TP RE51 : Calcul distribué de la valeur de π

Christophe Dumez
UTBM

Table des matières

1 Introduction **2**

2 Méthode de Monte Carlo **2**
2.1 Présentation de la méthode 2
2.2 Présentation de l'algorithme 2
2.3 Travail demandé 3

3 Formule de Ramanujan **3**
3.1 Présentation de la formule 3
3.2 Travail demandé 3

1 Introduction

π est une constante mathématique représentant le ratio entre la circonférence de n'importe quel cercle par rapport à son diamètre. Dans ce TP, il vous est demandé de calculer la valeur de Pi de manière distribuée à l'aide de la méthode de Monte Carlo.

La programmation sera réalisée en C à l'aide la la bibliothèque OpenMPI.

2 Méthode de Monte Carlo

2.1 Présentation de la méthode

On appelle méthode de *Monte Carlo* toute méthode visant à calculer une valeur numérique, et utilisant des procédés aléatoires, c'est à dire des techniques probabilistes. Nous allons présenter dans cette partie une application de la méthode de Monte Carlo au calcul de π.

Supposons que vous ayez un carré, disons de côté 2. Son aire est donc de 4. A l'intérieur de ce carré, on inscrit un disque, de centre le centre du carré, et de rayon 1. L'aire de ce disque est π . Si l'on choisit au hasard un point du carré, la probabilité pour qu'il soit à l'intérieur du disque est donc de $\pi/4$. Voici comment nous allons procéder. Nous tirons au hasard un grand nombre de points du carré au hasard. On peut raisonnablement espérer que *(nb pts à l'intérieur du disque)/(nb pts tirés)* va nous donner une approximation de $\pi/4$.

2.2 Présentation de l'algorithme

Le listing 1 présente une implémentation en C de l'algorithme de Monte Carlo.

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <time.h>

int main(int argc, char** argv) {
   if(argc < 2) {
     printf("Usage: %s <nb_iter>\n", argv[0]);
     return 1;
   }
   int niter = atoi(argv[1]);

   srand(time(NULL));
   int count=0;
   int i;
   for(i=0; i<niter; ++i) {
      double x = (double)rand()/RAND_MAX;
      double y = (double)rand()/RAND_MAX;
      double z = x*x+y*y;
```

```
      if (z<=1) ++count; /* Dans le cercle */
   }
   double pi = (double)count/niter*4;
   printf("nb essais= %d , estimation de Pi %g \n",niter,pi);
   return 0;
}
```

Listing 1 – Algorithme pour la methode de Monte Carlo

2.3 Travail demandé

Le travail consiste à adapter le code fourni dans le listing 1 afin de le rendre exécutable dans un environnement distribué. Le code doit etre réalisé en C à l'aide de la librarie OpenMPI.

3 Formule de Ramanujan

3.1 Présentation de la formule

L'efficacité de la méthode de Monte Carlo n'est pas géniale et il faut beaucoup de tirages au sort pour obtenir plus que les deux premiers chiffres décimaux. Il existe heureusement des méthodes bien plus performantes pour calculer π. Parmi ces méthodes, la formule de Ramanujan est très utilisée.

La formule de Ramanujan permettant de déterminer la valeur de π est la suivante :

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!(1103 + 26390n)}{(n!)^4 396^{4n}}$$

3.2 Travail demandé

Le travail consiste à réaliser un programme en C, à l'aide de la librarie OpenMPI et utilisant la formule précédente pour estimer la valeur de Pi.