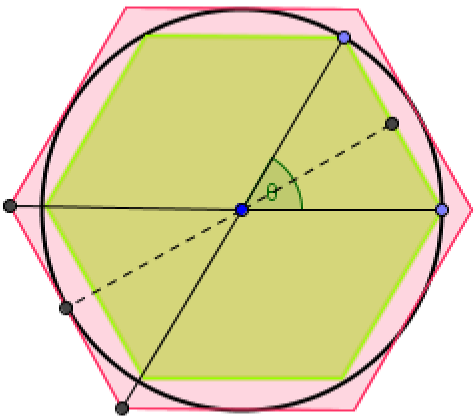
I Géométrie

I.1 Archimède



Méthode 01 : Archimède (-300)

Soit C un cercle de rayon 1. On construit pour tout $n \geqslant 1$ deux polygones réguliers P_n et Q_n ayant 3×2^n côtés, P_n étant inscrit dans C et Q_n exinscrit à C.
On note p_n et q_n les demi-périmètres de P_n et Q_n.

$$p_n < \pi < q_n$$
$$\lim_{n\to+\infty} p_n = \lim_{n\to+\infty} 3 \times 2^n \sin\left(\frac{\pi}{3 \times 2^n}\right) = \pi$$
$$\lim_{n\to+\infty} q_n = \lim_{n\to+\infty} 3 \times 2^n \tan\left(\frac{\pi}{3 \times 2^n}\right) = \pi$$
$$\frac{1}{q_{n+1}} = \frac{1}{2}\left(\frac{1}{p_n} + \frac{1}{q_n}\right) \text{ et } p_{n+1} = \sqrt{p_n \times q_{n+1}}$$

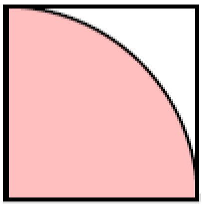
Les suites (p_n) et (q_n) sont adjacentes et $q_n - p_n \leqslant \dfrac{1}{2^n}$

Algorithme

```
import math
def Archimede(N):
        p=3;q=2*math.sqrt(3);
        for i in range(1,N+1):
                q=math.pow((1/2)*(1/p+1/q),-1);
                p=math.sqrt(p*q);
        print("p= ",p, " et q=", q);
```

II Probabilités

II.1 Monte-Carlo



Méthode 03 : Monte-Carlo (1777)

On trace un quart de cercle inscrit dans un carré et on lance des fléchettes sur cette cible. Il faut lancer un très grand nombre de fléchettes, pour obtenir une approximation de $\dfrac{\pi}{4}$

Si le phénomène est aléatoire et uniforme, le rapport du disque au cérré est de $\dfrac{\pi r^2}{4} \times \dfrac{1}{r^2}$

Remarque : Soit X_i la varaible aléatoire valant 1 si la fléchette et dans le disque et $M_n = \dfrac{1}{n}\sum_{i=1}^{n} X_i$ alors

Lancer dans le quart de disque c'est choisir au hasard (uniforme) X dans $[0;1]$ et Y dans $[0;1]$. Pour que la flèche arrive dans le quart de disque, il faut que $\sqrt{X^2+Y^2} \leqslant 1$
On génère donc deux nombres aléatoires distribués dans $[0;1]$ et $[0;1]$ puis on compte le nombre de fois ou le lancer est dans le disque. Une estimation de π sera obtenue en faisant quatre fois le rapport du nombre de lancers dans le disque et et du nombre de lancers à l'extérieur.

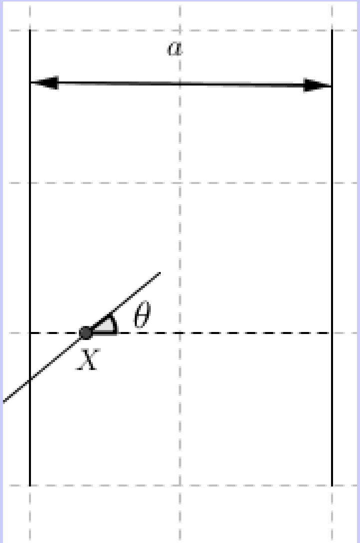
$$P\left(\left|M_n - \frac{\pi}{4}\right| \geqslant \alpha\right) \leqslant \frac{\pi}{n\pi\alpha^2}\left(1-\frac{\pi}{4}\right)$$

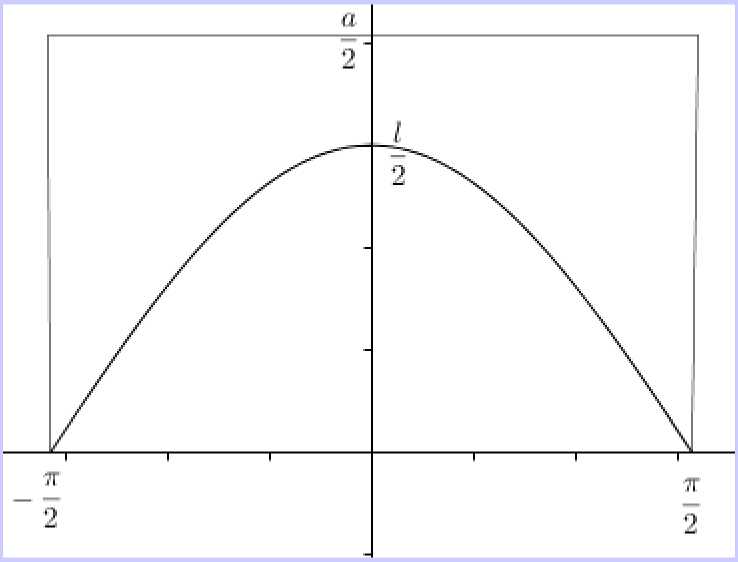
Algorithme

```
import random, math
def MonteCarlo(n):
        c=0;d=0;
        for i in range(1,n+1):
                x=random.uniform(0,1);
                y=random.uniform(0,1);
                d+=1;
                if math.sqrt(x*x+y*y)<=1:c+=1;
        return(4*(c/d));
```

II.2 Aiguilles de Buffon(1777)

Méthode 03 : Aiguille de Buffon (1777)





Développement

La probabilité qu'une aiguille de longueur l lancée sur un parquet dont les lattes ont une largeur a coupe le bord d'une latte est $\frac{2l}{\pi a}$

Si on prend une aiguille de longueur $\frac{a}{2}$ on obtient une probabilité de $\frac{1}{\pi}$

Lancer une aiguille au hasard dans ce cas, c'est choisir au hasard (de faC con uniforme) X dans $\left[0;\frac{a}{2}\right]$ et $\cos(\theta)$ dans $[-1;1]$. Pour que l'aiguille coupe le bord d'une latte, il faut que $X \leqslant \frac{l}{2}\cos\theta$

On génère donc trois nombres aléatoires distribués dans $\left[0;\frac{a}{2}\right]$, $x \in [0,1]$ et $y \in [-1,1]$ puis on compte le nombre d'intersections. Une estimation de π sera obtenue en faisant le rapport du nombre de lancers et du nombre d'intersections.

Remarque : Soit X_i la variable aléatoire valant 1 si l'aiguille coupe une latte et $M_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ alors

$$P\left(\left|M_n - \frac{1}{\pi}\right| \geqslant \alpha\right) \leqslant \frac{1}{n\pi\alpha^2}\left(1-\frac{1}{\pi}\right)$$

Algorithme

```
import random
from math import pi, cos, sqrt
def buffon(n):
        c=0;d=0;
        for i in range(1,n+1):
                X=random.uniform(0,0.5);
                x=random.uniform(0,1)
                y=random.uniform(-1,1);
                if sqrt(x*x+y*y)<1: cos=x/sqrt(x*x+y*y);
                d+=1;
                if X<=(0.25)*cos:c+=1;
        return(d/c);
```

III Analyse

III.1 Séries

Méthode 04 : Approximation par des séries

Fonction Zêta : $\zeta(2) = \sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ et $\zeta(4) = \sum_{n=1}^{+\infty} \frac{1}{n^4} = \frac{\pi^2}{90}$

On peut démontrer ces formules en utilisant la formule de Parseval pour la fonction 2π périodique telle que $f(x) = x$ pour $|x| < \pi$ et la fonction 2π péridique telle que $g(x) = x^2$ pour $|x| < \pi$.

```
import math
def Euler(N):
        S=0;
        for i in range(1,N+1):
                S=S+pow((1/i),2);
        R=math.sqrt(6*S)
        print(R);
```

Fonction arctangente : $\arctan(x) = \sum_{k=0}^{+\infty} \frac{(-1)^k x^{2k+1}}{2k+1}$ et $\arctan(1) = \sum_{k=0}^{+\infty} \frac{(-1)^k}{2k+1} = \frac{\pi}{4}$

En 1682 Leibniz montre que $\left|(x) - \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{2k+1}\right| < \frac{x^{2n+3}}{2n+3}$

En 1706, John Machin : $4\arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right) = \frac{\pi}{4}$

```
import math
def Arctan(N):
        S=0;
        for i in range(0,N+1):
                S=S+pow(-1,i)/(2*i+1);
        print(4*S);

def DeltaArctan(N):
        S=0;
        for i in range(0,N+1):
                T=S;
                S=T+pow(-1,i)/(2*i+1);
                U=S;
                V=U+pow(-1,i+1)/(2*(i+1)+1);
                W=T-(U-T)*(U-T)/(T-2*U+V);
        print(4*W);
```

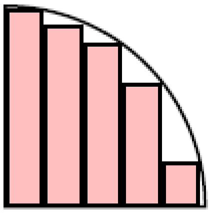
Formule de David Bailey, Peter Borwein et Simon Plouffe (1995) :

$$\sum_{k=0}^{+\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

```
import math
def bbp(N):
        S=0;
        for i in range(0,N+1):
                S=S+pow((1/16),i)*(4/(8*i+1)-2/(8*i+4)-1/(8*i+5)-1/(8*i+6));
        print(S);
```

IV Méthode par des intégrales

IV.1 Méthode des rectangles



Méthode 05 : Méthode des rectangles

On considère le quart de disque de rayon 1 et on partage le segment $[0,1]$ en n subdivisions égales. Sur chque intervalle on remplace l'intégrale par un rectangle dont on connait l'aire.

L'aire du quart de disque de rayon 1 vaut $\int_0^1 f(t)dt = \int_0^1 \sqrt{1-t^2}dt$

En posant : $R_n(f) = \frac{1}{n}\sum_{i=0}^{n} f\left(\frac{i}{n}\right)$ on obtient : $\left|\int_0^1 f(t)dt - R_n(f)\right| \leqslant \frac{1}{2n}$

```
import math
def f(x):
        return math.sqrt(1-x*x);
def Rectangle(n,a,b):
        S=0;
        for i in range(0,n+1):
                x=a+i*(b-a)/n;
                S=S+f(x);
        S=((b-a)/n)*S
        return 4*S;
```

IV.2 Moyenne arithmético-géométrique

Méthode 06 : Moyenne arithmético-géométrique

On note (u), (v), (c) et (S) quatre suites vérifiant

$$u_0 = 1,\ v_0 = \frac{1}{\sqrt{2}} \text{ et } S_0 = \frac{1}{2}$$

Avec comme relations de récurrence :

$$\begin{aligned} u_{n+1} &= \frac{u_n + v_n}{2} \\ v_{n+1} &= \sqrt{u_n v_n} \\ c_{n+1} &= a_{n+1}^2 - v_{n+1}^2 \\ S_{n+1} &= S_n - 2^{n+1} \times c_{n+1} \end{aligned}$$

Alors la suite $p_n = \dfrac{2 \times a_n^2}{S_n}$ converge rapidement vers π

```
mag(epsi):
    a=1;b=1/math.sqrt(2);S=0.5;
    while  abs(a-b)>epsi:
            a,b=(a+b)*0.5,math.sqrt(a*b);
            c=a*a-b*b;
            S=S-2*c;
    return(2*a*a/S);
```