# Aléatoire et calculatrices

Une calculatrice peut simuler des tirages aléatoires de nombres compris entre 0 et 1 ( touche Random : Rand, Ran#, Rn# selon les machines ).

**A  Répartition des nombres tirés**

Nous nous proposons de vérifier la répartition des nombres tirés en affichant 100 points dont les coordonnées sont des tirages successifs.

1) Choisir comme fenêtre d'affichage : Pour Texas Instruments $0 \leqslant X \leqslant 1$ et $0 \leqslant Y \leqslant 1$ ou pour Casio $0 \leqslant X \leqslant 1$ et $-0{,}2 \leqslant Y \leqslant 1$.
Entrer le programme suivant :
TI 82 ou TI 80 :
    ClrDraw: For (I,1,100): rand → X: rand → Y:Pt-On(X,Y): End
TI 81
    ClrDraw: 0 → N: Lbl 0: rand → X: rand → Y:Pt-On(X,Y): IS>(N,100): Goto 0
Casio
    Cls: 0 → N: Lbl 0: Ran# → X: Ran# → Y: Plot X,Y: N+1 → N: N<100 ⇒ Goto 0

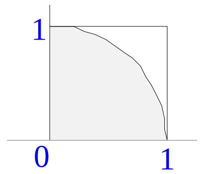
2) Remplacer dans le programme 100 par 1000. La répartition semble-t-elle uniforme ?

**B  Approximation de π par la méthode de Monte-Carlo**

Pour obtenir une valeur approchée de π, on effectue des tirs aléatoires sur un carré de côté 1. Si les tirs sont suffisamment nombreux et uniformément répartis, le rapport du nombre d'impacts atteignant le quart de disque au nombre total d'impacts est proche du rapport des aires du quart de disque et du carré, donc proche de $\frac{\pi}{4}$.

1) Construire un programme sur votre calculatrice :
- Tirer deux nombres aléatoires X et Y
- Si le point de coordonnées X et Y appartient au quart de disque ( *ie* $X^2 + Y^2 \to 1$ ) incrémenter une mémoire N
- Effectuer 1000 tirs successifs
- Calculer $4\frac{N}{1000}$

2) Utiliser le programme pour donner une valeur approchée de π.
    Comparer cette valeur à celle proposée par la calculatrice.




**C  Simulation d'un jet de pièce ou de dé**

1) Utiliser la calculatrice pour calculer la partie entière de chacun des nombres suivants ( touche Int ) : 37,5 ; 0,35 ; $\frac{16}{3}$ ; -2,7 ; -12.
    On retrouve ainsi la définition de la partie entière d'un nombre réel.
2) Taper plusieurs fois la séquence : Int(2Rand) ou Int(2Ran#). Qu'obtient-on ? Pourquoi ?
    Expliquer comment cette séquence peut simuler le tirage aléatoire d'une pièce de monnaie.
3) Quelle séquence devrait-on écrire pour simuler un jet de dé ( éventualités : 1, 2, 3, 4, 5, 6 ) ?

**D  Simulation de 1000 lancers de dé**

On se propose de simuler 1000 lancers d'un dé à jouer ; on notera les fréquences d'apparition de chaque éventualité puis la moyenne des tirages.

1) Construire un programme en suivant les instructions :
    a) Mettre à zéro six mémoires indicées
        Pour Casio, créer d'abord six mémoires indicées Z[]
        Pour TI 81, utiliser les éléments d'une matrice [A] de six lignes et une colonne
        Pour TI 82 ou TI 80, utiliser une liste, par exemple $L_1$, indicée par $L_1(1)$, $L_2(2)$, ...
    b) Tirer un nombre entier aléatoire de 1 à 6
    c) Incrémenter la mémoire qui a pour indice cet entier
    d) Recommencer (de manière à effectuer 100 lancers) les séquences b et c
    e) Afficher l'effectif de chacune des mémoires
    f) Calculer la moyenne des tirages
2) En supposant le dé bien équilibré, quelle devrait être la fréquence d'apparition de chaque éventualité ?
    Combien devrait valoir la moyenne des tirages ?
    Comparer ces valeurs théoriques aux valeurs trouvées grâce au programme du 1)

**Programmes pour Casio 9900**

• Approximation de $\pi$ par la méthode de Monte-Carlo

0 → N
0 → I
Lbl 0
Ran# → X
Ran# → Y
(X² + Y² ≤ 1) ⇒ N + 1 → N
I + 1 → I
I < 1000 ⇒ goto 0
4(N ÷ 1000) ◢

•Simulation d'un jet de pièce ou de dé

Int(6Ran#) + 1

•Simulation de 1000 lancers de dé

0 → A~Z
Lbl 0
Int(6Ran#) + 1 → Z
Isz A[Z]
Isz A
A < 1000 ⇒ goto 0
"1" : B ◢
B ÷ 1000 ◢
"2" : C ◢
C ÷ 1000 ◢
"3" : D ◢
D ÷ 1000 ◢
"4" : E ◢
E ÷ 1000 ◢
"5" : F ◢
F ÷ 1000 ◢
"6" : G ◢
G ÷ 1000 ◢

Programme pour la TI 85

Cldrw
0 → N
For (I,1,1000)
Ran → X : Ran → Y
If X² + Y² ≤ 1
N + 1 → N
If X² + Y² ≤ 1
Pton(x,y)
End
Disp4N/1000

**Programme pour TI 83**

Clr List $L_1$ : 6 → dim($L_1$)
0 → A : Lbl 0
int(6Rand + 1) → I
$L_1$(I) + 1 → $L_1$(I)
A + 1 → A : if A < 1000 goto 0

1 → B : Clrhome : Lbl 1
Output(B,1,B)
Output(B,3,$L_1$(B))
Output(B,8,B + 3)
Output(B,10,$L_1$(B+3))
B + 1 → B : if B ≤ 3 : goto 1
Output(U,1,"MOY")
Output(4,6,mean($L_1$))