

# Systèmes informatiques - LSINF1252

## Projet Matrix

Cabrera Alba Laritza  
noma : —

Chaque fichier contient les valeurs d'une matrice sous forme binaire et non textuelle. Voici une explication de ce format pour chacune des structures des matrices ("**sp\_matrix**" et "**matrix**"). Pour l'explication, quand on parle de bytes on considère que :

`sizeof(unsigned int) = 4 bytes`

`sizeof(double) = 8 bytes`

### 1 Format du file qui contient une matrice creuse "**sp\_matrix**"

- Les  $(4+4+8) = 16$  premiers bytes sont les données nécessaires pour l'information générale de la matrice
  - ◊ 4 bytes représentent le nombre de lignes ( $m$ ) qu'a la matrice.
  - ◊ 4 bytes suivants représentent le nombre de colonne ( $n$ ) qui a la matrice.
  - ◊ 8 bytes suivants représentent la précision de la matrice.
- Les  $(4+4+8) = 16$  bytes suivants sont les données nécessaires pour stocker le premier élément non nul dans la matrice
  - ◊ 4 bytes représentent la file( $i$ ) d'une valeur de la matrice.
  - ◊ 4 bytes représentent la colonne ( $j$ ) d'une valeur de la matrice.
  - ◊ 8 bytes représentent la valeur (value). Alors on sait que value est dans la position  $i, j$  de la matrice.

Les 16 bytes suivant (file, colonne, value) ont l'information du deuxième élément non zéro de la matrice et ainsi de suite jusqu'à la fin du fichier. Un fichier aura toujours une taille multiple de 16 bytes.

Si la Matrice a que des valeurs zéro alors, on stocke seulement, les dimensions  $m \times n$  et la précision de la matrice donc 16bytes.

Avec ce format on optimise de l'espace vu qu'on stocke " $2 * (4 \text{ bytes}) + (8 \text{ bytes})$ " pour chaque élément, ce qui est raisonnable pour une matrice creuse. Si la matrice à stocker n'est pas creuse il est conseillé d'utiliser la structure matrix.

### 2 Format du file qui contient une matrice non creuse "**matrix**"

- Les  $(4+4) = 8$  premiers bytes sont les données nécessaires pour l'information générale de la matrice
  - ◊ 4 bytes représentent le nombre de lignes ( $m$ ) qui a la matrice.
  - ◊ 4 bytes suivants représentent le nombre de colonne ( $n$ ) qui a la matrice.

- Les 8 bytes suivants stockent le premier élément (de la première ligne  $\times$  la première colonne) de la matrice. Il est facile de calculer la position de chaque élément à partir de la dimension de la matrice.

Les 8 bytes suivants représentent le deuxième élément (de ligne  $\times$  colonne correspondante) de la matrice et ainsi de suite jusqu'à la fin du fichier. Un fichier aura toujours une taille multiple de 8 bytes.

Ce format est avantageux pour les matrices non creuses parce qu'on stocke tous les éléments, sans l'information de la (ligne  $\times$  colonne) correspondante, comme dis précédemment en sachant les dimensions de la matrice on peut facilement calculer la position ( $i \times j$ ) à partir de l'information générale. Si vous allez utiliser de matrices non creuses cet format est conseillé.