

TD N°6 : Programmation parallèle avec MPI**Exercice 1**

Écrire un programme MPI dans lequel le processus 0 lit une donnée insérée par l'utilisateur et la transmet à tous les autres processus en l'envoyant dans un anneau. Autrement dit, le processus i devrait recevoir la donnée à partir du processus $i - 1$ et l'envoyer au processus $i + 1$, jusqu'à ce que le dernier processus soit atteint.

Exercice 2

Soit le code séquentiel présenté ci-dessous permettant le calcul du nombre π .

```
# include <stdio.h>
# include <stdlib.h>
# include <math.h>
# define PI 3.1415926535L

double f (double a) {
    return (double)4.0/(( double )1.0+(a*a));
}

int main () {
    int n,i;
    double h, pi, sum, x;

    for (;;) {
        printf ("Enter the number of intervals:(0 quits )");
        scanf ("%d", &n);
        if (n ==0) break ;
        h = ((double)1.0)/(double)n;
        sum = 0.0;
        for (i =1; i <=n; i++) {
            x = h*((double)i-(double)0.5);
            sum += f(x);
        }
        pi = h*sum ;
        printf ("PI is approximately: %.16f Error is: %.16f\n", pi , fabs(pi-PI));
    }

    return 0;
}
```

1. Modifier le programme ci-dessus afin de pouvoir le réaliser par deux processus en utilisant les fonctions MPI_Send () et MPI_Recv ().
2. Modifier le programme ci-dessus afin de pouvoir le réaliser par quatre processus en utilisant les fonctions MPI_Bcast () et MPI_Reduce () .

Exercice 3

On se propose de calculer un produit matrice-vecteur ($Y = A \times X$), où A est une matrice de taille 4×4 et X un vecteur de taille 4. Donner le programme MPI implémentant le produit matrice-vecteur selon les contraintes suivantes :

- Allouer et initialiser la matrice A et le vecteur Y par le processus 0.
- Chaque processus doit calculer un élément du vecteur Y .
- Regrouper, par le processus 0, tous les éléments calculés dans le vecteur Y .

Exercice 4

Donner le programme MPI permettant de calculer le produit de deux matrices de taille 8×8 par 8 processus.