

Atelier 6

Objectifs de formation

- Inverser une matrice sur papier et sur ordinateur
- Résoudre des systèmes linéaires en parallèle (systèmes matricielles) et itératifs (décomposition LU)
- Écrire une mini bibliothèque d'algèbre linéaire et l'utiliser sur des problèmes simples

1. : 90 mn

- Écrire une fonction `void invmat(float *a, int n)` qui inverse la matrice A en renvoyant l'inverse dans A elle-même en utilisant l'algorithme de Gauss-Jordan en utilisant l'algorithme vu en cours.
- Tester la fonction sur la matrice

$$\begin{pmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{pmatrix}$$

- Optimiser la fonction avec des pointeurs

2. 90 mn donnant lieu à un rendu

On veut calculer l'inverse d'une matrice en commençant par une décomposition LU telle que $A = LU$. Puis on calculera L^{-1} et U^{-1} directement en tenant compte du fait que ce sont des matrices triangulaires. Au final, on obtiendra $A^{-1} = U^{-1} \times L^{-1}$.

- Écrire une fonction `void decomplu(float *a, int n)` qui décompose la matrice A en renvoyant L et U dans A elle-même en utilisant l'algorithme vu en cours.

Tester la fonction sur la matrice

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & -2 & 4 & -8 & 16 \\ 1 & -3 & 9 & -27 & 81 \end{pmatrix}$$

en utilisant la fonction

```
void propmatmat(float *a, float *b, float *res, int n)
```

vu dans l'atelier 3.

- Écrire une fonction `void invlinf(float *l, int n)` qui inverse une matrice L triangulaire inférieure en renvoyant L^{-1} dans L elle-même.

Tester la fonction sur la matrice L obtenu dans la décomposition précédente.

- Ecrire une fonction `void invusup(float *u, int n)` qui inverse une matrice U triangulaire supérieure avec des 1 sur la diagonale en renvoyant U^{-1} dans U elle-même.

Tester la fonction sur la matrice U obtenu dans la décomposition précédente.

- Calculer l'inverse d'une matrice A en utilisant les fonctions précédentes.

Tester votre programme sur la matrice

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & -2 & 4 & -8 & 16 \\ 1 & -3 & 9 & -27 & 81 \end{pmatrix}$$