

# Programmation parallèle

UFR SAT, CFPP - MaDSI 1

Travaux pratiques 8

REPUBLIQUE DU SENEGAL  
MINISTRE DE L'ENSEIGNEMENT  
SUPERIEUR ET DE LA RECHERCHE



UNITÉ DE FORMATION ET DE RECHERCHE  
DE SCIENCES APPLIQUÉES ET DE  
TECHNOLOGIE  
Section Informatique

-----

Cet exercice présente comme sur le premier TP, un programme simple pour calculer la valeur de pi. La méthode évalue l'intégrale de  $4/(1+x^2)$  entre 0 et 1. La méthode est simple: l'intégrale est approximée par la somme de **n** intervalles; l'approximation de l'intégrale dans chaque intervalle est  $(1/n)*4/(1+x^2)$ . Le processus maître (rang 0) demande à l'utilisateur le nombre d'intervalles; le maître communique ce nombre à tous les autres processus. Chaque processus additionne ensuite chaque **i**ème intervalle (**x=rang/n, rang/n+taille/n, ...**). Enfin, les sommes calculées par chaque processus sont additionnées à l'aide d'une réduction avec la fonction **MPI\_Reduce**.

Voici les routines MPI utilisées dans ce TP :

**MPI\_Bcast MPI\_Reduce**

**Code : pi.c**

```
#include <stdio.h>
#include "mpi.h"
#include <math.h>

int main(argc,argv)
int argc;
char *argv[];
{
    int done = 0, n, myid, numprocs, i;
    double PI25DT = 3.141592653589793238462643;
    double mypi, pi, h, sum, x;

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    while (!done)
    {
        if (myid == 0) {
            printf("Entrer le nombre d'intervalles: (0 pour quitter)
");
            scanf("%d",&n);
        }
    }
```

```

MPI Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
if (n == 0) break;

h    = 1.0 / (double) n;
sum  = 0.0;
for (i = myid + 1; i <= n; i += numprocs) {
    x = h * ((double)i - 0.5);
    sum += 4.0 / (1.0 + x*x);
}
mypi = h * sum;

MPI Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);

if (myid == 0)
    printf("Valeur Pi = %.16f, Erreur = %.16f\n",pi, fabs(pi
- PI25DT));
}
MPI Finalize();
return 0;
}

```

## Sortie:

```

www:parallel babacardiop$ mplexed -np 5 ./pi
Entrer le nombre d'intervalles: (0 pour quitter) 0
www:parallel babacardiop$ mplexed -np 5 ./pi
Entrer le nombre d'intervalles: (0 pour quitter) 10
Valeur Pi = 3.1424259850010978, Erreur = 0.000033333314113047
Entrer le nombre d'intervalles: (0 pour quitter) 50
Valeur Pi = 3.1416259869230032, Erreur = 0.00003333333332101
Entrer le nombre d'intervalles: (0 pour quitter) 100
Valeur Pi = 3.1416009869231245, Erreur = 0.00000033333333314
Entrer le nombre d'intervalles: (0 pour quitter) 1000
Valeur Pi = 3.1415927369231271, Erreur = 0.00000000333333340
Entrer le nombre d'intervalles: (0 pour quitter) 10000
Valeur Pi = 3.1415926544231230, Erreur = 0.0000000000333298
Entrer le nombre d'intervalles: (0 pour quitter) 100000
Valeur Pi = 3.1415926535981336, Erreur = 0.00000000000003404
Entrer le nombre d'intervalles: (0 pour quitter) 1000000
Valeur Pi = 3.1415926535899197, Erreur = 0.000000000000001266
Entrer le nombre d'intervalles: (0 pour quitter) 10000000
Valeur Pi = 3.1415926535897145, Erreur = 0.00000000000000786
Entrer le nombre d'intervalles: (0 pour quitter) 0
www:parallel babacardiop$ █

```

## Questions:

1. Quelles sont les routines MPI utilisées dans ce TP ?
2. Que fait la fonction **MPI\_Reduce** ?
3. Quelle remarque faites-vous quant à la valeur de Pi et son erreur par rapport au nombre d'intervalles ?