov statistics as in (5.1) and (5.2) with F being replaced by G_n . In this case, (5.4) and (5.5) hold when we replace $n^{1/2}D_n^+$ and $n^{1/2}D_n$ by $(n/2)^{1/2}D_n^+$ and $(n/2)^{1/2}D_n$, respectively. For this two-sample case, Birnbaum and Hall (1960) have tabulated the distributions for specific values of n , and we adopt their tables to provide the critical values for specific level of significances. For the case of more than two samples, we refer to Section 7 for some tabulation of the asymptotic critical values, mostly due to Kiefer (1959). In Tables 5.1 and 5.2, the entries refer to λ and α_n for (5.3) in the one and two-sample cases.

Table 5.2
Table for the critical values of two-sample D_n^+ and D_n for some specific n

n	D_n^+		D_n	
5	4/5 (0.040)	5/5 (0.004)	4/5 (0.079)	5/5 (0.008)
6	4/6 (0.061)	5/6 (0.013)	5/6 (0.026)	6/6 (0.002)
7	5/7 (0.027)	6/7 (0.004)	5/7 (0.053)	6/7 (0.008)
8	5/8 (0.044)	6/8 (0.009)	5/8 (0.087)	6/8 (0.019)
9	5/9 (0.063)	6/9 (0.017)	6/9 (0.034)	7/9 (0.006)
10	6/10(0.026)	7/10(0.006)	6/10(0.053)	7/10(0.012)
11	6/11(0.038)	7/11(0.010)	6/11(0.075)	7/11(0.020)
12	6/12(0.050)	7/12(0.016)	7/12(0.031)	8/12(0.008)
13	7/13(0.022)	8/13(0.006)	7/13(0.045)	8/13(0.013)
14	7/14(0.030)	8/14(0.009)	7/14(0.059)	8/14(0.019)
15	7/15(0.038)	8/15(0.013)	8/15(0.026)	9/15(0.008)
16	7/16(0.047)	8/16(0.017)	8/16(0.035)	9/16(0.011)
17	8/17(0.022)	9/17(0.008)	8/17(0.045)	9/17(0.016)
18	8/18(0.028)	9/18(0.010)	8/18(0.056)	9/18(0.021)
19	8/19(0.034)	9/19(0.013)	9/19(0.027)	10/19(0.009)
20	8/20(0.041)	10/20(0.006)	9/20(0.034)	10/20(0.012)
21	8/21(0.047)	10/21(0.008)	9/21(0.041)	11/21(0.006)
22	8/22(0.055)	10/22(0.010)	9/22(0.049)	11/22(0.007)
23	9/23(0.029)	11/23(0.005)	9/23(0.058)	11/23(0.009)
24	9/24(0.034)	11/24(0.006)	10/24(0.030)	11/24(0.012)
25	9/25(0.039)	11/25(0.007)	10/25(0.036)	12/25(0.006)
26	9/26(0.045)	11/26(0.009)	10/26(0.042)	12/26(0.007)
27	9/27(0.050)	11/27(0.011)	10/27(0.049)	12/27(0.009)
28	9/28(0.055)	12/28(0.005)	10/28(0.056)	12/28(0.011)
29	10/29(0.032)	12/29(0.007)	11/29(0.030)	13/29(0.005)
30	10/30(0.035)	12/30(0.008)	11/30(0.035)	13/30(0.007)

Bibliographie sommaire:

- (1) JOHNSON N,L & S KOTZ: Distributions in statistics: discrete distributions. Houghton Mifflin, Boston - (1969).
- (2) JOHNSON N,L & S KOTZ: Continuous univariate distributions 1. Houghton Mifflin, Boston - (1970 a).
- (3) JOHNSON N,L & S KOTZ: Continuous univariate distributions 2. Houghton Mifflin, Boston -(1970 b).
- (4) JOHNSON N,L & S KOTZ: Distributions in statistics: continuous multivariate distributions. Wiley, New York.
- (5) KNUTH D,E : The Art of Computer Programming Seminumerical Algorithms. Volume 2 . Addison Wesley, Reading Mass - (1969).
- (7) KRISHNAIAH P,R & P K SEN : Handbook of statistics 4 nonparametric methods. North Holland - (1984).
- (8) MORGAN B,J,T: Elements of simulation. CHAPMAN AND HALL. London, New York - (1986).
- (9) VILA J P: Simulation. Publication conjointe INRA, université PARIS XI - (1984).