

# REVUE DE STATISTIQUE APPLIQUÉE

J.-P. ASSELIN DE BEAUVILLE

## Les sous-programmes usuels de simulation statistique

*Revue de statistique appliquée*, tome 22, n° 4 (1974), p. 57-87

[http://www.numdam.org/item?id=RSA\\_1974\\_\\_22\\_4\\_57\\_0](http://www.numdam.org/item?id=RSA_1974__22_4_57_0)

© Société française de statistique, 1974, tous droits réservés.

L'accès aux archives de la revue « *Revue de statistique appliquée* » (<http://www.sfds.asso.fr/publicat/rsa.htm>) implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

# LES SOUS-PROGRAMMES USUELS DE SIMULATION STATISTIQUE

J.-P. ASSELIN de BEAUVILLE  
Université de Tours

*Cet article présente brièvement les sous-programmes Fortran de création d'échantillons artificiels des lois statistiques usuelles. Pour chaque loi il n'a été retenu, parmi les nombreux algorithmes possibles, que la méthode la plus simple à mettre en oeuvre. Le lecteur désireux d'approfondir le problème pourra se référer aux ouvrages cités dans la bibliographie.*

## TABLE DES MATIERES

- I – La loi continue uniforme (Nombres pseudo-aléatoires)
- II – La loi Normale
- III – La loi Log-normale
- IV – La distribution du khi-deux
- V – La loi exponentielle
- VI – La distribution logistique
- VII – La loi Gamma
- VIII – La loi Béta
- IX – La loi de F.
- X – La loi de Student
- XI – La loi de Cauchy
- XII – La loi de Poisson
- XIII – La loi Binomiale
- XIV – Loi expérimentale quelconque
- XV – Loi Théorique quelconque
  - Bibliographie

-----  
(\* ) Article remis le 13/6/73. Révisé le 10/5/74

I – LA LOI CONTINUE UNIFORME (Nombres pseudo-aléatoires) :

I – 1. Caractéristiques de la loi continue uniforme :

On appelle variable continue uniforme, la variable aléatoire X distribuée suivant la loi de densité de probabilité suivante :

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

on note généralement :  $X \sim U(0, 1)$ .

Les moments par rapport à la moyenne (ou moments centrés) d'ordre pair ( $2r$ ) sont égaux à :

$$\mu_{2r} = \frac{1}{(2r+1) 4^r}$$

Ceux d'ordre impair sont nuls à cause de la symétrie.

Les moments non centrés sont donnés par :

$$\mu'_1 = \int_0^1 x^r dx = \frac{1}{r+1}$$

On déduit de ces relations :

$$E(X) = \mu'_1 = \frac{1}{2}$$

$$V(X) = \mu_2 = \frac{1}{12}$$

$$\mu_3 = 0$$

$$\mu_4 = \frac{1}{80}$$

Par extension on appelle : "loi Rectangulaire"  $U(a, b)$  la loi de densité de probabilité :

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } a < x < b \\ 0 & \text{si } x < a \text{ ou } x > b \end{cases}$$

Pour cette loi on a :

$$E(X) = \frac{a+b}{2}$$

$$\mu_{2r} = \frac{(b-a)^{2r}}{(2r+1) \cdot 4^r}$$

### I-2. Processus générateur :

On appelle "Nombres aléatoires" (ou nombres au hasard) une suite de nombres distribués suivant  $U(0,1)$ .

Lorsque cette suite est obtenue à partir d'une relation mathématique on dit : "nombres pseudo-aléatoires" (ou nombres pseudo-au Hasard).

Lorsqu'on dispose d'un ordinateur, la procédure la plus utilisée aujourd'hui pour engendrer une suite de nombres pseudo-aléatoires est la méthode dite congruentielle multiplicative. Lehmer [1] suggéra, en 1951, qu'une suite pseudo-aléatoire pouvait être obtenue par la relation de récurrence :

$$U_{i+1} \equiv \lambda \cdot U_i \pmod{m}$$

avec  $i = 0, 1, 2, \dots$

Cette notation signifie que  $U_{i+1}$  est le reste de la division de  $\lambda \cdot U_i$  par  $m$ .

où  $\left\{ \begin{array}{l} m = \text{nombre entier de grande taille (généralement une puissance de 2 pour} \\ \text{les machines binaires)} \\ \lambda = \text{nombre entier compris entre 0 et } m-1 \end{array} \right.$

Les nombres  $\frac{U_{i+1}}{m}$ , compris entre 0 et 1, sont alors utilisés comme suite pseudo-aléatoire. Il faut remarquer qu'une telle suite sera toujours périodique et qu'elle se répétera après au plus  $m$  termes [3]. Le nombre  $m$  est généralement une puissance de la base de numération du calculateur utilisé. Pour une machine binaire on choisira donc  $m = 2^{n-1}$  où  $n$  représente la longueur d'un mot machine en bits (le premier bit étant généralement réservé au signe). On montre [voir (2) par exemple] que la période maximum de la suite est  $\frac{m}{4}$  si les conditions suivantes sont réalisées :

1)  $\lambda = \pm 3 \pmod{8}$  ou ce qui est équivalent :

$$\lambda = 8t \pm 3 \text{ où } t \text{ est un nombre entier positif quelconque.}$$

En général on prendra [3] :

$$\lambda = 5^{13} = 1220703125$$

2) La valeur initiale  $U_0$  de  $U_i$  est un nombre entier positif *impair* quelconque.

### I-3. Sous-programme FØRTRAN :

Sur un ordinateur il est possible de réaliser la congruence modulo  $2^{n-1}$  de la façon suivante [4] :

Les deux nombres entiers  $U_{i+1}$  et  $\lambda \cdot U_i$  sont congruents modulo  $2^{n-1}$  lorsque divisés par  $2^{n-1}$  ils fournissent le même reste, donc si :

$$\begin{cases} \lambda \cdot U_i = q \cdot 2^{n-1} + r \\ U_{i+1} = q' \cdot 2^{n-1} + r \end{cases}$$

Or  $\lambda \cdot U_i$  s'écrit en binaire :

$$a_{n+p} \cdot 2^{n+p} + a_{n+p-1} \cdot 2^{n+p-1} + \dots + a_n \cdot 2^n + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

où  $a_i = 0$  ou  $1$ .

Soit encore :

$$2^{n-1} \cdot (a_{n+p} \cdot 2^{p+1} + a_{n+p-1} \cdot 2^p + \dots + a_n \cdot 2 + a_{n-1} \cdot 2^0) + a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

On obtient alors en identifiant cette expression avec celle de  $\lambda \cdot U_i$  :

$$q = a_{n+p} \cdot 2^{p+1} + a_{n+p-1} \cdot 2^p + \dots + a_n \cdot 2 + a_{n-1} \cdot 2^0$$

$$r = a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

D'autre part, on peut choisir  $q' = 0$  donc :

$$U_{i+1} = r = a_{n-2} \cdot 2^{n-2} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

Sur une machine à mot de  $n$  bits ceux d'ordre  $(n+p)$ ,  $(n+p-1)$ ,  $\dots$ ,  $(n)$  sont automatiquement éliminés (troncature par la gauche).

Si  $a_{n-1} = 0$  le bit correspondant est éliminé naturellement.

Si  $a_{n-1} = 1$ , on élimine ce bit (puisque le premier bit égal à 1 signifierait que  $U_{i+1}$  est négatif) en inversant la parité du nombre  $U_{i+1}$  (En effet,  $a_{n-1} = 0$  pour un nombre positif). Pour cela, on peut ajouter à  $U_{i+1}$  le nombre positif le plus grand représentable dans un mot-machine plus 1, soit :  $2^{n-1}$  pour un calculateur à mot de  $n$  bits.

Considérons un exemple pour fixer les idées :

Supposons une machine pour laquelle  $n = 5$  (5 bits par mot). Le premier bit étant le bit de parité (0 si nombre positif et 1 si le nombre est négatif), on conçoit que l'entier *positif* le plus grand que puisse contenir cette machine sera en représentation binaire :

$$0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \text{ ou encore en binaire pur : } 01111 = +15 \text{ en décimal.}$$

On remarque bien que  $15 = 2^{n-1} - 1$ .

Supposons, d'autre part, que le produit  $\lambda \cdot U_0$  soit égal à :

$$a_9 \cdot 2^9 + a_8 \cdot 2^8 + a_7 \cdot 2^7 + a_6 \cdot 2^6 + a_5 \cdot 2^5 + a_4 \cdot 2^4 + a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

Le mot machine ne comportant que 5 bits, on trouvera en mémoire :  $a_4 a_3 a_2 a_1 a_0$ , tous les bits d'ordre supérieur à  $a_4$  ( $a_{n-1}$ ) étant supprimés par troncature.

1) Si  $a_4 = 0$ , le premier bit étant réservé au signe, on a bien un nombre positif et le nombre  $U_1$  congru à  $U_0$  modulo  $2^4$  sera simplement :

$$U_1 = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

2) Si  $a_4 = 1$ , le nombre est négatif et pour l'utiliser il faut, au préalable, inverser son signe. Pour cela on lui ajoute le nombre  $2^{n-1}$  soit, en décimal, le nombre  $2^4 = 16$  : soit en binaire :  $2^4 = 10000$

On obtient donc :

$$\begin{array}{r} 1 \ a_3 \ a_2 \ a_1 \ a_0 \\ + 1 \ 0 \ 0 \ 0 \ 0 \\ \hline 1 \ 0 \ a_3 \ a_2 \ a_1 \ a_0 \end{array}$$

En mémoire il restera :

$$0 \ a_3 \ a_2 \ a_1 \ a_0$$

Le nombre  $U_1$  sera donc, comme précédemment, égal à :

$$U_1 = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

Pour une machine dont le mot comporte 32 bits (telles que les machines I.B.M., série 360, par exemple) on pourra donc utiliser le sous-programme suivant, en remarquant que :

$$2^{31} = 2147483648$$

$$\text{et } \frac{1}{2^{31}-1} = 0,4656613 \cdot 10^{-9}$$

SUBROUTINE HASARD (IX,IY,U)

IY=IX\*1220703125

IF=(IY)5,6,6

5 IY=IY+2147483647+1

6 U=IY

U=U\*0.4656613E-9

RETURN

END

1er argument : IX, type entier, c'est la valeur de départ de l'itération. C'est un nombre entier positif *impair* comportant moins de 9 chiffres décimaux (pour I.B.M. 360).

2ème argument : IY, type entier, c'est un nombre entier au hasard compris entre 0 et  $2^{31}-1$ . Il peut être ensuite utilisé comme valeur de réinitialisation de IX.

3ème argument : U, type réel, c'est le résultat. U est une variable aléatoire continue uniforme.

utilisation : IX=nombre entier positif impair quelconque  
CALL HASARD (IX, IY, U)  
IX=IY

Ce schéma d'utilisation sera toujours identique pour tous les sous-programmes inclus dans cet article. En particulier, il conviendra de ne pas omettre l'instruction IX=IY qui suit l'appel de Hasard, car c'est elle qui permet la réinitialisation de IX pour le prochain appel du sous-programme.

Pour obtenir une variable aléatoire (AN), rectangulaire sur le segment [B,C] il suffit d'ajouter au sous-programme Hasard une sixième instructions telles que :

SUBROUTINE RECTAN (IX,B,C,AN,IY)

.  
. .  
. .  
. .  
. .  
. .

AN=U\*(C-B)+B

RETURN

END

Notons, pour terminer, qu'il est toujours recommandé de vérifier qu'une suite de nombres pseudo-aléatoires engendrés par un algorithme mathématique remplit bien la condition d'équiprobabilité. Le test du Khi-deux pourra être utilisé dans ce but. D'autres tests plus importants sont décrits dans [5].

## II – LA LOI NORMALE :

### II-I. Caractéristiques de la loi normale :

Une variable aléatoire X distribuée normalement a pour fonction de densité de probabilité :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp \left[ -\frac{(x-m)^2}{2\sigma^2} \right] \text{ avec } -\infty < x < +\infty$$

où

$$\begin{cases} m = \text{moyenne (centre de symétrie)} \\ \sigma = \text{écart-type} \end{cases}$$

On note généralement  $X \sim N(m, \sigma)$ . Dans le cas où  $m = 0$  et  $\sigma = 1$ , on dit que  $X$  suit la loi normale réduite ( $X \sim N(0, 1)$ ). Si la variable aléatoire  $Z$  est distribuée suivant la loi normale réduite, la variable aléatoire  $X = \sigma.Z + m$  sera distribuée suivant  $N(m, \sigma)$ .

Dans la suite de cette étude on utilisera pour décrire la loi suivie par une variable aléatoire les coefficients d'asymétrie ( $\gamma_1$ ) et d'aplatissement ( $\gamma_2$ ) décrits par Fisher :

$$\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}}$$

$$\gamma_2 = \frac{\mu_4}{\mu_2^2} - 3$$

$\gamma_1$  est nul pour une loi symétrique mais l'inverse n'est en général pas vrai.  $\gamma_1$  est positif pour une distribution unimodale plus étalée vers la droite et négatif dans le cas contraire.  $\gamma_2$  est nul pour une distribution normale (condition non suffisante).  $\gamma_2$  est positif si la loi décrite est moins aplatie que la loi normale (de même  $\sigma$ ) et négatif dans le cas contraire.

Ainsi pour la loi  $N(m, \sigma)$  on a :  $\gamma_1 = \gamma_2 = 0$ .

## II – 2 Processus générateur :

Nous décrivons deux méthodes, On trouvera d'autres algorithmes dans [3].

### a) Utilisation du Théorème Central limite :

Ce théorème stipule que si  $r_1, r_2, \dots, r_n$  sont des variables aléatoires indépendantes distribuées suivant la même loi avec  $E(r_i) = m$  et  $\text{Var}(r_i) = \sigma^2$ , on a alors :

$$\lim. \text{Prob} \left[ a < \frac{\sum_{i=1}^N (r_i) - N.m}{\sigma \sqrt{N}} < b \right] = \frac{1}{\sqrt{2\pi}} \cdot \int_a^b \exp \left( -\frac{z^2}{2} \right) dz$$

avec :

$$E \left( \sum_{i=1}^N r_i \right) = \sum_{i=1}^N E(r_i) = N.m$$

$$\text{Var} \left( \sum_{i=1}^N r_i \right) = \sum_{i=1}^N [\text{Var}(r_i)] = N.\sigma^2$$

$$z = \frac{\sum_{i=1}^N r_i - N.m}{\sigma \sqrt{N}}$$

$z$  suit la loi normale réduite lorsque  $N$  tend vers l'infini.



Si on choisit K variables  $r_i$  distribuées suivant  $U(0,1)$ , la variable aléatoire définie par  $\sum_{i=1}^K r_i$  aura pour moyenne et pour variance :

$$E\left[\sum_{i=1}^K r_i\right] = \frac{K}{2}$$

$$\text{Var}\left[\sum_{i=1}^K r_i\right] = \frac{K}{12}$$

et la variable  $z = \frac{\sum_{i=1}^K r_i - k/2}{\sqrt{K/12}}$  sera distribuée suivant  $N(0,1)$  si K est assez grand.

En fin de compte  $X = \sigma.Z + m$  suivra la loi  $N(m, \sigma)$ . On en déduit :

$$X = \sigma \cdot \left(\frac{12}{K}\right)^{1/2} \left[\sum_{i=1}^K r_i - \frac{K}{2}\right] + m$$

Ainsi pour obtenir une valeur de X (variable normale) il faudra utiliser K nombres pseudo-aléatoires. Plus la valeur de K est élevée, meilleure est la précision ; néanmoins afin de ne pas allonger le temps de calcul, on doit limiter cette valeur, pour des raisons évidentes. On pose généralement  $K = 12$ . Ce choix a pour résultat de tronquer la loi normale aux points d'abscisse  $\pm 6\sigma$  [6]. Pour plus de précision sur les queues de la loi il faudra utiliser une autre méthode.

#### b) Méthode de BOX et MULLER [7]

Pour obtenir une variable aléatoire distribuée exactement suivant  $N(m, \sigma)$ , on pourra procéder comme suit :

Si  $X_1$  et  $X_2$  sont deux variables aléatoires indépendantes distribuées suivant la loi normale réduite, la loi du couple  $(X_1, X_2)$  est :

$$f(x_1, x_2) = \frac{1}{2\pi} \cdot \exp\left[-\frac{x_1^2 + x_2^2}{2}\right]$$

posons

$$x_1 = r \cdot \cos \theta$$

$$x_2 = r \cdot \sin \theta$$

La loi du couple  $(r, \theta)$  des nouvelles variables

est :  $g(r, \theta) = f(x_1, x_2) \cdot |J|$ . où

$|J|$  est le Jacobien de la transformation.

Ici :

$$|J| = \begin{vmatrix} \frac{\delta x_1}{\delta r} & \frac{\delta x_1}{\delta \theta} \\ \frac{\delta x_2}{\delta r} & \frac{\delta x_2}{\delta \theta} \end{vmatrix} = r(\cos^2 \theta + \sin^2 \theta) = r$$

$$\text{d'où } g(r, \theta) = \frac{1}{2\pi} \cdot \exp\left[-\frac{r^2}{2}\right] \cdot r$$

soit, en considérant la probabilité élémentaire associée au domaine  $(dr, d\theta)$  :

$$\begin{aligned} g(r, \theta) \cdot dr \cdot d\theta &= \frac{1}{2\pi} \cdot \exp\left[-\frac{r^2}{2}\right] \cdot r \cdot dr \cdot d\theta \\ &= \exp\left[-\frac{r^2}{2}\right] \cdot d\left(\frac{r^2}{2}\right) \cdot \frac{d\theta}{2\pi} \end{aligned}$$

Cette dernière relation montre que  $r^2$  et  $\theta$  sont distribuées indépendamment l'une de l'autre et que :

a)  $\theta$  est distribuée rectangulairement sur  $[0, 2\pi]$ .

b)  $-\frac{r^2}{2}$  est distribuée exponentiellement.

Posons  $y = \frac{r^2}{2}$ , on peut écrire que :

$f(y) = \exp(-y)$  ou encore :

$$F(Y) = \int_0^y f(y) \, dy = 1 - \exp(-y) \text{ avec } y \geq 0.$$

Une variable de densité  $f(y)$  peut toujours être transformée en une variable aléatoire continue uniforme  $U_1$  par la transformation :  $F(Y) = U_1$ . En effet :

$$dF(y) = f(y) \cdot dy$$

or

$$du_1 = dF(y) = f(y) \cdot dy \quad \text{d'où : } \frac{dy}{du_1} = \frac{1}{f(y)}$$

Mais on peut écrire :

$$dF(y) = f(y) \cdot \frac{dy}{du_1} \cdot du_1 = \frac{f(y)}{f(y)} \cdot du_1$$

soit,  $dF(y) = du_1$  ce qui entraîne :

$$f(u_1) = 1 \text{ avec } 0 \leq u_1 \leq 1 \text{ c.q.f.d.}$$

Appliquons ce résultat :

$$1 - \exp(-y) = U_1$$

et

$$y = -\text{Log } U_1$$

$$\frac{r^2}{2} = -\text{Log } U_1$$

d'où

$$r = \sqrt{-2 \cdot \text{Log } U_1}$$

Pour  $\theta$  on obtient :

$$f(\theta) = \frac{1}{2\pi} \rightarrow F(\theta) = \int_0^\theta f(\theta) \cdot d\theta = \frac{\theta}{2\pi}$$

$$\text{avec } 0 \leq \theta \leq 2\pi$$

Si  $U_2$  est distribuée suivant  $N(0,1)$  on peut poser :

$$F(\theta) = U_2 \text{ d'où}$$

$$\theta = 2\pi \cdot U_2$$

En conclusion, si les deux variables  $U_1$  et  $U_2$  sont continues uniformes les variables

$$R = \sqrt{-2 \text{ Log } U_1}$$

$$\theta = 2\pi U_2$$

sont distribuées exactement comme  $r$  et  $\theta$ . Ceci implique que :

$$X = \sqrt{-2 \text{ Log } U_1} \cdot \cos(2\pi U_2)$$

$$Y = \sqrt{-2 \text{ Log } U_1} \cdot \sin(2\pi U_2)$$

sont deux variables aléatoires indépendantes distribuées exactement suivant  $N(0,1)$

En général il est plus rapide, sur un ordinateur, de travailler sur des petits angles. C'est pourquoi on peut poser :

$$\alpha = \frac{\theta}{4} = \frac{\pi}{2} \cdot U_2$$

et donc :  $0 \leq \sin \alpha \leq 1$  tandis que  $-1 \leq \sin \theta \leq 1$ . Ainsi, en faisant précéder d'un signe au hasard la variable

$$Z = \sqrt{-2 \text{ Log } U_1} \cdot \left( \sin \frac{\pi}{2} \cdot U_2 \right)$$

on obtient une variable aléatoire distribuée exactement suivant  $N(0,1)$ .

### II-3 Sous-programme FORTRAN :

A chacun des deux algorithmes exposés ci-dessus correspond un sous-programme :

a) *Théorème central limite* :

```
SUBRØUTINE GAUSS (IX,S,AM,V,IY)
C SOUS-PROGRAMME NECESSAIRE : HASARD
A=0.0
DØ 50 I=1, 12
CALL HASARD (IX,IY, YFL)
IX=IY
50 V=(A-6.0)*S+AM
RETURN
END
```

1er argument : IX, identique au sous-programme Hasard

2ème argument : S, type réel, c'est la valeur de l'écart-type ( $\sigma$ ) de la loi normale que l'on désire simuler.

3ème argument : AM, type réel, c'est la valeur de la moyenne ( $m$ ) de la loi N ( $m, \sigma$ ) désirée.

4ème argument : V, type réel, c'est le résultat. C'est-à-dire une valeur de la variable aléatoire distribuée suivant N( $m, \sigma$ ) ou N(AM,S).

5ème argument : IY identique au sous-programme Hasard

b) *BOX et MULLER* :

```
SUBRØUTINE NØRMAL (IX,S,AM,V,IY)
PI=3.1415926
CALL HASARD (IX,IY,U1)
IX=IY
ALPHA=SQRT (-2.0*ALOG(U1))
CALL HASAR (IX,IY,U2)
IX=IY
BETA=SIN ((PI/2.0)*U2)
V=(ALPHA*BETA*S)+AM
CALL HASARD (IX,IY,U3)
IX=IY
IF(U3.LE.0.5)V=-V
RETURN
END
```

Les arguments sont identiques à ceux de Gauss.

### III – LA LOI LOG–NORMALE :

#### III – I, Caractéristiques de la loi Log-normale :

Une variable aléatoire  $X$  suit la loi log-normale de paramètre  $m$  et  $\sigma$  si le logarithme népérien de  $X$  suit la loi  $N(m, \sigma)$ . La densité de probabilité est :

$$f(x) = \frac{1}{\sigma \cdot x \cdot \sqrt{2\pi}} \exp \left[ -\frac{1}{2} \cdot \left( \frac{\text{Log}(x) - m}{\sigma} \right)^2 \right]$$

avec  $x > 0$

Le mode est égal à  $\exp(m - \sigma^2)$ .

C'est une loi non symétrique étalée vers la droite (biais positif).

Les moments non centrés sont donnés par :

$$\mu'_r = E(X^r) = \exp \left( r \cdot m + \frac{r^2 \sigma^2}{2} \right)$$

On en déduit :

$$\mu'_1 = E(X) = \exp \left( m + \frac{\sigma^2}{2} \right)$$

et, puisque  $\text{Var}(X) = \mu_2 = \mu'_2 - (\mu'_1)^2$  on trouve

$$\text{Var}(X) = [E(X)]^2 \cdot [\exp(\sigma^2) - 1]$$

$$\mu_3 = \exp(3m) \cdot [\exp(9\sigma^2/2) - 3\exp(5\sigma^2/2) + 2\exp(3\sigma^2/2)]$$

$$\mu_4 = \exp(4m) \cdot [\exp(2\sigma^2)] \cdot [\exp(\sigma^2) - 1] \cdot [\exp(4\sigma^2) + 2\exp(3\sigma^2) + 3\exp(2\sigma^2) - 3].$$

#### III – 2 Processus générateur :

Afin de simuler une variable log-normale de moyenne et de variance données il faut au préalable exprimer  $m$  et  $\sigma$  en fonction de  $E(X)$  et  $\text{Var}(X)$ . On tire de la dernière expression du paragraphe III–I ; ci-dessus :

$$\frac{\text{Var}(X)}{[E(X)]^2} = \exp(\sigma^2) - 1$$

$$\exp(\sigma^2) = \frac{\text{Var}(X)}{[E(X)]^2} + 1$$

En prenant le logarithme népérien des deux membres :

$$\sigma^2 = \text{Log} \left[ \frac{\text{Var}(X)}{[E(X)]^2} + 1 \right]$$

Prenons le logarithme des deux membres de l'équation donnant E(X) :

$$\text{Log } [E(X)] = m + \frac{\sigma^2}{2}$$

d'où :

$$m = \text{Log } [E(X)] - \left\{ \frac{+1}{2} \text{Log} \left[ \frac{\text{Var}(X)}{[E(X)]^2} + 1 \right] \right\}$$

Puisque Log x est distribuée suivant N(m,σ), la variable normale réduite z est définie par :

$$z = \frac{\text{Log } x - m}{\sigma}$$

soit :

$$\text{Log } x = m + \sigma \cdot z$$

$$x = \exp (m + z \cdot \sigma)$$

En II-2. on a montré que :

$$z = \frac{\sum_{i=1}^K r_i - K/2}{\sqrt{K/12}} \quad \text{où } r_i \sim U(0,1)$$

suit la loi N(0,1). On peut donc écrire :

$$x = \exp \left[ m + \sigma \cdot \left( \frac{K}{12} \right)^{1/2} \cdot \left( \sum_{i=1}^K (r_i) - \frac{K}{2} \right) \right]$$

Si on désire plus de précision, on pourra utiliser l'algorithme de Box et Muller (voir II-2). pour simuler z.

### III – 3. Sous-programme Fortran :

Comme pour le sous-programme Gauss, on prendra K = 12.

```

SUBRØUTINE LØGNØR (IX,SX,EX,X,IY)
SUM=0.0
A = ALØG (((SX + SX)/(EX * EX)) + 1.0)
SIGMA=SQRT (A)
AMØY=ALØG (EX) -0.5*A
DØ 5 I=1, 12
CALL HASARD (IX,IY,YFL)
IX=IY
5 SUM=SUM+YFL
X=EXP (AMØY+SIGMA*(SUM-6.0))
RETURN
END
    
```

- 1er argument : IX, identique au sous-programme Hasard.  
 2ème argument : SX, type réel, c'est l'écart-type de la loi log-normale désirée.  
 3ème argument : EX, type réel, c'est la valeur de la moyenne de la loi log-normale.  
 4ème argument : X, type réel, c'est le résultat. une valeur d'une variable aléatoire log-normale.  
 5ème argument : IY, identique au sous-programme Hasard.  
 Sous-programme nécessaire dans cette subroutine : sous-programme Hasard.

#### IV. LA DISTRIBUTION DU KHI-DEUX :

##### IV-1. Caractéristiques de la loi du Khi-deux :

Si  $X_1, X_2, \dots, X_n$  sont  $n$  variables aléatoires indépendantes  $N(0,1)$ , la variable aléatoire :

$X = X_1^2 + X_2^2 + \dots + X_n^2$  est distribuée suivant une loi du khi-deux à  $n$  degrés de liberté. Sa fonction de densité de probabilité est donnée par :

$$f(x) = \frac{\left(\frac{n}{2} - 1\right)}{x^{\frac{n}{2}}} \cdot \exp(-x/2) \text{ avec } x > 0 \text{ et } n = 1, 2, \dots$$

$$\Gamma\left(\frac{n}{2}\right) \cdot 2^{n/2}$$

On note généralement  $X \sim$  khi-deux ( $n$ )

##### IV-2. Processus générateur :

En II-2. (b) il a été établi que les V.A. :

$$Z = \sqrt{-2 \text{Log } U_1} \cdot \cos(2\pi U_2)$$

où  $U_1$  et  $U_2$  sont indépendantes

$$Y = \sqrt{-2 \text{Log } U_1} \cdot \sin(2\pi U_2)$$

et  $\sim U(0,1)$ .

suivent la loi  $N(0,1)$

Donc la variable aléatoire  $X = Z^2 + Y^2 = -2 \text{Log } U_1$  est distribuée suivant une loi du khi-deux à 2 degrés de liberté. Plus généralement, si  $U_1, U_2, \dots, U_m$  sont indépendamment distribuées suivant  $U(0,1)$  alors :

$$X = -2 \cdot \text{Log}(U_1 \cdot U_2 \cdot \dots \cdot U_m)$$

suit une loi du khi-deux à 2  $m$  degrés de liberté. D'autre part si  $Y$ , indépendamment de  $X$ , est distribuée suivant  $N(0,1)$  il est clair que :

$$Z = X + Y^2 \sim \text{khi-deux à } 2m + 1 \text{ degrés de liberté.}$$

En résumé, si n (nombre de degrés de liberté) est pair :

$$X = -2 \left( \text{Log} \prod_{i=1}^{n/2} U_i \right) \sim \text{khi-deux}(n)$$

si n est impair :

$$X = -2 \text{Log} \left( \prod_{i=1}^{\frac{n-1}{2}} U_i \right) + z^2 \sim \text{khi-deux}(n)$$

où  $z \sim N(0,1)$

#### IV-3. Sous-programme Fortran :

SUBROUTINE KH12 (N,IX,X,IY)

C SØUS-PRØGRAMMES NECESSAIRES : HASARD et NØRMAL

IR=MØD(N,2)

TR=1.0

IF (IR.NE.0) GØTØ 1

K=N/2

DØ 2 I= 1,K

CALL HASARD (IX,IY,YFL)

IX=IY

2 TR=TR\*YFL

X=-2.0\*ALØG(TR)

RETURN

1 K=(N-1)/2

DØ 3 I=1,K

CALL HASARD (IX,IY,YFL)

IX=IY

3 TR=TR\*YFL

S=1.0

A=0.0

CALL NØRMAL (IX,S,A,Z,IY)

IX=IY

X=-2.0\*ALØG (TR))+(Z\*Z)

RETURN

STOP

1er argument : N, type entier, c'est le nombre de degrés de liberté de la loi du Khi-deux.



2ème argument : IX, identique au sous-programme Hasard

3ème argument : X, type réel, c'est le résultat. Une variable aléatoire distribuée suivant la loi khi-deux (n).

4ème argument : IY, identique au sous-programme Hasard.

## V – LA LOI EXPONENTIELLE

### V-1. Caractéristiques de la loi exponentielle :

Une variable aléatoire X possède une distribution exponentielle de paramètre  $\alpha$  si sa fonction de densité de probabilité est :

$$f(x) = \alpha \cdot e^{-\alpha \cdot x} \text{ avec } \alpha > 0 \text{ et } x \geq 0$$

Sa fonction de répartition est :

$$F(x) = \int_0^x \alpha \cdot e^{-\alpha t} dt = 1 - e^{-\alpha x}$$

Le mode de  $f(x)$  est toujours égal à zéro.

Les moments non centrés sont fournis par :  $\mu'_r = \frac{r!}{\alpha^r}$

d'où on tire facilement :

$$\mu'_1 = E(X) = \frac{1}{\alpha}$$

$$\mu_2 = \text{Var}(X) = \mu'_2 - (\mu'_1)^2 = \frac{1}{\alpha^2}$$

On peut donc exprimer le paramètre  $\alpha$  de la loi exponentielle sous la forme :

$$\alpha = \frac{1}{E(X)}$$

### V-2. Processus générateur :

Le procédé le plus simple pour obtenir une variable exponentielle est d'inverser la fonction de répartition :

$$F(x) = 1 - e^{-\alpha x}$$

soit

$$1 - F(x) = e^{-\alpha x}$$

or, comme  $F(x)$ ,  $(1-F(x))$  est compris entre 0 et 1 ; si U est une variable continue uniforme on peut donc écrire :

$$U = e^{-\alpha x}$$

d'où on tire

$$x = \frac{-1}{\alpha} \text{Log}(U) = -E(X) \cdot \text{Log} U$$

**V-3 Sous-programme Fortran :**

SUBROUTINE EXPØN (IX,EX,X,IY)

**C SØUS-PRØGRAMME NECESSAIRE : HASARD**

CALL HASARD (IX,IY,YFL)

IX=IY

X=EX\*ALØG(YFL)

RETURN

END

1er argument : IX, identique au sous-programme Hasard.

2ème argument : EX, type réel, c'est la moyenne exponentielle désirée.

3ème argument : X, type réel, c'est le résultat. X est une variable aléatoire distribuée suivant la loi exponentielle de moyenne EX.

4ème argument : IY, identique au sous-programme Hasard;

**VI – LA DISTRIBUTION LOGISTIQUE :**

**VI-I. Caractéristiques de la loi logistique :**

Citons, pour mémoire, cette distribution utilisée quelquefois dans les simulations de Monte-Carlo. Une variable logistique X a pour fonction de densité de probabilité :

$$f(x) = \frac{\exp[-(x-\alpha)/\beta]}{\beta \cdot \{1 + \exp[-(x-\alpha)/\beta]\}^2} \quad \text{avec } \beta > 0$$

Sa fonction de répartition est :

$$F(x) = \frac{1}{1 + \exp[-(x-\alpha)/\beta]}$$

En procédant comme pour la loi exponentielle, on montre que : Si  $U(0,1)$  alors,

$$x = \alpha - \beta \cdot \text{Log} \left( \frac{1-u}{u} \right)$$

suit la loi logistique.

## VI-2. Sous-programme Fortran :

```
SUBROUTINE LOGIST (IX,A,B,X,IY)
C  SØUS-PRØGRAMME NECESSAIRE : HASARD
CALL HASARD (IX,IY,YFL)
IX=IY
X=A-B*ALØG ((1.0-YFL)/YFL)
RETURN
END
```

1er argument : IX, identique au sous-programme Hasard.

2ème argument : A, type réel, c'est le premier paramètre de la loi logistique.

3ème argument : B, type réel, c'est le second paramètre de la loi ( $B > 0$ ).

4ème argument : X, type réel, c'est le résultat : une variable aléatoire distribuée suivant la loi logistique (A,B).

5ème argument : IY, identique au sous-programme Hasard.

## VII – LA LOI GAMMA (OU DISTRIBUTION DE PEARSON TYPE III) :

### VII-I. Caractéristiques de la loi Gamma :

Une variable aléatoire X suit une loi Gamma de paramètres k et  $\lambda$  si sa fonction de densité de probabilité est :

$$f(x) = \frac{\lambda}{\Gamma(k)} \cdot (\lambda x)^{k-1} \cdot e^{-\lambda x} = \frac{\lambda^k}{(k-1)!} \cdot x^{(k-1)} \cdot e^{-\lambda x}$$

avec  $\lambda > 0$ ,  $x \geq 0$  et  $k = 1, 2, \dots, n, \dots$ .

On note généralement  $X \sim \gamma(\lambda, k)$ .

Les quatre premiers moments de X sont donnés par :

$$\mu'_1 = E(X) = \frac{k}{\lambda}$$

$$\mu_2 = \text{Var}(X) = \frac{k}{\lambda^2}$$

$$\mu_3 = \frac{2k}{\lambda^3}$$

$$\mu_4 = \frac{3k(k+2)}{\lambda^4}$$

Remarquons que :

+ Si  $k = \frac{n}{2}$  et  $\lambda = \frac{1}{2}$  on obtient la distribution du Khi-deux avec n degrés de liberté.

+ Si  $k = 1$  la distribution Gamma est identique à la loi exponentielle de paramètre  $\lambda$ .

La fonction génératrice de moments de la loi Gamma est [9] :

$$M_x(t) = \left(1 - \frac{t}{\lambda}\right)^{-k} \text{ avec } t < \lambda$$

Or, si  $Y$  est une variable aléatoire exponentielle de moyenne  $E(Y) = \frac{1}{\lambda}$ , elle a pour densité :

$$h(y) = \lambda \cdot e^{-\lambda y} \text{ avec } y > 0 \text{ et } \lambda > 0$$

Cette fonction exponentielle admet pour fonction génératrice de moments

$$M_y(t) = \left(1 - \frac{t}{\lambda}\right)^{-1}$$

Si  $Y_1, Y_2, \dots, Y_k$  sont  $k$  variables aléatoires indépendantes distribuées chacune suivant  $h(y)$ , alors la fonction génératrice de moments de la variable :

$$X = \sum_{i=1}^k Y_i$$

est

$$M_x(t) = \left(1 - \frac{t}{\lambda}\right)^{-k} \text{ (voir [10])}$$

soit exactement la fonction génératrice de la loi Gamma. En conséquence, la variable aléatoire

$$X = \sum_{i=1}^k Y_i \text{ est distribuée suivant une loi Gamma } (\lambda, k)$$

### VII-2. Processus générateur :

On sait (voir V-2). que si  $U_i$  suit  $U(0,1)$  alors :

$$Y_i = -\frac{1}{\lambda} \text{Log}(U_i) \text{ suit une loi exponentielle de paramètre } \lambda. \text{ Donc si } U_i,$$

$U_2, \dots, U_k$  sont  $k$  variables aléatoires indépendantes distribuées suivant  $U(0,1)$  on en déduit que :

$$X = \sum_{i=1}^k Y_i = -\frac{1}{\lambda} \text{Log}\left(\prod_{i=1}^k U_i\right)$$

est distribuée suivant la loi Gamma  $(\lambda, k)$ .

### VII-3. Sous-programme Fortran :

SUBROUTINE GAMMA (IX,ALAMB,K,X,IY)

C SØUS-PROGRAMME NECESSAIRE : HASARD

TR=1.0

```

DØ 1 I=K
CALL HASARD (IX,IY,U)
IX=IY
1 TR=TR*U
X=- (ALØG(TR))/ALAMB
RETURN
END

```

1er arguemnt : IX ; identique au sous-programme Hasard.

2ème argument : ALAMB, type réel, c'est le paramètre  $\lambda$  de la loi Gamma ( $\lambda > 0$ ).

3ème argument : K, type entier, c'est le deuxième argument de la loi Gamma ( $k > 0$ ).

4ème argument : X, type réel, c'est le résultat. X = une valeur de la V.A. distribuée suivant Gamma ( $\lambda, k$ ).

5ème argument : IY, identique au sous-programme Hasard.

## VIII – LA LOI BETA

### VIII–1. Caractéristiques de la loi Beta :

Une variable aléatoire X est distribuée suivant la loi Beta de paramètres a et b si sa densité de probabilité est :

$$f(x) = \frac{\Gamma(a+b) \cdot x^{a-1} \cdot (1-x)^{b-1}}{\Gamma(a) \cdot \Gamma(b)}$$

avec  $0 < x < 1$ ,  $a = 1, 2, \dots$  et  $b = 1, 2, \dots$

Les deux premiers moments de cette loi sont :

$$\mu'_1 = E(X) = \frac{a}{a+b}$$

$$\mu_2 = \text{Var}(X) = \frac{a \cdot b}{(a+b)^2 \cdot (a+b+1)}$$

On la note généralement  $\beta(a,b)$ .

### VIII.2. Processus générateur :

On montre que, si  $X_1$  et  $X_2$  sont deux variables aléatoires indépendantes distribuées respectivement suivant les lois Gamma ( $\lambda, k_1$ ) et Gamma ( $\lambda, k_2$ ), la variable  $X_1 + X_2$  est distribuée suivant Gamma ( $\lambda, k$ ) avec

$$k = k_1 + k_2.$$

En outre, la variable  $X = \frac{X_1}{X_1 + X_2}$  (où  $0 < x < 1$ ) suit une loi  $\beta(k_1, k_2)$ .

Donc pour obtenir une variable  $X \sim \beta(k_1, k_2)$  on pourra effectuer le rapport de deux variables Gamma Y et Z telles que :

$$\left. \begin{array}{l} Y = X_1 \sim \text{Gamma}(\lambda, k_1) \\ Z = X_1 + X_2 \sim \text{Gamma}(\lambda, k_1 + k_2) \end{array} \right\} \Rightarrow \frac{Y}{Z} \sim \beta(k_1, k_2)$$

### VIII-3. Sous-programme Fortran :

SUBROUTINE BETA (IX,K1,K2,X,IY)

C SØUS-PRØGRAMMES NECESSAIRES : HASARD ET GAMMA

ALAMB=1.0

CALL GAMMA (IX,ALAMB,K1,X1,IY)

IX=IY

CALL GAMMA (IX,ALAMB, K2,X2,IY)

IX=IY

Z=X1+X2

X=X1/Z

RETURN

END

1er argument : IX, identique au sous-programme Hasard.

2ème argument : K1, type entier, c'est le premier paramètre de la loi Béta ( $K1 > 0$ )

3ème argument : K2, type entier, c'est le second paramètre de la loi Béta ( $K2 > 0$ )

4ème argument : X, type réel, c'est le résultat.  $X \sim \beta(k_1, k_2)$ .

5ème argument : IY, identique au sous-programme Hasard.

### IX- LA LOI DE F :

#### IX-1. Caractéristiques de la loi de F :

Une variable X est distribuée suivant la loi de F avec m et n degrés de liberté si sa densité est :

$$f(x) = \frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \cdot \Gamma\left(\frac{n}{2}\right)} \cdot \left(\frac{m}{n}\right)^{\frac{m}{2}} \frac{x^{\left(\frac{m}{2} - 1\right)}}{\left(1 + \frac{mx}{n}\right)^{\frac{m+n}{2}}}$$

avec  $x \geq 0$ ,  $m = 1, 2, \dots$  et  $n = 1, 2, \dots$

Les deux premiers moments [11] de cette loi sont :

$$\mu'_1 = E(X) = \frac{n}{n-2} \text{ où } n > 2$$

$$\mu_2 = \text{Var}(X) = \frac{2n^2 (m+n-2)}{m(n-2)^2 \cdot (n-4)} \text{ où } n > 4$$

On note généralement  $X \sim F_{m,n}$ .

### IX-2. Processus générateur :

Si deux variables aléatoires indépendantes Y et Z sont respectivement distribuées suivant deux lois du Khi-deux :  $\chi_m^2$  et  $\chi_n^2$  alors la variable  $X = \frac{Y}{Z}$  est distribuée suivant une loi de F avec m et n degrés de liberté [voir [12]].

### IX-3. Sous-programme Fortran :

```

SUBROUTINE F(IX,M,N,X,IY)
C  SØUS-PRØGRAMMES NECESSAIRES : KH I2 HASARD et NORMAL
CALL KH I2 (IX,M,Y,IY)
IX=IY
CALL KH I2 (IX,N,Z,IY)
IX=IY
IF (Z.EQ.0.0) Z = 1.0 E - 70
X=Y/Z
RETURN
END

```

1er argument : IX, identique au sous-programme Hasard.

2ème argument : M, type entier, c'est le premier nombre de degré de liberté de la loi  $F_{m,n}$ .

3ème argument : N, type entier, c'est le second nombre de degré de liberté de la loi  $F_{m,n}$ .

4ème argument : X, type réel, c'est le résultat,  $X \sim F_{m,n}$ .

5ème argument : IY, identique au sous-programme Hasard.

### X- LA LOI DE STUDENT :

Une variable X qui suit la loi de Student avec n degrés de liberté a pour densité :

$$f(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n \cdot \pi} \cdot \Gamma\left(\frac{n}{2}\right)} \cdot \left(1 + \frac{x^2}{n}\right)^{-\left(\frac{n+1}{2}\right)}$$

avec  $-\infty < x < +\infty$  et  $n = 1, 2, \dots$

Les deux premiers moments de cette loi sont :

$$\mu'_1 = E(X) = 0$$

$$\mu_2 = \text{Var}(X) = \frac{n}{n-2} \text{ où } n > 2$$

On montre [10] que si X est distribuée suivant une loi de Student à n degrés de liberté, la variable  $X^2 \sim F_{1,n}$ . Donc si  $Y \sim F_{1,n}$  la variable  $X = \pm\sqrt{Y}$  est distribuée suivant une loi de Student à n degrés de liberté.

SUBROUTINE STUDENT (IX,N,X,IY)

C SØUS-PRØGRAMMES NECESSAIRES : F,KHI2,HASARD,NØORMAL

CALL F (IX,1,N,Y,IY)

IX=IY

CALL HASARD (IX,IY,R)

IX=IY

X=SQRT (Y)

IF (R.LE.0.5) X=-X

RETURN

END

1er argument : IX, identique au sous-programme Hasard.

2ème argument : N, type entier, c'est le paramètre de la loi de Student  
n = 1, 2, ...)

3ème argument : X, type réel, c'est le résultat. X suit une loi de Student à n degrés de liberté.

4ème argument : IY, identique au sous-programme Hasard.

## XI – LA LOI DE CAUCHY :

X suit une loi de Cauchy si :

$$f(x) = \frac{1}{\pi.(1+x^2)} \text{ où } -\infty < x < +\infty$$

On remarque que si l'on fait n = 1 dans la loi de Student on obtient la loi de Cauchy (puisque :  $\Gamma(1) = 1$  et  $\Gamma(1/2) = \sqrt{\pi}$ ).

Il est donc simple de simuler une variable de Cauchy à partir du sous-programme Studen. On pourra aussi utiliser la méthode de réjection décrite en XV ci-après.

## XII – LA LOI DE POISSON :

### XII–I. Caractéristiques de la loi de Poisson :

On appelle variable de Poisson de paramètre m(>0), la variable aléatoire discrète définie sur l'ensemble des entiers non négatifs par les probabilités :

$$f(x) = \text{Prob}(X=x) = \frac{e^{-m} \cdot m^x}{x!} \text{ avec } m > 0 \text{ et } x = 0, 1, 2, \dots$$



On note généralement  $X \sim P(m)$ .

Le mode  $x_0$  de la loi de Poisson est fourni par la relation

$$m - 1 < x_0 < m$$

Si  $m$  est entier, ces deux valeurs sont modales. Les premiers moments sont :

$$\mu'_1 = E(X) = m$$

$$\mu_2 = \text{Var}(X) = m$$

$$\mu_3 = m$$

$$\mu_4 = m + 3m^2$$

On notera que pour une loi de Poisson on a la relation

$$E(X) = \text{Var}(X) = m$$

Les coefficients de Fisher sont :

$$\gamma_1 = \frac{1}{\sqrt{m}} > 0 \quad \text{Distribution étalée vers la droite.}$$

$$\gamma_2 = \frac{1}{m}$$

Notons que lorsque  $m \rightarrow \infty$  (en pratique  $m > 20$ ), la loi de Poisson converge vers la distribution normale :  $N(m, \sqrt{m})$  [13]

## XII-2. Processus générateur :

Considérons la variable aléatoire discrète  $K$  définie implicitement par les inégalités :

$$\sum_{i=1}^K X_i \leq 1 < \sum_{i=1}^{k+1} X_i$$

où les  $X_i$  sont des variables aléatoires indépendantes distribuées suivant une loi exponentielle de paramètre  $\lambda$ . On sait [voir VII-2] que :

$$Y_k = \sum_{i=1}^K X_i \text{ est alors distribuée suivant une loi } \gamma(\lambda, k).$$

Désignons par  $H$  la fonction de répartition de  $K$  :

$$H(k) = \text{Prob} [K \leq k].$$

On en déduit que la densité de  $K$  est :

$$h(k) = H(k) - H(k - 1)$$

D'autre part, on a évidemment :

$$\text{Prob} [K \geq k] = \text{Prob} [Y_k \leq 1]$$

Soit :

$$\text{Prob } [K \geq k] = \int_0^1 \frac{\lambda (\lambda y)^{k-1}}{(k-1)!} \cdot e^{-\lambda y} \cdot dy$$

où on a posé  $y = Y_k$

$$\begin{aligned} \text{Or par définition : } \text{Prob } [K \geq k] &= 1 - \text{Prob } [k < k] \\ &= 1 - H(k-1) \end{aligned}$$

Donc :

$$H(k-1) = 1 - \int_0^1 \frac{\lambda (\lambda y)^{k-1}}{(k-1)!} \cdot e^{-\lambda y} \cdot dy$$

Soit encore :

$$H(k-1) = \int_1^\infty \frac{\lambda (\lambda y)^{k-1}}{(k-1)!} \cdot e^{-\lambda y} \cdot dy$$

On en déduit immédiatement :

$$H(k) = \int_1^\infty \frac{\lambda (\lambda y)^k}{k!} \cdot e^{-\lambda y} \cdot dy$$

Il en résulte que :

$$h(k) = \int_1^\infty \left( \frac{\lambda^{k+1} \cdot y^k \cdot \exp(-\lambda y)}{k!} - \frac{\lambda^k \cdot y^{k-1} \cdot \exp(-\lambda y)}{(k-1)!} \right) dy$$

Posons

$$\begin{aligned} I_k &= \int_1^\infty y^k \cdot e^{-\lambda y} \cdot dy = \int_1^\infty u dv \\ & \quad (u = y^k ; dv = e^{-\lambda y} \cdot dy). \end{aligned}$$

Intégrons  $I_k$  par partie on obtient :

$$I_k = \frac{1}{\lambda} \cdot [\exp(-\lambda) + k \cdot I_{k-1}]$$

Il vient alors :

$$h(k) = \frac{\lambda^k}{k!} \cdot [\lambda \cdot I_k - k \cdot I_{k-1}]$$

soit

$$h(k) = \frac{\lambda^k}{k!} \cdot \exp(-\lambda)$$

La variable  $K$ , définie ci-dessus possède donc bien une distribution de Poisson de paramètre  $\lambda$ .

On peut réécrire les inégalités de définition de  $K$  de la façon suivante :

Si  $U_1, U_2, \dots, U_i, \dots$  sont des variables aléatoires indépendantes continues uniformes on sait que [V-2] :

$$X_i = -\frac{1}{\lambda} \text{Log}(U_i) \sim \text{Exponentielle}(\lambda)$$

On a donc :

$$\sum_{i=1}^{K+1} \left( \frac{\text{Log } U_i}{\lambda} \right) \leq 1 < \sum_{i=1}^K \left( \frac{\text{Log } U_i}{\lambda} \right)$$

Soit :

$$\prod_{i=1}^K U_i \geq e^{-\lambda} > \prod_{i=1}^{K+1} U_i$$

Ce sont ces dernières inégalités qui seront utilisées dans le sous-programme ci-dessous.

### XII-3. Sous-programme Fortran :

SUBROUTINE PØISSN (IX,ALAMB,X,IY)

C SØUS-PROGRAMME NECESSAIRE : HASARD

X=0.0

B=EXP (-ALAMB)

TR=1.0

5 CALL HASARD (IX,IY,U)

IX=IY

TR=TR\*U

IF(TR-B) 10,8,8

8 X=X+1.0

GØTØ5

10 RETURN

END

1er argument : IX, identique au sous-programme Hadard.

2ème argument : Alamb, type réel, c'est le paramètre de la loi de Poisson.  
P (λ).

3ème argument : X, type réel, c'est le résultat.  $X \sim P(\lambda)$ .

4ème argument : IY, identique au sous-programme Hasard.

### XIII – LA LOI BINOMIALE :

#### XIII – 1. Caractéristiques de la loi Binomiale :

Une variable aléatoire discrète X est distribuée suivant une loi binomiale de paramètres n et p si :

$$\text{Prob}[X = x] = C_n^x \cdot p^x \cdot (1-p)^{n-x} \quad \text{avec } x = 0, 1, 2, \dots, n$$

et

$$C_n^x = \frac{n!}{x! (n-x)!} \quad \text{et } 0 \leq x \leq n$$

On note généralement  $X \sim B(n, p)$ .

Le mode  $x_0$  est défini par :

$$(n \cdot p) - q < x_0 < (n \cdot p) + q \quad \text{où } q = 1 - p$$

$x_0$  doit évidemment être entier. Il se peut donc qu'il y ait deux valeurs modales.

Les premiers moments de la loi  $B(n, p)$  sont :

$$\mu'_1 = E(X) = n \cdot p$$

$$\mu_2 = \text{Var}(X) = n \cdot p \cdot q$$

$$\mu_3 = n \cdot p \cdot q (q - p)$$

$$\mu_4 = nqp(1 - 6pq) + 3npq^2$$

On en déduit les coefficients de Fisher :

$$\gamma_1 = \frac{q-p}{\sqrt{npq}} \quad \text{et} \quad \gamma_2 = \frac{1 - 6pq}{npq}$$

Mentionnons enfin que :

a) Si  $n \rightarrow +\infty$  et  $p \rightarrow 0$  de telle sorte que le produit  $n \cdot p \rightarrow \text{cte} = m$ , la loi  $B(n, p)$  tend vers la loi de Poisson  $P(m)$ . En pratique, on utilise l'approximation de Poisson dans les conditions suivantes :

$$n > 50, \quad p < 0,1 \implies B(n, p) \rightarrow P(n \cdot p)$$

$$n > 50, \quad p > 0,9 \implies B(n, p) \rightarrow n - P(n \cdot q)$$

b) Si  $n \rightarrow \infty$ ,  $p$  restant fixe alors la loi  $B(n, p) \rightarrow N(n \cdot p; \sqrt{n \cdot p \cdot q})$ . En pratique on utilise l'approximation normale si tôt que les produits  $n \cdot p$  et  $n \cdot q$  dépassent 15 à 20.

### XIII-2. Processus générateur :

Le procédé le plus simple consiste à utiliser la définition même de la loi Binomiale :

La loi Binomiale de paramètres  $n$  et  $p$  est la loi du nombre total de réalisations d'un événement  $A$  au cours de  $n$  épreuves identiques et indépendantes où  $A$  a la probabilité  $p$  d'être réalisée.

Désignons par  $U_1, U_2; \dots, U_n$  une suite de  $n$  nombres au hasard. Considérons l'évènement  $A =$  le nombre au hasard  $U_i$  est  $\leq p$ .  $p$  étant compris entre 0 et 1, l'évènement  $A$  défini ci-dessus a bien une probabilité de réalisation égale à  $p$ . Le processus de génération est alors le suivant :

- Poser  $x_0 = 0$
- Pour chaque nombre au hasard  $U_i$  ( $i = 1, 2, \dots, n$ ) on pose :

$$\begin{cases} x_i = x_{i-1} + 1 & \text{si } U_i \leq p \\ x_i = x_{i-1} & \text{si } U_i > p \end{cases}$$

- Une fois la suite de nombres  $U_i$  terminée la valeur  $x_n$  obtenue est une variable binomiale  $B(n, p)$ .

En effet, le nombre  $x$  représente bien le nombre total de réalisations de l'évènement  $A$  (de probabilité  $p$ ) au cours de  $n$  épreuves identiques et indépendantes.

### XIII-3. Sous-programmes Fortran :

SUBROUTINE BINOM (IX,P,X,IX)

C SØUS-PRØGRAMME NECESSAIRE : HASARD

X=0.0

DØ 7 I=1,N

CALL HASARD (IX,IY,U)

IX=IY

IF(U-P) 6,6,7

6 X=X+1.0

7 CØNTINUE

RETURN

END

1er argument : IX, identique au sous-programme Hasard.

2ème argument : N, type entier, c'est le premier paramètre de la loi  $B(n, p)$ .

3ème argument : P, type réel, c'est le second paramètre de  $B(n, p)$ .

4ème argument : X, type réel, c'est le résultat.  $X \sim B(n, p)$ .

5ème argument : IY, identique au sous-programme Hasard.

### XIV - LA LOI EXPERIMENTALE QUELCONQUE :

Pour illustrer la méthode prenons un exemple :

On possède un histogramme de fréquences expérimentales et on désire un échantillon de cette distribution.

Le premier stade consiste en la transformation de la distribution de fréquences en un histogramme des fréquences cumulées  $F(x)$  (fig. 1).

Diviser ensuite l'intervalle  $[0, 1]$  de définition des nombres pseudo-aléatoires en un nombre de classes identiques à celui de la distribution expérimentale. Tirer un nombre au hasard, soit par exemple le nombre 0,32 ; on voit sur la figure 1 que la valeur de  $x$  correspondante sera 15. Il faut remarquer que la probabilité

d'obtenir un nombre pseudo-aléatoire compris entre 0,15 et 0,38 est  $= (0,38 - 0,15) = 0,23$ . Or sur l'histogramme il est facile de trouver la probabilité que la valeur de  $x$  soit comprise entre 10 et 20, est exactement égale à 0,23.

Répéter ce processus  $n$  fois si on désire un échantillon de taille  $n$  extrait de la loi expérimentale donnée.

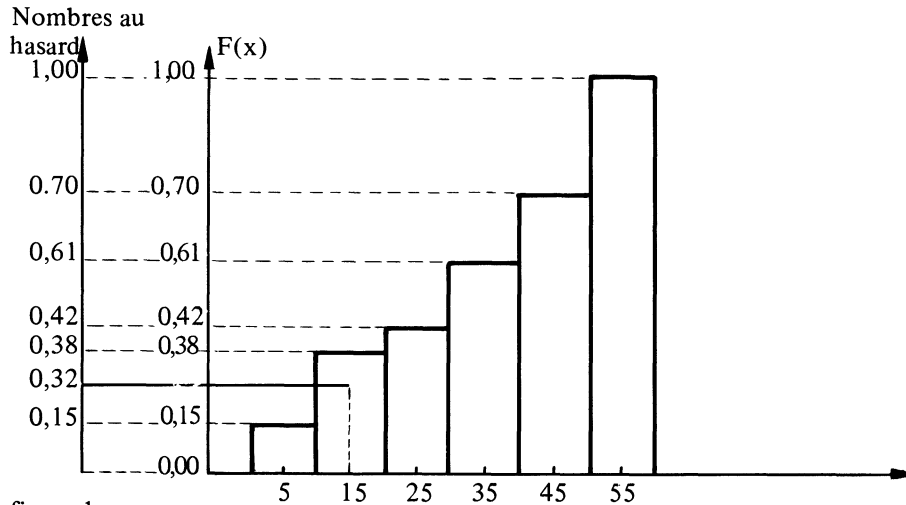


figure 1

On peut programmer cet algorithme de la façon suivante :

```

SUBROUTINE LØIEXP (IX,X,FX,NCLASS,V,IY)
C   SØUS-PRØGRAMME NECESSAIRE : HASARD
DIMENSION X (NCLASS), FX (NCLASS)
CALL HASARD (IX,IY,U)
IX=IY
DØ 1 J=1, NCLASS
1  IF (U.LE.FX (J)) GØTØ 2
2  V=X(J)
RETURN
END

```

1er argument : IX, identique au sous-programme Hasard

2ème argument : X, type réel, c'est le vecteur contenant les centres de classes. A fournir au sous-programme.

3ème argument : FX, type réel, c'est le vecteur contenant les fréquences cumulées de chaque classe. A fournir au sous-programme.

4ème argument : NCLASS, type entier, c'est le nombre de classes de l'histogramme. A fournir au sous-programme.

5ème argument : V, type réel, c'est le résultat. V est une variable aléatoire de même distribution que la loi expérimentale.

6ème argument : IY, identique au sous-programme Hasard.

## XV LA LOI THEORIQUE QUELCONQUE :

Soit une fonction de densité de probabilité bornée, c'est-à-dire telle que :

$$p(x) = \begin{cases} 0 & \text{si } x < a \\ f(x) & \text{si } a \leq x \leq b \text{ et } 0 \leq f(x) \leq M \\ 0 & \text{si } x > b \end{cases}$$

Pour une valeur  $x$  quelconque dans l'intervalle  $[a, b]$  on calcule  $y = \frac{f(x)}{M}$ . Il est clair que :  $0 \leq y \leq 1$

La probabilité qu'une variable aléatoire continue uniforme  $U$  ne soit pas supérieure à  $y$  est exactement  $y$ .

Si on tire au hasard une valeur  $X$  sur l'intervalle  $[a, b]$  et une valeur  $U$  sur  $[0, 1]$ , l'application du test suivant :

Si  $U > y$  : rejeter la valeur de  $x$ .

Si  $U \leq y$  : Accepter la valeur de  $x$ .

conduit à une variable aléatoire  $X$  distribuée suivant  $p(x)$ .

En effet, la probabilité qu'une paire  $(x, u)$  fournisse une valeur de  $x$  acceptable est  $y \cdot dx$ . Il en résulte que la probabilité d'être acceptée pour une valeur quelconque de l'intervalle  $[a, b]$  est :

$$\int_a^b y dx = \frac{1}{M} \cdot \int_a^b f(x) \cdot dx = \frac{1}{M}$$

Finalement, si  $x$  est fixée à l'avance, la probabilité conditionnelle que la paire  $(x, u)$  soit acceptée est :

$$M \cdot y \cdot dx = f(x) \cdot dx.$$

et la fonction de densité de probabilité des valeurs de  $x$  acceptées est précisément  $f(x)$ .

## BIBLIOGRAPHIE

- [1] D.H. LEHMER (1951) – Mathematical methods in large-scale computing units. *Ann. Comp. Lab. Harvard Univ.* 26, 141–146.
- [2] THOMAS H : NAYLOR, J.L. BALINTFY, D.S. BURDICK, KONG–CHU (1968) – Computer Simulation techniques Ed. John Wiley (New York) p. 51–52.
- [3] K.D. TOCHER : The art of simulation (1969) Ed. The English Universities Press Ltd p.77

- [4] W.J. HEMMERLE – Generating pseudorandom numbers on a two's complement machine such as the I.B.M. 360. *Communication of the A.C.M.*, 12, 7, p. 382.
- [5] International Business Machines Corporation – Random number generation and testing. Reference manual (C20–8011), New York 1959.
- [6] MULLER, MERVIN E – A comparison of methods for generating normal deviates on digital computers (1959). *Journal of the association for computing machinery* VI, p.376–383.
- [7] BOX and MULLER (1958) – A note on the generation of normal deviates. *Ann. Math. Stat.*, 28, p.610–11.
- [8] H. CRAMER – Mathematical methods of statistics (1971). Ed. Princeton University Press p. 234.
- [9] M.G. KENDALL and A. STUART – The advanced theory of statistics (1969). Ed. C. Griffin et Cie (Londres) Vol. 1 p.62–63
- [10] A.M. MOOD and F.A. GRAYBILL – Introduction to the theory of statistics (1969) Ed. Mc Graw Hill
- [11] WILKS, S.S. – Mathematical statistics (1962) Ed. John Wiley (New York) p. 187
- [12] B.L. MYERS and N.L. ENRICK – Statistical functions (1970) Ed. Kent State University Press p. 61–63
- [13] G. CALOT – Cours de calcul des Probabilités (1971) Ed. Dunod p. 330–331