Soufiane Chaieb
L2 informatique

TP 3 : Application de la simulation de Monte Carlo et le calcul des intervalles de confiance

Introduction :

Monte Carlo simulation fait allusion à la fameuse destination des jeux de hasard à Monaco, parce que les résultats obtenus par cette méthode sont aléatoires comme c'est le cas pour les jeux de roulette ou les jeux de dés. Elle a été développée pour la première fois par le mathématicien Stanislaw Ulam.

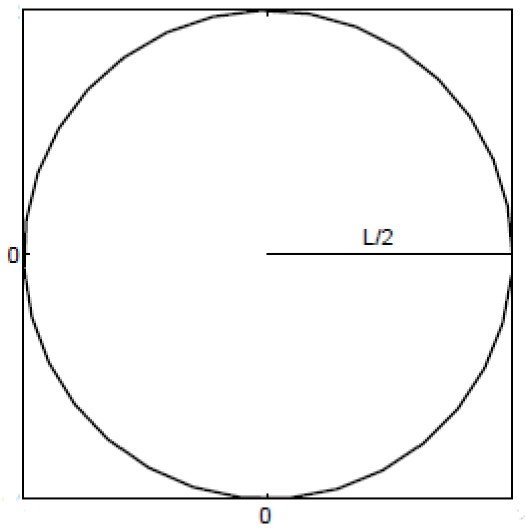
Cette méthode repose sur le fait que la probabilité des résultats obtenus ne peut pas être déterminée grâce à l'utilisation des variables aléatoires, concrètement, elle affecte aux variables incertaines (que l'on peut pas prédire la valeur exacte) des valeurs aléatoires, puis on exécute le modèle et on obtient un résultat, on réexécute ce processus plusieurs fois en gardant les résultats obtenus, et à la fin de la simulation on calcule la moyenne des ces résultats pour en obtenir une valeur estimée.

Cette technique est utilisée pour modéliser la probabilité de différents résultats dans un processus qui ne peut pas être facilement prédit à cause de l'intervention de variables aléatoires, elle est utilisée pour comprendre l'impact de risque et de l'incertitude dans la conception des modèles dans différents domaines tels que la finance, l'ingénierie ou la science.

I. Application de la simulation de Monte Carlo pour l'estimation de π :

1. calcul de π :

Nous pouvons obtenir une estimation de la valeur de π, en utilisant la méthode de Monte Carlo, imaginons que dans un champ on dessine un carrée, la longueur de chacun de ses cotés est L, ce qui donne une surface $S_{carré} = L \times L$ et à l'intérieur on dessine le plus grand cercle possible donc de diamètre égale à L, sa surface sera $S_{cercle} = \pi L^2/4$. Lorsqu'il a commencé à pleuvoir, on calcule le nombre de gouttes de pluie qui se sont tombées dans le carré (cercle inclus), on le note N tel que $N = S_{carré}$ et on compte le nombre de gouttes tombées dans le cercle seulement, on le note N_{in} tel que $N_{in} = S_{cercle}$.



Soit N le nombre de goutte dans le carré, égal à la surface du carré :

$$N = S_{carré} = L \times L \qquad (1)$$

Donc N_{in} le nombre de goutte dans le cercle, est la surface du cercle :

$$N_{in} = S_{cercle} = \pi L^2/4 \qquad (2)$$

Substituant l'équation (1) dans (2), on obtient :

$$\Pi = 4\ N_{in}/N$$

Alors, π égale 4 fois le nombre de gouttes de pluie dans le cercle sur le nombre de gouttes dans le carré(cercle inclus).

2. Implémentation en C :

Pour notre implémentation, on a utilisé le générateur Mersenne Twister, découvert en TP2, connu par sa performance et sa longue période, et qui a été optimisé pour être utiliser dans le cadre de simulations de Monte Carlo. En plus, on a pris (0,0) comme coordonnés du centre du cercle, du rayon 1. Les points seront projetés seulement sur un quart du cercle car sa surface égale à $\pi/4 = N_{in}/N$, donc on peut en déduire π. on prend le quart supérieur droit alors les points générés devront être dans un intervalle [0,1]

L'algorithme suivant a été utilisé pour implémenter nos résultats(vous trouverez le code C équivalent en fichier.c) :

Fonction simPy (entier nbr_points)

{

% nbr_points : c'est le nombre total de gouttes de pluie.

Déclaration des variables

Pour i=0…nbr_point :

X = random %abscisse de chaque point

Y = random %ordonnée de chaque point

%random est un appel au générateur MT pour générer des nombres dans [0,1].

Distance = %distance entre le point et le centre

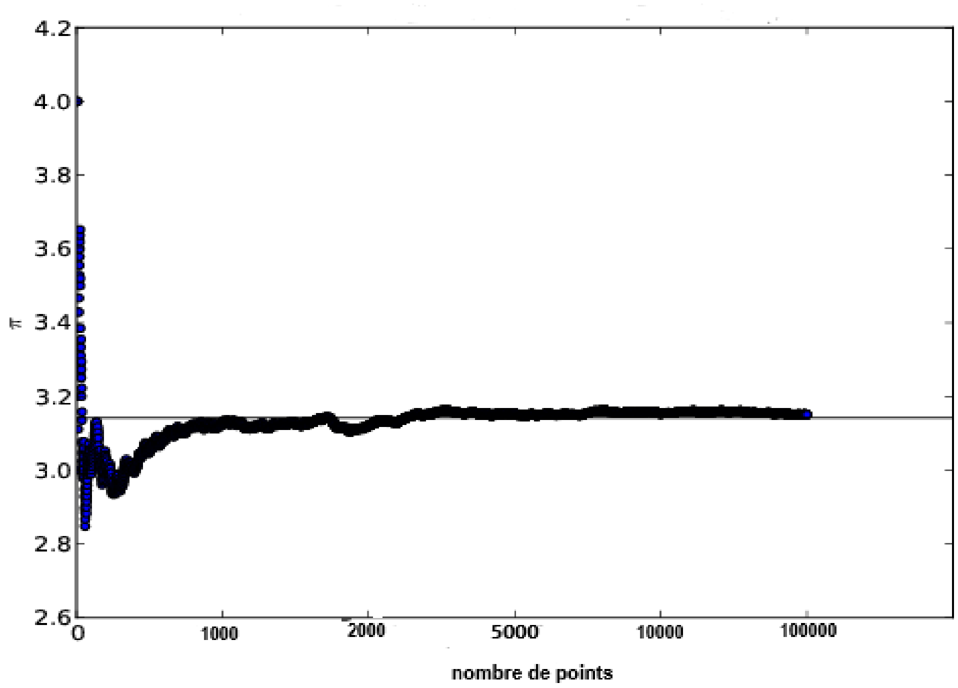
N_{in} = %les points dans le cercle

*simPy = 4*N_{in}/nbr_points ;*

}

Résultats :

Nombre de points	Estimation de π
500	3.09600000
1000	3.12400000
5000	3.14080000
10000	3.14800000
100000	3.14592000
10.000.000	3.14128560
1000.000.000	3.14154551



D'après les résultats on déduit que si le nombre de points est grand, la précision augmente et on approche de plus en plus de la valeur de π, pour N = 5000 on obtient une précision de (10^{-2}), pour N = 10.000.000 une précision de (10^{-3}) et pour N = 1000.000.000 on a une précision de (10^{-4}).

3. Estimation de π :

Pour avoir une estimation de π, on fait plusieurs expériences indépendantes, on garde les résultats et puis on calcule la moyenne des résultats obtenus. Pour cela, on aura besoin de la fonction simPy (vu dans la question précédente) qui calcule π, à partir d'un nombre de points que l'on donne(plus le nombre de points est grand, plus la précision est satisfaisante) et après on précise le nombre d'expériences que l'on veut faire, pour chaque expérience on fait appel à simPy et on enregistre le résultat sans devoir réinitialiser le générateur MT, pour pouvoir achever l'indépendance des expériences. Enfin, on calcul la moyenne des valeurs obtenues et on obtient une estimation de π.

L'algorithme suivant permet de calculer une estimation de π (vous trouverez le code C en fichier.c) :

Fonction estimPy(tableau, nbr_exp, nbr_point)

{

%tableau de double, de taille égale au nbr_exp pour enregistre les résultats.

%nbr_exp : nombre d'expériences. (entre 1 et 30(inclus) expériences autorisées)

Déclaration variables nécessaires

Pour i=0….nbr_exp

Tab[i] = %on garde les résultats de l'appel de simPy dans un tableau

Somme = %on calcule la somme de tous les appels

estimPy = somme / nbr_exp ;

}

Résultats :

Nombre de points	Nombre d'expériences	Estimations de π
1000	30	3.13973333
5000	30	3.14717333
100000	30	3.14236133
100000	15	3.14237333
100000	10	3.14476000
100000	5	3.14363200
1000000	30	3.14114240
10000000	30	3.14155184

On remarque que la précision évolue si on augmente le nombre de points et le nombre d'expériences, pour 10000000 points et 30 expériences on a une précision de (10^{-4}).

4. Calcul de l’intervalle de confiance :

Pour le calcul de l’intervalle de confiance, on a utilisé les résultats de n expériences indépendantes effectuées pour le calcul de la moyenne, cet intervalle encadre l’estimation de π et permet de définir une marge d’erreur R. le niveau que l’on considéré est de 95%.

L’algorithme suivant illustre notre travail :

Fonction conf_Int(tab_exp, nbr_exp, moyenne)

{

Déclaration des variables nécessaires

Pour i=0….nbr_exp

S = %la somme de différence entre le résultat de chaque expérience et la moyenne, au carré.

S^2 = S / (nbr_exp -1) %calcule de la variance

R = %calcule de la marge d’erreur

Conf_Int = [moyenne - R, moyenne + R] ;

}

Résultats :

Nombre de points	Nombre d’expérience	moyenne	Intervalle de confiance
1000	30	3.13973333	[3.12301255, 3.15645411]
10000	30	3.14401333	[3.13768031, 3.15034635]
100000	30	3.14236133	[3.14006487, 3.14465780]
1000000	30	3.14114240	[3.14049188, 3.14179292]
100000000	30	3.14155958	[3.14150001, 3.14161915]

On remarque que l’on trouve un intervalle plus réduit, en augmentant le nombre de points.

Conclusion :

la simulation de Monte Carlo permet de calculer une valeur approchée en utilisant des procédés aléatoires, elle est particulièrement utilisée dans le calcul d’une surface ou d’un volume, qui peuvent être modéliser pour avoir des composantes aléatoires, et ensuite faire des expériences successives, et utiliser la moyenne obtenue comme estimation, et enfin évaluer la précision de cette estimation.