L'algèbre linéaire et l'ordinateur

Les vecteurs et l'ordinateur

Notion de Vecteurs

L'ensemble des n-uplets à coefficients dans $\mathbb{R}$, noté $\mathbb{R}^n$, est un espace vectoriel de dimension n.

Un élément v de $\mathbb{R}^n$, noté $\begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{pmatrix}$, est appelé un vecteur.

Opérations sur les Vecteurs

Addition : Soient $v \in \mathbb{R}^n$ et $w \in \mathbb{R}^n$,

$$v + w = \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 + w_1 \\ v_2 + w_2 \\ \dots \\ v_n + w_n \end{pmatrix} = s.$$

Produit par un scalaire : Soient $\lambda \in \mathbb{R}$ et $v \in \mathbb{R}^n$,

$$\lambda \times v = \lambda \times \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{pmatrix} = \begin{pmatrix} \lambda v_1 \\ \lambda v_2 \\ \dots \\ \lambda v_n \end{pmatrix} = p.$$

Combinaison linéaire : Soient $\lambda_i \in \mathbb{R}$ et $V_i \in \mathbb{R}^n$, $\sum^{n} \lambda_i V_i$

Le produit scalaire

Soient $v \in \mathbb{R}^n$ et $w \in \mathbb{R}^n$,

$$v \cdot w = \begin{pmatrix} v_1 & v_2 & \dots & v_n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{pmatrix} = \sum_{i=1}^{n} v_i \times w_i \in \mathbb{R}.$$

Produit Scalaire : Algorithme

ProdScal(v, w, n)

Entrées $v = \begin{pmatrix} v_1 & v_2 & \dots & v_n \end{pmatrix}$ et
$w = \begin{pmatrix} w_1 & w_2 & \dots & w_n \end{pmatrix}$ deux vecteurs.

Sortie $\lambda \in \mathbb{R}$ avec $\lambda = v \cdot w$

Corps $\lambda \leftarrow 0$

Pour $i = 1$ à n faire

$\lambda \leftarrow \lambda + v_i \times w_i$

Rappel sur les pointeurs

```
uint32_t a;
float v[10];
```

Une déclaration de variable entraine la déclaration implicite de l'adresse, notée $\&a$, de la variable et la réservation mémoire, i.e. le nombre d'octets nécessaires, pour stocker son contenu à l'adresse fournie par le compilateur. Pour a, 4 octets.

Dans le cas d'un tableau, v **est** l'adresse du tableau, i.e. l'adresse du premier élément du tableau. Pour stocker 10 flottants en simple précision, le compilateur réservera 40 octets.

Deux façons de parcourir un tableau en C

```
int i;
float v[10], w[10], pscla = 0;
for(i=0; i<n; i++)
   pscal += v[i]*w[i];
```

et on calcule $v + 4 * i$ et $w + 4 * i$ à chaque itération

ou bien

```
float v[10], w[10], pscal = 0, *p1, *p2;
for(p1 = v, p2 = w; p1 < v + n; p1++, p2++)
    pscal += *p1 * *p2;
```

et il n'y a aucun calcul d'adresse.

L'allocation dynamique de mémoire

```
#include <stdlib.h> // A ne pas oublier

void lagrange(double(*ff)(double), double *p, double *x, int n))
{
  double *v, *b;
    ..........
  // besoin d'une matrice de dimension n+1
  // et d'un vecteur de taille n
  v = (double *) malloc (sizeof(double)*(n+1)*(n+1));
  b = (double *) malloc (sizeof(double)*(n+1));
  if(v == NULL || b == NULL)
  {
    perror("Echec allocation memoire");
    exit(1);
  }
    ..........
  free(v);
}
```

Fin

Les matrices et l'ordinateur

Notion de matrice

Soient n vecteurs $V_i = \begin{pmatrix} v_{1,i} \\ v_{2,i} \\ \dots \\ v_{n,i} \end{pmatrix}$.

L'ensemble $A = (V_1, ..., V_n)$ est appelé matrice carrée d'ordre n et noté

$$A = \begin{pmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,n} \\ v_{2,1} & v_{2,2} & \dots & v_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & \dots & v_{n,n} \end{pmatrix}$$

La transposée

Si $A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}$,

la transposée de A est définie par

$$A^T = V = \begin{pmatrix} a_{1,1} & a_{2,1} & \dots & a_{n,1} \\ a_{1,2} & a_{2,2} & \dots & a_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1,n} & a_{2,n} & \dots & a_{n,n} \end{pmatrix}$$

On inverse i et j.

L'addition

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} + \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix}$$

$$= \begin{pmatrix} a_{1,1} + b_{1,1} & a_{1,2} + b_{1,2} & \dots & a_{1,n} + b_{1,n} \\ a_{2,1} + b_{2,1} & a_{2,2} + b_{2,2} & \dots & a_{2,n} + b_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} + b_{n,1} & a_{n,2} + b_{n,2} & \dots & a_{n,n} + b_{n,n} \end{pmatrix}$$

Le nombre d'opération est n^2.

Produit par un scalaire

$$\lambda \cdot \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} = \begin{pmatrix} \lambda a_{1,1} & \lambda a_{1,2} & \dots & \lambda a_{1,n} \\ \lambda a_{2,1} & \lambda a_{2,2} & \dots & \lambda a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda a_{n,1} & \lambda a_{n,2} & \dots & \lambda a_{n,n} \end{pmatrix}.$$

Le nombre d'opération est n^2.

Le produit Matrice Vecteur

Soit $A = (a_{i,j})$ une matrice carrée d'ordre n et $V = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$ un vecteur d'ordre n.

Le produit matrice-vecteur est défini par

$$A \times V = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} a_{1,i} v_i \\ \sum_{i=1}^{n} a_{2,i} v_i \\ \vdots \\ \sum_{i=1}^{n} a_{n,i} v_i \end{pmatrix}.$$

Le nombre d'opérations est $2 \times n^2$

Multiplication de matrices

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \times \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix}$$

$$= \begin{pmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,n} \\ p_{2,1} & p_{2,2} & \dots & p_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n,1} & p_{n,2} & \dots & p_{n,n} \end{pmatrix}.$$

avec $p_{i,j} = \begin{pmatrix} a_{i,1} & a_{i,2} & \dots & a_{i,n} \end{pmatrix} \cdot \begin{pmatrix} b_{1,j} \\ b_{2,j} \\ \vdots \\ b_{n,j} \end{pmatrix} = \sum_{k=1}^{n} a_{i,k} b_{k,j}$.

On remarquera le principe *des dominos*.

Le nombre d'opération est $2 \times n^3$.

Deux déclarations possibles

```
float v[N][N];
```

et on accède aux éléments de la matrice par $v[i][j]$

ou bien

```
float v[N*N];
```

et on accède aux éléments de la matrice par $*(v + i * N + j)$.

Le produit matrice-vecteur en C

```
#include <stdio.h>
#include <stdlib.h>

#define N 20

void prodmatvec(float **a, float *x, float *res, int N)
{
  int i, j;
  for(i=0; i<N; i++)
    {
      res[i]= 0;
      for(j=0; j<N; j++)
        res[i] += a[i][j]*x[j];
    }
}
```

Le produit matrice-vecteur en C optimisé

```
#include <stdio.h>
#include <stdlib.h>

#define N 20

void prodmatvec(float *a, float *x, float *res, int N)
{
  float *p1, *p2, *p3;
  for(p1=a, p3=res ; p3<res+N; p3++)
    {
      *p3 = 0;
      for(p2=x; p2<x+N; p1++, p2++)
        *p3 += (*p1) * (*p2);
    }
}
```

Fin

Trois transformations matricielles

Un exemple

Si $\forall A \in \mathcal{M}_n(\mathbb{R})$ la ligne i_0 de A reste inchangée dans le produit PA, alors la ligne i_0 de P est celle de l'identité I_d.

En effet, soit $A = I_d$, si la ligne i_0 de I_d reste inchangée dans le produit $P \times I_d = P$ alors la ligne i_0 de P est celle de l'identité I_d.

Réciproquement si la ligne i_0 de P est celle de l'identité I_d, alors, soit $A = (a_{i,j})$ le calcul de la ligne i_0 de PA donne

$$\forall 1 \leq j \leq n, (PA)_{i_0,j} = \sum_{k=1}^{n} p_{i_0,k} a_{k,j} = a_{i_0,j}$$

Transformation de type 1

On échange les lignes k et l de la matrice $A = (a_{i,j}), \forall A$.

Cela revient à multiplier A à gauche par

$$P^1_{k,l} = \begin{array}{c} \\ \vdots \\ k \\ \vdots \\ l \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccc} \dots & k & \dots & l & \dots & & \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \end{array}$$

P^1 est obtenue à partir de l'identité en échangeant les lignes k et l.

Le déterminant de P^1 vaut -1 et $P^1 \times P^1 = I_d$.

Si P existe alors P est la matrice I_t obtenue où l'on a effectué la transformation sur l'identité.

$$\left\{ \begin{array}{rcl} P \times I_d & = & I_t \\ P \times I_d & = & P \end{array} \right. \Rightarrow P = I_t = P^1_{k,l}$$

Inversement, calculons $P^1_{k,l} \times A$

$$P^1_{k,l} \times A = \begin{array}{c} \\ \vdots \\ k \\ \vdots \\ l \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccccccc} & \ldots & k & \ldots & l & \ldots \end{array} \\ \left(\begin{array}{cccccccc} & & & & & & \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \hline & & & a_{i,j} & & & \\ \hline \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \hline & & & a_{i,j} & & & \\ \hline \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ & & & a_{i,j} & & & \\ & & & & & & \end{array} \right) \end{array}$$

Pour la ligne k (idem pour la ligne l) :

$$\forall 1 \leq j \leq n, (PA)_{k,j} = \sum_{i=1}^{n} \delta_{l,i} a_{i,j} = a_{l,j}$$

Transformation de type 2

On multiplie la ligne k de A par λ.

Cela revient à multiplier A à gauche par

$$P_k^2 = \begin{array}{c} \\ \vdots \\ k \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} & & k & & \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \lambda & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \end{array}$$

Le déterminant de P^2 vaut λ et son inverse est P^2 en changeant λ par $1/\lambda$.

Si P existe alors P est la matrice I_λ obtenue où l'on a effectué la transformation sur l'identité.

$$\left\{ \begin{array}{rcl} P \times I_d & = & I_\lambda \\ P \times I_d & = & P \end{array} \right. \Rightarrow P = I_\lambda = P_k^2$$

Inversement, calculons $P_k^2 \times A$

$$P_k^2 \times A = \begin{array}{c} \\ \\ k \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & \cdots & & \cdots & & \cdots \end{array} \\ \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & a_{i,j} & & & & & \\ \hline \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ & & & a_{i,j} & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & a_{i,j} & \\ \end{pmatrix} \end{array}$$

Pour la ligne k :

$$\forall 1 \leq j \leq n, (PA)_{k,j} = \sum_{i=1}^{n} \lambda \times \delta_{k,i} \times a_{i,j} = \lambda \times a_{k,j}$$

Transformation de type 3

On soustrait la ligne l multipliée par λ à la ligne k de A.

Cela revient à multiplier A à gauche par

$$P^3_{k,l} = \begin{array}{c} \\ \vdots \\ k \\ \vdots \\ l \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccc} \ldots & k & \ldots & l & \ldots & & \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 1 & \ldots & -\lambda & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix} \end{array}$$

Le déterminant de P^3 vaut 1 et son inverse est P^3 en changeant λ par $-\lambda$.

Si P existe alors P est la matrice Q obtenue où l'on a effectué la transformation sur l'identité.

$$\begin{cases} P \times I_d &= Q \\ P \times I_d &= P \end{cases} \Rightarrow P = Q = P^3_{k,l}$$

Inversement, calculons $P^3_{k,l} \times A$

$$P^3_{k,l} \times A = \begin{array}{c} \\ \\ k \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & \dots & k & \dots & l & \dots & \end{array} \\ \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & a_{i,j} & & & & & \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & a_{i,j} & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & a_{i,j} & \vdots \end{pmatrix} \end{array}$$

Pour la ligne k :

$$\forall 1 \leq j \leq n, (PA)_{k,j} = \sum_{i=1}^{n} p_{k,i} a_{i,j} = a_{k,j} - \lambda \times a_{l,j}$$

cqfd.

Fin