$1^{\text{ère}}$ S

Schéma de Bernoulli (1)

Plan du chapitre :

I. Exemple 1

II. Exemple 2

III. Épreuve de Bernoulli

IV. Schéma de Bernoulli

V. Nombre de succès

VI. Programmes Python de simulations

VII. Variable de Bernoulli

VIII. Méthode de Monte-Carlo

I. Exemple 1

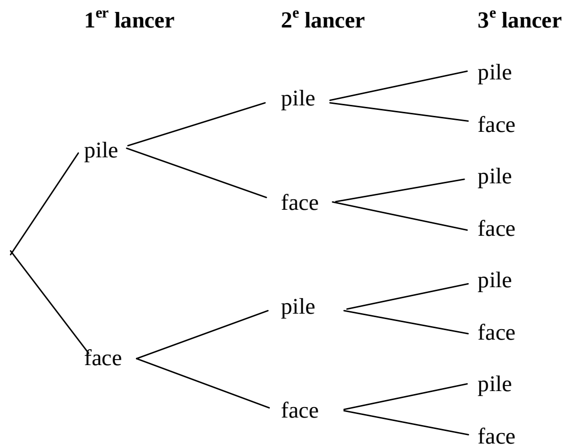
● Protocole expérimental

On lance 3 fois une pièce non truquée.
On note les côtés qu'elle présente dans l'ordre d'apparition.

● Événements considérés

A : « obtenir que des piles »
B : « obtenir au moins une fois face »

● Arbre (pas obligatoire)



Arbre binaire

● Probabilités

$$P(\mathrm{A}) = P(\text{pile} - \text{pile} - \text{pile})$$
$$= P(\text{pile}) \times P(\text{pile}) \times P(\text{pile})$$
$$= \left(\frac{1}{2}\right)^3$$
$$= \frac{1}{8}$$

lancers indépendants
principe multiplicatif

$$\mathrm{B} = \overline{\mathrm{A}}$$

$$P(\mathrm{B}) = 1 - P(\mathrm{A}) = 1 - \frac{1}{8} = \frac{7}{8}$$

● Généralisation : n lancers ($n \geqslant 1$)

$$P(\mathrm{A}) = \left(\frac{1}{2}\right)^n$$

$$P(\mathrm{B}) = 1 - \left(\frac{1}{2}\right)^n$$

II. Exemple 2

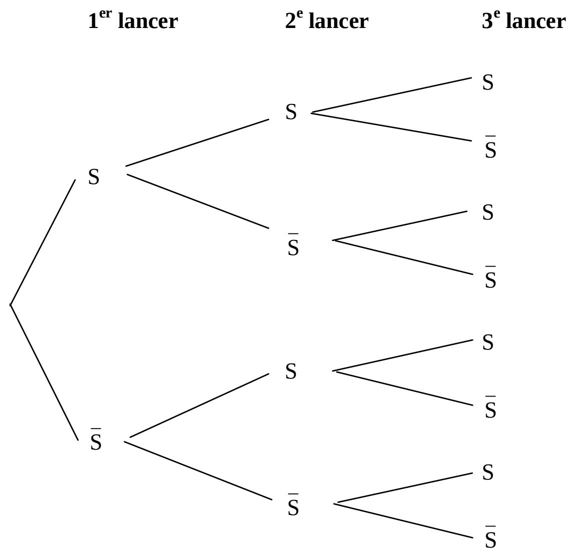
• Protocole expérimental

On considère un dé truqué tel que la probabilité d'obtenir un numéro pair en un lancer soit égale à 0,4.
On le lance 3 fois de suite.
On note les numéros des faces supérieures dans l'ordre.

• Événements considérés

A : « obtenir exactement 2 numéros pairs sur les 3 »
B : « obtenir au moins 2 fois un numéro pair »

• Arbre (pas obligatoire)



Arbre binaire

S : « obtenir un numéro pair » (pour un lancer)

$P(S)=0,4$

$P(\overline{S})=1-0,4=0,6$

• Probabilités

A : « 2 S et 1 seul $\overline{S}$ »

$$P(A)=P(S-S-\overline{S})+P(S-\overline{S}-S)+P(\overline{S}-S-S)$$

$\downarrow$ principe multiplicatif

$$=P(S)\times P(S)\times P(\overline{S})+P(S)\times P(\overline{S})\times P(S)+P(\overline{S})\times P(S)\times P(S)=3\times(0,4)^2\times 0,6=0,288$$

B : « obtenir au moins 2 fois un numéro pair »

$$P(B)=P(A)+P(S-S-S)=0,288+(0,4)^3=0,352$$

III. Épreuve de Bernoulli

1°) Définition

On appelle **épreuve de Bernoulli*** une expérience aléatoire dans laquelle on s'intéresse à la réalisation d'un événement S (**succès**) ou de son contraire $\overline{S}$ (**échec**).

Bernoulli* : grande famille de mathématiciens des XVII^e et XVIII^e siècles.
Jacques Bernoulli (1654-1705) : mathématicien suisse, « *Ars conjectandi* » (ouvrage posthume 1713).

2°) Modélisation

épreuve de Bernoulli $\nearrow$ succès S $\quad P(S)=p$

épreuve de Bernoulli $\searrow$ échec $\overline{S}$ $\quad P(\overline{S})=1-P(S)=1-p=q$

3°) Simulation

- simulation « à la main » en utilisant une table de nombres au hasard

- simulation sur calculatrice ou sur ordinateur : voir le paragraphe **VI** sur la simulation du lancer d'une pièce.

4°) Le modèle de l'urne de Bernoulli

On appelle **urne de Bernoulli** une urne ne contenant que des boules blanches et noires.

On suppose que la proportion des boules blanches est égale à p.

On tire une boule au hasard.
On note sa couleur.

En notant S l'événement « obtenir une boule blanche », cette expérience aléatoire est une épreuve de Bernoulli.

IV. Schéma de Bernoulli

1°) Définition

On appelle **schéma de Bernoulli** la répétition d'une même épreuve de Bernoulli dans des conditions identiques indépendantes.

2°) Modélisation

On modélise un schéma de Bernoulli en utilisant les principes de la répétition d'une expérience aléatoire dans des conditions identiques indépendantes.

L'outil essentiel dans ce chapitre sera l'utilisation d'arbres pondérés (**« arbres de Bernoulli »**), arbres binaires à plusieurs niveaux, permettant de calculer des probabilités.

Pour des raisons pratiques évidentes, nous nous limiterons dans ce chapitre à l'étude de schémas de Bernoulli qui sont la répétition de n épreuves de Bernoulli avec $n \leqslant 5$.

Nous verrons dans un chapitre ultérieur des formules permettant de calculer des probabilités en se passant d'arbres, ce qui est particulièrement intéressant lorsque l'on a des valeurs de n qui dépassent 5.

3°) Simulation

- simulation « à la main » en utilisant une table de nombres au hasard

- simulation sur calculatrice ou sur ordinateur : voir le paragraphe **VI** sur la simulation du lancer d'une pièce.

V. Nombre de succès

1°) Variable aléatoire

On répète n fois une même épreuve de Bernoulli dans des conditions identiques indépendantes.

On s'intéresse habituellement au nombre de succès à l'issue des n épreuves.

On note **X la variable aléatoire qui compte le nombre de succès à l'issue des *n* épreuves**.

2°) Exemple

$n = 3$

On pose $P(\text{S}) = p$ et $P(\overline{\text{S}}) = q$.
On a donc $p + q = 1$.

X peut prendre les valeurs 0, 1, 2 ou 3.

On établit aisément la loi de probabilité de X donnée dans le tableau suivant.

x_i	0	1	2	3
$P(\text{X} = x_i)$	q^3	$3pq^2$	$3p^2q$	p^3

Pour 0 succès : chemin $\overline{\text{S}}$ - $\overline{\text{S}}$ - $\overline{\text{S}}$ donc la probabilité est $q \times q \times q = q^3$

Pour 3 succès : chemin S-S-S donc la probabilité est $p \times p \times p = p^3$

Pour 1 succès : chemins S- $\overline{\text{S}}$ - $\overline{\text{S}}$, $\overline{\text{S}}$-S-$\overline{\text{S}}$, $\overline{\text{S}}$ - $\overline{\text{S}}$ -S donc la probabilité est
$p \times q \times q + q \times p \times q + q \times q \times p = 3pq^2$

Pour 2 succès : chemins S- S-$\overline{\text{S}}$, S-$\overline{\text{S}}$-S, $\overline{\text{S}}$-S-S donc la probabilité est
$p \times p \times q + p \times q \times p + q \times p \times p = 3p^2q$

On peut vérifier aisément que la somme des probabilités est égale à 1.

En effet : $\displaystyle\sum_{i=1}^{i=4} P(\text{X} = x_i) = q^3 + 3q^2p + 3pq^2 + p^3 = (p+q)^3 = 1^3 = 1$.

On utilise l'identité remarquable cubique : $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$.

4°) Simulation

- simulation « à la main » en utilisant une table de chiffres au hasard (voir exercices)

- simulation sur calculatrice ou sur ordinateur

5°) Loi du nombre de succès

On note p la probabilité d'un succès pour l'épreuve de Bernoulli considérée
et n le nombre de répétitions de l'épreuve de Bernoulli.

La variable aléatoire X suit une loi de probabilité X appelée **loi binomiale de paramètres *n* et *p***.

Cette loi sera étudiée dans un chapitre ultérieur.

2 cas particuliers :

Par application du principe multiplicatif, on obtient :

- $P(\text{X} = 0) = P(\overline{\text{S}} - \overline{\text{S}} - ... - \overline{\text{S}}) = \left[P(\overline{\text{S}})\right]^n = q^n$

- $P(\text{X} = n) = P(\text{S} - \text{S} - ... - \text{S}) = \left[P(\text{S})\right]^n = p^n$

VI. Programmes Python de simulations

1°) Fonction `random()`

On l'importe de la bibliothèque random.
La fonction `random()` renvoie un réel aléatoire entre 0 et 1 (0 compris et 1 exclus) selon une loi de distribution uniforme.
Plus précisément, la fonction `random()` génère un nombre à virgule flottante aléatoire (ou plutôt pseudo-aléatoire) de façon uniforme dans l'intervalle $[0\,;1[$ selon un algorithme qui recrée du hasard artificiellement.

2°) Principe important

Soit p un réel fixé compris entre 0 et 1.
Si on choisit un réel x au hasard dans l'intervalle $[0\,;1[$, la probabilité que $x \leqslant p$ est égal à p.

3°) Simulation d'une épreuve de Bernoulli telle que la probabilité d'un succès soit égale à *p*

- Exemple : simulation du lancer d'une pièce telle que la probabilité d'obtenir pile en un lancer soit égale à p

• Programme général :

On code le succès par 1 et l'échec par 0.
On appelle cette fonction `bern(p)` (bern comme « Bernoulli », c'est un choix purement personnel).
Il s'agit d'une fonction prenant pour argument un réel p entre 0 et 1 qui correspond à la probabilité d'un succès.

```
from random import random

def bern(p):
    r=random()
    if r<= p:
       x=1
    else:
       x=0
    return x
```

• Exemple : simulation du lancer d'une pièce truquée telle que la probabilité d'obtenir pile en un lancer soit égale à $\frac{1}{3}$

On définit une fonction `lancer()` sans argument.

1ère façon : en utilisant la fonction random

```
from random import random

def lancer():
    r=random()
    if r<= 1/3:
       x=1
    else:
       x=0
    return x
```

2e façon : en utilisant la fonction `randint`

```
from random import randint

def lancer():
    r=randint(1,3)
    if r=1:
       x=1
    else:
       x=0
    return x
```


4°) Simulation d'un schéma de Bernoulli

On considère une épreuve de Bernoulli telle que la probabilité d'un succès soit égale à p.
On répète n fois cette épreuve dans des conditions identiques indépendantes.

On appelle la fonction `simul(n,p)`. Elle prend pour arguments un entier naturel n supérieur ou égal à 1 et p un réel compris entre 0 et 1.

On utilise la fonction `bern(p)` donc on suppose que celle-ci a été rentrée auparavant.

On utilise une boucle « Pour ».

```
def simul(n,p):
    X=0
    for i in range(1,n+1):
        x=bern(p)
        X=X+x
    return X
```

On peut remplacer les instructions `x=bern(p)` et `X=X+x` par `X=X+bern(p)`.

5°) Simulation d'une loi géométrique (temps d'attente du premier succès)

On reprend les mêmes notations que précédemment avec une épreuve de Bernoulli que l'on répète dans des conditions identiques indépendantes. On s'intéresse au numéro de l'épreuve qui donne un succès pour la première fois.

On utilise la fonction `bern(p)` donc on suppose que celle-ci a été rentrée auparavant.
On utilise une boucle « Tantque ».

```
    def temps_prem_succes(p):
        T=0
        x=2 # on met n'importe quelle valeur autre que 1 pour que les
épreuves puissent commencer
        while x !=1:
              x=bern(p)
              T=T+1
        return T
```

T est une variable aléatoire dont la loi sera étudiée plus tard (il s'agit d'une loi géométrique de paramètre p).

VII. Variable de Bernoulli

1°) Définition et propriétés

On considère une épreuve de Bernoulli telle que la probabilité d'un succès soit égale à p, où p est un réel compris entre 0 et 1.
On note X la variable aléatoire qui vaut 1 si on obtient un succès et 0 si on obtient un échec.

On pose $q = 1 - p$ (q est la probabilité d'obtenir un échec).

x_i	1	0
$P(X = x_i)$	p	q

Calculons l'espérance et la variance de X.

$$E(X) = 1 \times p + 0 \times q = p$$

$$V(X) = 1^2 \times p + 0^2 \times q - p^2 = p - p^2 = p(1-p) = pq$$ (utilisation de la formule de König-Huygens)

Cette variable aléatoire X a été utilisée dans les programmes de simulation du paragraphe précédent.

La variable X peut sembler pas intéressante. Nous allons voir qu'elle permet néanmoins d'établir des formules d'espérance et de variance.

2°) Schéma de Bernoulli

On répète *n* fois l'épreuve de Bernoulli dans des conditions identiques indépendantes, *n* étant un entier naturel supérieur ou égal à 1.
On numérote les épreuves 1, 2, …, *n*.
Pour tout entier naturel *i* compris entre 1 et *n*, on note X_i la variable aléatoire de Bernoulli associée à l'épreuve numéro *i*.
On note X le nombre de succès à l'issue des *n* épreuves.

On a $X = X_1 + X_2 + ... + X_n$ (on peut écrire $X = \sum_{i=1}^{i=n} X_i$).

On va calculer l'espérance mathématique et la variance de X.

Calcul de l'espérance mathématique :

Pour cela, on utilise la propriété suivante sur les variables aléatoire qui peut se démontrer assez facilement :
« L'espérance mathématique d'une somme de variables aléatoires est égale à la somme des espérances des variables aléatoires ».

$$E(X) = \sum_{i=1}^{i=n} E(X_i)$$

$$= \sum_{i=1}^{i=n} p$$

$$= n \times p$$

Calcul de la variance :

Pour cela, on utilise la propriété suivante sur les variables aléatoire que nous allons admettre :
« La variance d'une somme de variables aléatoires indépendantes est égale à la somme des variance des variables aléatoires ».

$$V(X) = \sum_{i=1}^{i=n} V(X_i)$$ car les X_i sont des variables indépendantes

$$= \sum_{i=1}^{i=n} pq$$

$$= npq$$

NB : Pour $n = 1$, on retrouve les formules du 1°). Une épreuve de Bernoulli est un schéma de Bernoulli avec une seule répétition.

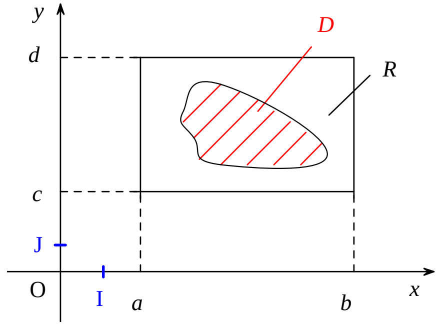
VIII. Méthode de Monte-Carlo

Ce paragraphe est une initiation à une méthode stochastique importante utilisée dans plusieurs domaines aujourd'hui.

1°) Principe général

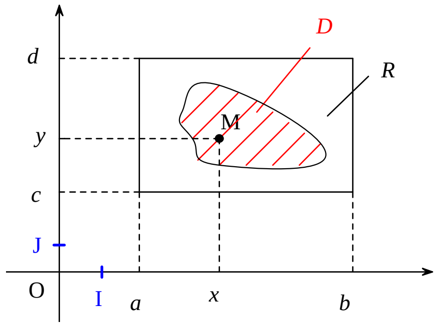
On se place dans le plan muni d'un repère orthogonal (O, I, J).

Soit *D* une partie du plan contenue dans le rectangle *R* défini par le système d'inéquations $\begin{cases} a \leqslant x \leqslant b \\ c \leqslant y \leqslant d \end{cases}$ où *a*, *b*, *c*, *d* sont des réels tels que $a < b$ et $c < d$.



On choisit des points au hasard dans le rectangle *R* et on compte ceux qui appartiennent à *D*.

Le choix d'un point M au hasard dans *R* se traduit par le choix de ses coordonnées $(x\,;\,y)$ avec *x* réel aléatoire dans $[a\,;\,b]$ et *y* réel aléatoire dans $[c\,;\,d]$. On parle de tirage.



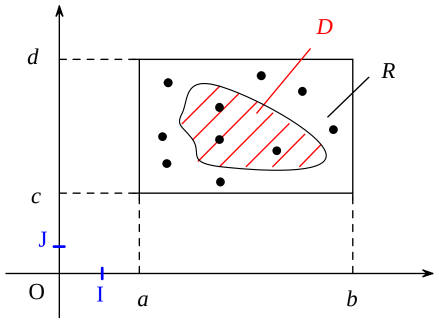
On passe à la simulation informatique.

On définit une fonction tirage() qui simule le tirage d'un point au hasard dans *R*.

```
Fonction tirage()
    x, y ← réel aléatoire dans [a ; b], réel aléatoire dans [c ; d]
    Si le point de coordonnées (x ; y) appartient à D
        Alors t ← 1
        Sinon t ← 0
    FinSi
    Renvoyer t
FinFonction
```

En Python, on utilise la fonction `uniform(a,b)` de la bibliothèque random qui renvoie un réel (ou plutôt un nombre décimal) aléatoire dans l'intervalle $[a, b[$.

On définit une fonction repet(n) qui simule n tirages avec n entier naturel supérieur ou égal à 1.
La variable t est un compteur dont la valeur renvoyée à la fin donne le nombre de points dans *D*.




```
Fonction repet(n)
    t ← 0
    Pour i allant de 1 à n Faire
        t ← t + tirage( )
    FinPour
    Renvoyer t
FinFonction
```

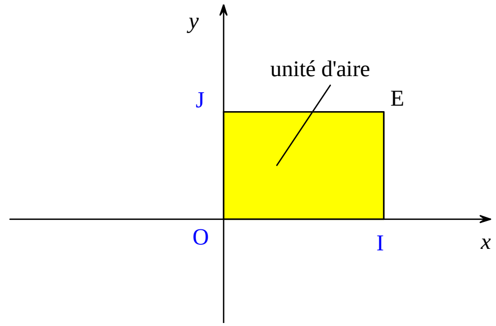
2°) Application aux calculs d'aires

Nous allons voir une application surprenante pour estimer l'aire d'un domaine en utilisant les probabilités.

Une unité de longueur est fixée dans le plan.

On suppose à présent que *D* est une partie du plan qui possède une aire (on dit que *D* est *quarrable*).
On pourrait supposée que *D* est borélienne (notion définie dans le supérieur).
Comme *D* est contenue dans *R*, son aire est finie.

L'unité d'aire dans le plan est donnée par l'aire du rectangle unité OIEJ avec $\text{E}(1\,;1)$.



Le tirage d'un point au hasard dans *D* est une épreuve de Bernoulli conduisant soit à un succès S : « Le point appartient à *D* » soit à un échec $\overline{\text{S}}$: « Le point n'appartient pas à *D* ».

La probabilité de S est $p = \dfrac{\text{aire de } D}{\text{aire de } R}$ soit $p = \dfrac{\text{aire de } D}{(b-a)(d-c)}$.

On effectue n tirages répétés dans des conditions identiques indépendants puis on calcule la fréquence f (la proportion) de points appartenant à *D*.
Cette fréquence est donnée par la formule $f = \dfrac{\text{nombre de points dans } D}{\text{nombre total de points tirés dans } R}$ c'est-à-dire avec les notations précédentes, $f = \dfrac{t}{n}$.

On utilise le principe de l'estimation d'une probabilité par une fréquence observée sur un échantillon en utilisant la version vulgarisée suivante de la loi des grands nombres : « Lorsque n est grand, sauf exception, la fréquence observée est proche de la probabilité. »

On a la convergence $f \xrightarrow[n \to +\infty]{} p$.

On peut donc dire que pour *n* donné, *f* est une estimation de *p*.

En multipliant par l'aire de *R*, on peut dire que $f \times (b-a) \times (d-c)$ est une estimation de l'aire de *D*.

```
Fonction estimation_aire(n)
    t ← 0
    Pour k allant de 1 à n Faire
        x, y ← réel aléatoire dans [a ; b], réel aléatoire dans [c ; d]
        Si le point de coordonnées (x ; y) appartient à D
            Alors t ← t + 1
        FinSi
    FinPour
    Renvoyer t/n × (b – a) × (d – c)
FinFonction
```

Exemple :

On suppose que *R* est un rectangle de longueur 4 et de largeur 3.
On suppose que l'on a effectué $n = 1000$ tirages et qu'à l'issue de ces tirages on a trouvé 600 points dans *D*.

La proportion de points dans *D* est 0,6.

Une valeur approchée de l'aire de *D* est donc $0{,}6 \times 12$ soit 7,2 unités d'aires.

On n'a aucun renseignement sur la précision de cette approximation. Il ne s'agit que d'une estimation.
On peut juste dire que plus le nombre de points tirés est grand plus l'approximation est précise.

Cette méthode présente un intérêt pour des domaines dont on ne sait pas calculer l'aire de manière exacte.

3°) Exemple fondamental : valeurs approchées de π

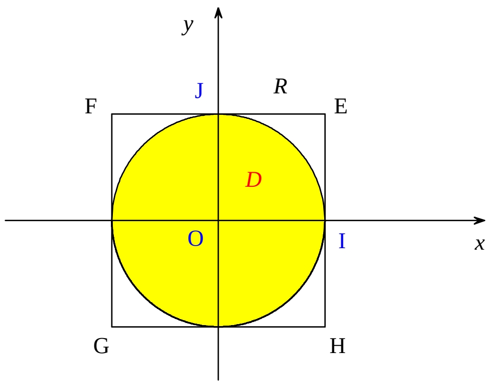
Nous allons voir une application surprenante du paragraphe précédent pour déterminer des approximations de π.

On suppose que le repère $(\mathrm{O}, \mathrm{I}, \mathrm{J})$ est orthonormé.
On utilise un disque ou une partie de disque à l'intérieur d'un carré ou d'un rectangle.

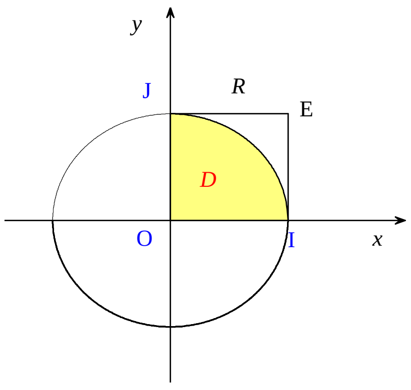
On prend pour *R* un rectangle dont les côtés sont parallèles aux axes et *D* un disque tel que l'intersection de *D* et *R* soit non vide.

En général, on prend pour *D* le disque fermé de centre O et de rayon 1.

Pour *R*, on peut choisir par exemple :
- le carré EFGH avec $\mathrm{E}(1;1)$, $\mathrm{F}(-1;1)$, $\mathrm{G}(-1;-1)$, $\mathrm{H}(1;-1)$ [1er cas] ;
- le carré OIEJ avec $\mathrm{E}(1;1)$ [2e cas].



1er cas



2e cas

Dans le 1er cas, *D* est entièrement contenu dans *R*.
L'aire de *D* est $\pi \times 1^2 = \pi$.
L'aire de *R* est $2^2 = 4$ (car ABCD est un carré de côté 2).
La fréquence ou proportion de points dans *D* donne une estimation de $\frac{\pi}{4}$ d'autant meilleure que *n* est grand.

Dans le 2e cas, l'intersection de *D* et *R* est un quart de disque d'aire $\frac{\pi}{4}$.
L'aire de *R* est $1^2 = 1$ car OIEJ est un carré de côté 1 (on dit que *R* est le « carré unité » du repère).

La condition d'appartenance d'un point $\mathrm{M}(x\,;y)$ à *D* est $x^2 + y^2 \leqslant 1$.
On utilise cette condition très facilement pour la programmation.

On choisit *n* points au hasard dans le carré. On note *f* la fréquence (ou proportion) de points dans *D*.

La valeur de *f* donne une approximation de $\frac{\pi}{4}$ donc 4*f* donne une approximation de π.
On multiplie le résultat de la fréquence par 4 pour avoir une approximation de π.

Programme Python (algorithme) dans le 1er cas :

Dans ce cas, on choisit x et y au hasard dans l'intervalle $[-1\,;1]$.

```
from random import uniform

def estim(n):
    t=0
    for i in range(1,n+1):
        x,y = uniform(-1,1), uniform(-1,1)
        if x**2+y**2<=1 :
           t=t+1
    return 4*t/n
```

L'étude de la précision des approximations obtenues avec cette méthode est un problème complexe que nous aborderons un peu plus tard cette année (mais sans approfondir) avec l'inégalité de Bienaymé-Tchebytcheff et les intervalles de confiance.

4°) Quelques notions générales

On appelle méthodes de Monte-Carlo un ensemble de méthodes utilisant le hasard pour calculer des aires et des volumes.
Ces méthodes se classent en deux grandes catégories :
① la méthode du rejet ;
② la méthode de l'espérance.
Ces méthodes ont été inventées dans les années 1945 par le mathématicien américain Ulam (1909-1984).
Elles ont pris de l'importance en mathématiques avec le développement de l'informatique dans la seconde moitié du XXe siècle.
Elles tirent leur nom de celui de la ville de Monte-Carlo, célèbre pour ses jeux d'argent.
Les méthodes de Monte-Carlo interviennent aussi bien en finance qu'en physique des particules.

La méthode marche aussi en dimension 3.

5°) Quelques remarques d'utilisation

On se place dans le plan muni d'un repère orthogonal $(\mathrm{O}, \mathrm{I}, \mathrm{J})$.
On note R le rectangle unité (autrement dit le rectangle OIKJ où K est le point de coordonnées (1 ; 1)).
Soit D une partie du plan qui possède une aire finie (on dit que D est quarrable).

On cherche l'aire du domaine $D' = R \cap D$ (on admet la propriété : « L'intersection de deux parties quarrables du plan est quarrable »).

On obtient une estimation de l'aire de $D' = R \cap D$.

Cas fréquents :

• Soit f une fonction continue sur $[0\,;1]$ à valeurs positives ou nulles.
On note $\mathscr{C}$ la courbe représentative de f dans le plan.
On note D le domaine du plan en dessous de $\mathscr{C}$ c'est-à-dire limité par l'axe des abscisses, la courbe $\mathscr{C}$ et les droites d'équations $x = 0$ et $x = 1$.
Soit M un point quelconque du plan dont l'abscisse x et l'ordonnée y appartiennent à l'intervalle $[0\,;1]$.
$\mathrm{M} \in D \Leftrightarrow y \leqslant f(x)$

On utilise cette condition en programmation.

• Mêmes hypothèses.
On note D le domaine du plan situé au-dessus de $\mathscr{C}$.
Soit M un point quelconque du plan dont l'abscisse x et l'ordonnée y appartiennent à l'intervalle $[0\,;1]$.
$\mathrm{M} \in D \Leftrightarrow y \geqslant f(x)$

• On considère deux fonctions f et g définies sur $[0\,;1]$ telles que $\forall x \in [0\,;1] \quad f(x) \leqslant g(x)$.
On note D le domaine du plan compris entre les courbes représentatives de f et g.
Soit M un point quelconque du plan dont l'abscisse x et l'ordonnée y appartiennent à l'intervalle $[0\,;1]$.
$\mathrm{M} \in D \Leftrightarrow f(x) \leqslant y \leqslant g(x)$

En programmation, on peut très bien faire marquer les points qui sont dans D.