
TP n° 10
Assembleur + C

Exercice 1 - On s'intéresse au produit d'une matrice 4x4 de floats par un vecteur de 4 floats qui est une opération souvent utilisée en modélisation 3D. Une matrice 4x4 sera modélisée sous forme d'un tableau de 16 floats et un vecteur sous forme d'un tableau de 4 floats.

$$\begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & m_{0,3} \\ m_{1,0} & m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,0} & m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,0} & m_{3,1} & m_{3,2} & m_{3,3} \end{bmatrix} \times \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

```
// compute w = m * v
void prod(float *m, float *v, float *w) {
    int i,j;

    for (i=0;i<4;++i) {
        float s=0.0;
        for (j=0;j<4;++j) {
            s+=m[i*4+j]*v[j];
        }
        w[i]=s;
    }
}
```

On vous demande de comparer une version traditionnelle (voir code ci-dessus) traduite par le compilateur C avec deux versions SSE écrites en assembleur :

- dans la première version SSE vous réalisez le même produit que précédemment en utilisant les instructions `mulps` et `haddps`
- dans la seconde version, on considère que `haddps` n'existe pas (jeu d'instructions SSE2) et pour réaliser le calcul on transposera d'abord la matrice et on effectuera le calcul comme suit en utilisant `mulps`, `addps`, `movss` et `pshufd` :

$$\begin{bmatrix} m_{0,0} \times v_0 \\ m_{0,1} \times v_0 \\ m_{0,2} \times v_0 \\ m_{0,3} \times v_0 \end{bmatrix} + \begin{bmatrix} m_{1,0} \times v_1 \\ m_{1,1} \times v_1 \\ m_{1,2} \times v_1 \\ m_{1,3} \times v_1 \end{bmatrix} + \begin{bmatrix} m_{2,0} \times v_2 \\ m_{2,1} \times v_2 \\ m_{2,2} \times v_2 \\ m_{2,3} \times v_2 \end{bmatrix} + \begin{bmatrix} m_{3,0} \times v_3 \\ m_{3,1} \times v_3 \\ m_{3,2} \times v_3 \\ m_{3,3} \times v_3 \end{bmatrix}$$