

# Sujets de TD n° 2

## Exercice 1.

Ecrire un programme parallèle OpenMP qui réalise l'opération SAXPY, qui multiplie chaque élément d'un vecteur X de nombres flottants par une constante A puis lui ajoute l'élément de même rang du vecteur Y :  $S = A.X + Y$

## Exercice 2.

Paralléliser (avec OpenMP) le code suivant :

```
for (i=0 ; i<m ; i++){
    rowterm[i] = 0.0 ;
    for (j=0 ; j<p ; j++){
        rowterm[i] += a[i][2*j] * a[i][2*j+1] ;
    }
for (i=0 ; i<q ; i++){
    colterm[i] = 0.0 ;
    for (j=0 ; j<p ; j++){
        colterm[i] += b[2*j][i] * b[2*j+1][i] ;
    }
}
```

## Exercice 3.

Reprendre l'exercice 2 de la feuille précédente (simulation de l'évolution des températures dans une tige métallique) et en écrire une version parallèle en utilisant OpenMP.

On rappelle le programme séquentiel :

```
float temp[N] = {100.0, 0.0, 0.0, 0.0, ..., 0.0};
int t, s ;
/* calcul de d'évolution des températures */
for (t = 0 ; t < TMAX ; t++){
    for (s = 1 ; s < N-1 ; s++){
        temp[s] = (temp[s-1] + temp[s+1]) / 2.0;
    }
}
```

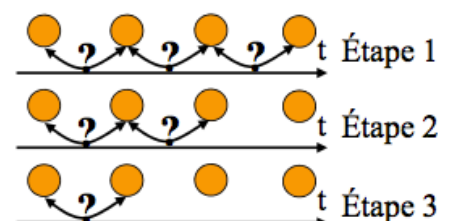
Pour la parallélisation, on admet que le calcul d'un segment à l'itération  $t$  se fasse avec les valeurs des segments voisins calculées à l'itération  $t$  ou  $t-1$ .

## Exercice 4.

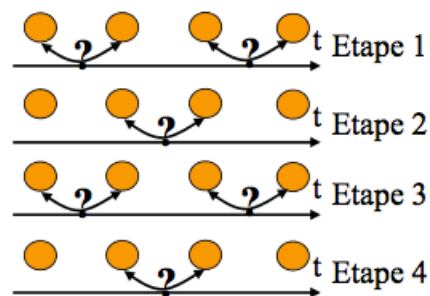
L'algorithme du tri-bulle compare toutes les paires d'éléments adjacents d'un vecteur et les échange si le premier est supérieur au deuxième :

### Question 1.

Est-ce que cet algorithme vous semble parallélisable ?  
Pourquoi ?



Une autre version, de même complexité, compare (a) dans un premier temps, les paires dont le premier élément est de rang pair, (b) dans un second temps, celle dont le premier élément est de rang impair :



### Question 2.

Ecrire un programme parallèle réalisant cet algorithme, en utilisant des directives OpenMP.

## Exercice 5.

On veut écrire un programme parallèle qui calcule la somme des éléments d'un vecteur. Pour éviter des conflits lors de l'accès à la somme globale, on choisit d'utiliser des sommes partielles : chaque thread calcule la somme des éléments qui lui sont attribués, puis on ajoute les sommes partielles.

### Question 1.

Ecrire un programme parallèle dans lequel la somme globale est calculée par un des threads.

### Question 2.

La figure ci-dessous illustre une autre manière de calculer la somme globale. Ecrire un programme parallèle qui réalise le calcul de cette manière-là.

