Cours de ***David Hill***

Année Universitaire 2023-2024

ZZ2 promo 25

TP 3

Simulation de Monte Carlo & Intervalles de confiance

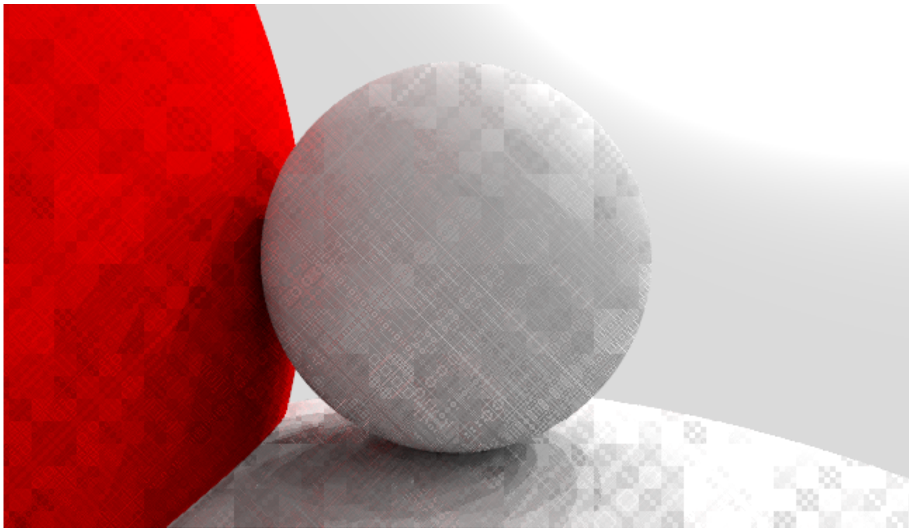

figure 0 - Shader utilisant un lancé de rayon basé sur la méthode Monté-Carlo (lien intégré)

Jouen Martin

Introduction

Dans ce TP, nous avons l'objectif de simuler une valeur approchée du nombre π.
Pour cela, une modélisation stochastique sera utilisée. A l'opposé de la simulation déterministe qui permet d'obtenir un résultat en général plus précis au prix d'une conception plus difficile, la modélisation stochastique est souvent plus simple à concevoir, avec en contrepartie une marge d'erreur.

Quand elle est disponible, la méthode déterministe est à privilégier.

Cependant, bien que la méthode déterministe pour calculer π est connue de tous, l'objectif de ce TP est de pratiquer l'utilisation de la simulation stochastique.

Plus spécifiquement, ce sera la méthode de Monte-Carlo qui sera utilisée pour approximer le nombre π. Cette méthode se base sur l'utilisation d'échantillonnages aléatoires pour estimer des résultats numériques.

1 - Implémentation de la fonction *simPi*

La première étape d'une simulation stochastique consiste à pouvoir faire une simulation. Ici, nous voulons approcher π avec 5 chiffres significatifs. Pour cela, nous devons générer des points à des positions aléatoires dans un carré de côté 2 centré sur l'origine. Il faut maintenant distinguer le cercle inscrit à l'intérieur de ce carré. La proportion du nombre de points se trouvant à l'intérieur du cercle approchera π.

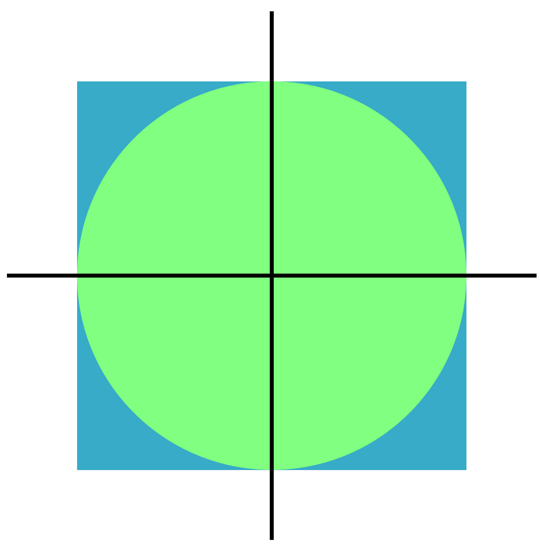

figure 1 - cercle de rayon 1 inscrit dans le carré de côté 2.

Ainsi le nombre de points se trouvant dans la zone verte divisé par le nombre total de point donnera une approximation du résultat voulu.

Pour simplifier la génération de points, ceux-ci se trouveront uniquement dans le sous carré haut-droit. De cette manière les points générés ont des coordonnées positives, ce qui est légèrement plus rapide.

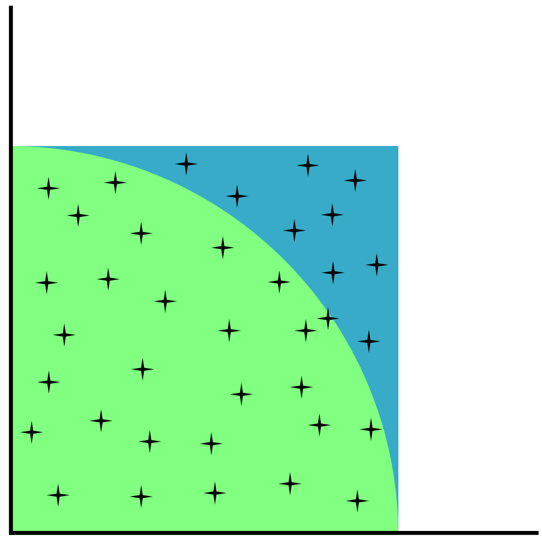

figure 2 - lancé de 37 points sur le quart de cercle

Voici un résultat avec un échantillon de points schématisable (les points sont placés à la main de manière arbitrairement aléatoire). Il y a un total de 37 points.

Pour savoir si un point est dans le quart de cercle, il suffit que sa distance au point $(0;0)$ soit inférieure à 1 (rayon du cercle). On optimise la fonction en ne calculant pas la racine carrée : le résultat exact ne nous intéresse pas, un pré-résultat inférieur ou égal à 1 ne sortira jamais de la zone verte après la racine carré.

La fonction finale est donc :

```
double x, y;
unsigned long somme = 0;
for (unsigned long i = 0; i < n; i++)
{
    x = genrand_real1();
    y = genrand_real1();

    if (x*x + y*y < 1)
    {
        somme++;
    }
}
```

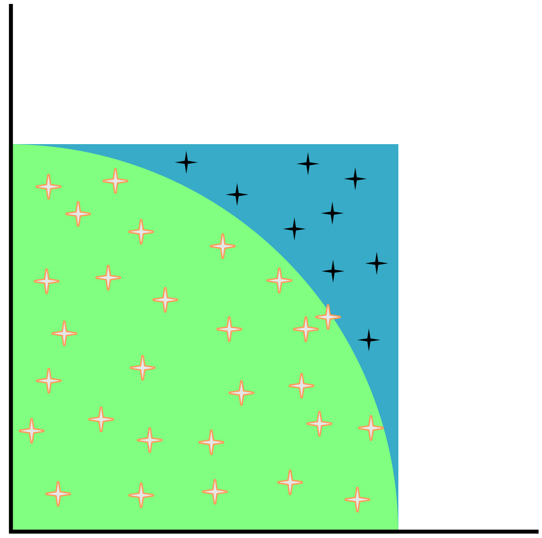

figure 3 - Mise en évidence des points compté pour le ratio de π.

Dont 28 à l'intérieur du cercle (ceux en jaune). On a donc un ratio de $\frac{28}{37} \approx 0.76$ qui est l'aire de la surface verte (le carré a une aire de 1) On multiplie par 4 pour obtenir l'aire du cercle complet. On obtient une approximation de $\pi \approx 3$

On se rend compte qu'il va falloir une grande quantité de points lancés sur la zone pour avoir un résultats avec la précision attendue.

Résultats

Pour n = 1 000
simulation de PI : 3.124000 (2 chiffres significatifs)

Pour n = 1 000 000
simulation de PI : 3.144760 (3 chiffres significatifs)

Pour n = 1 000 000 000
simulation de PI : 3.141546 (5 chiffres significatifs (chance ?))

2 - Expériences indépendantes et moyenne des résultats

Cette fois il suffit faire la moyenne de n simulations de π. Faire cela n'augmentera pas le nombre de chiffres significatif (bien trop long à augmenter), mais à fortifier la crédibilité du résultat, que l'on quantifiera dans la prochaine section.

On peut ensuite vérifier l'erreur avec la constante M_PI fournie dans *math.h*

Résultats

Moyenne de 10 PI à 1 000 lancés : 3.1480000000, erreur : 1.0020395217
Moyenne de 10 PI à 1 000 000 lancés : 3.1412544000, erreur : 0.9998923305
Moyenne de 10 PI à 1 000 000 000 lancés : 3.1415643560, erreur : 0.9999909926

Plus l'erreur est proche de 1, plus le résultat est fidèle à la réalité. On constate entre les deux résultats extrêmes, que malgré 10^6 fois plus de lancés, on n'obtient que deux chiffres significatifs supplémentaires. La convergence est donc très lente avec la méthode de MC.

3 - Intervalles de confiance

Ici nous utiliserons la loi de student pour donner un intervalle de confiance, que nous prendrons à 95 et 99% de confiance.

Les valeurs possibles pour nos expérience sont comprises dans l'intervalle [0 ;4] (dans les cas extrêmement improbables que tous les points se retrouvent à l'extérieur ou à l'intérieur du cercle), cependant, l'intervalle de confiance indique entre quelles valeurs on peut retrouver 95 ou 99 % des valeurs.

Résultats

nombre de lancés	95%	99%
1 000	[3.11901506, 3.17698493]	[3.10677321, 3.18922678]
1 000 000	[3.14011344, 3.14239535]	[3.13963156, 3.14287723]
1 000 000 000	[3.14156493, 3.14160374]	[3.14155820, 3.14161046]

Tableau sur 10 simulations de π.

nombre de lancés	95%	99%
1 000	[3.12301255, 3.15645411]	[3.11721514, 3.16225152]
1 000 000	[3.14053422, 3.14176017]	[3.14032169, 3.14197270]
1 000 000 000	[3.14153600, 3.14159270]	[3.14152403, 3.14160467]

Tableau sur 30 simulations de π.

nombre de lancés	95%	99%
1 000	[3.12941568, 3.15511764]	[3.12517486, 3.15935847]
1 000 000	[3.14139845, 3.14224140]	[3.14039087, 3.14157072]
1 000 000 000	[3.14156816, 3.14160416]	[3.14157263, 3.14159970]

Tableau sur 60 simulations de π.

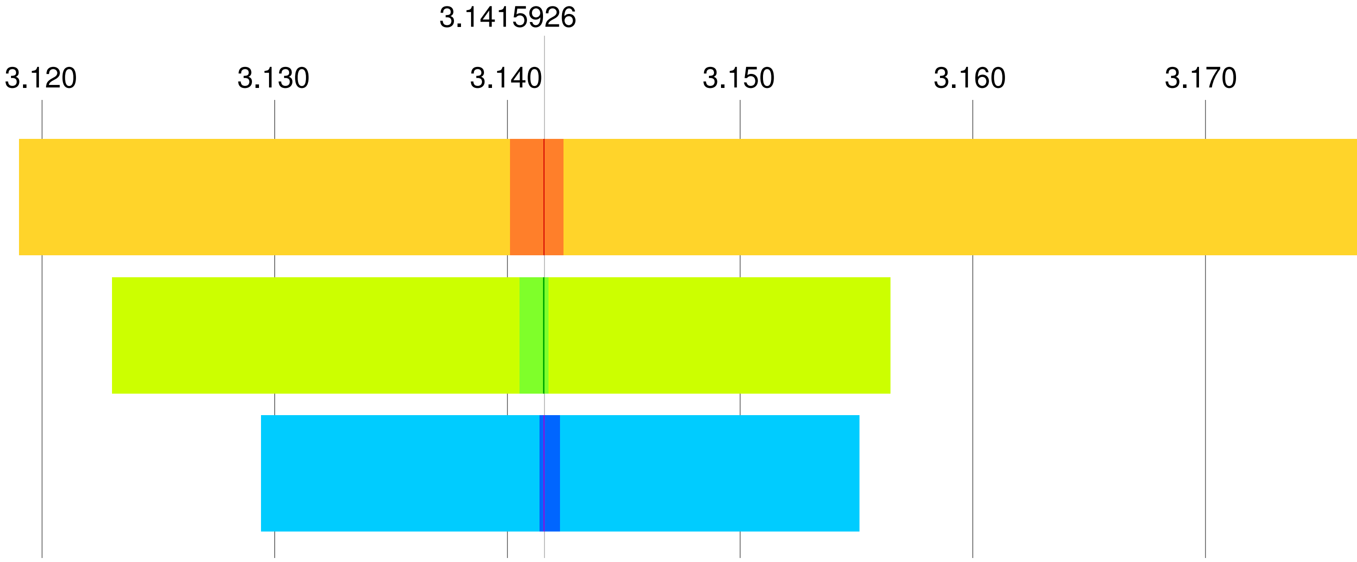

figure 4 - comparaison de l'ordre de grandeur des intervalles de confiance.

La figure ci-dessus reprends les résultats pour les intervalles de confiance à 95 % uniquement. L'axe orange reprends le premier tableau sur 10 simulations, le vert le deuxième pour 30, et le bleu le dernier pour 60 simulations. Chaque sous axe prends un nombre de lancé différent. Ainsi les rectangles les plus clairs sont ceux à 1 000 lancés, les intermédiaires à 1 000 000. Les simulations à 1 000 000 000 ne sont pas très comparables mais donnent tout de même l'ordre de grandeur (ne pas hésiter à zoomer sur la figure pour distinguer les rectangles les plus foncés).

On constate maintenant très bien que la simulation ne convergent pas linéairement par rapport au nombre de lancés.

Il semble se passer la même chose pour le nombre d'expériences. Par exemple, de orange à vert, on fait 3 fois plus d'expériences, et l'intervalle est réduit d'un peine la moitié.

4 - Conclusion

On a pu voir que la méthode de Monte-Carlo -un type de simulation stochastique- permet d'obtenir des résultats très fiables, mais au prix d'un grand nombre d'expériences et de lancés, et donc de temps. Cependant, il est beaucoup plus facile d'imaginer ce genre de simulation que de trouver les équations de ma simulation déterministe. Les intervalles de confiance permettent de bien visualiser la précision de nos résultats.