Programmation Parallèle et Distribuée
TD 4

Exercice I: Produit scalaire

Soit la fonction :

```
double produit_scalaire( int N, double *a, double *b )
{
        double res = 0 ;
        for( int i = 0 ; i < N ; i++ )
                res += a[i] * b[i] ;
        return res ;
}
```

qui calcule le produit scalaire de deux vecteurs a et b de dimension N en séquentiel.

Paralléliser la fonction produit_scalaire dans le cas où a et b sont des vecteurs distribués sur tous les processus MPI (N devient alors le nombre d'éléments associés au processus MPI appelant produit_scalaire).

Exercice II : Collecte

Soit la fonction :

```
void gather( double in, double *out, int root ) ;
```

qui collecte dans le tableau out du processus de rang root toutes les valeurs in de tous les processus. On retrouve dans out[0] la valeur in du processus 0, dans out[1] la valeur in du processus 1, etc ...
Cette fonction doit être appelée simultanément par tous les processus MPI.

1. Quelle doit être la taille du tableau out
   - pour le processus root ?
   - pour les autres processus ?
2. Ecrire la fonction gather en utilisant des communications point-à-point MPI non bloquantes.

Exercice III : Réduction

Ecrire la fonction :

```
double reduction_somme( double in ) ;
```

qui retourne la somme de tous les `in` de tous les processus MPI :

- en utilisant des communications point à point bloquantes ;
- en utilisant des communications collectives autres que `MPI_Reduce` et `MPI_Allreduce`.

Tester cette fonction en reprenant l'exercice I.