Calcul Vectoriel et Matriciel
Méthodes Numériques

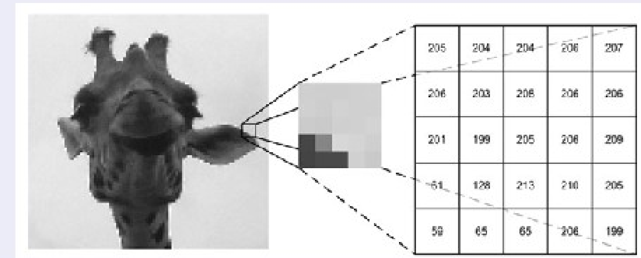
J-F. Méhaut
UGA – CEA
Polytech Grenoble

Plan

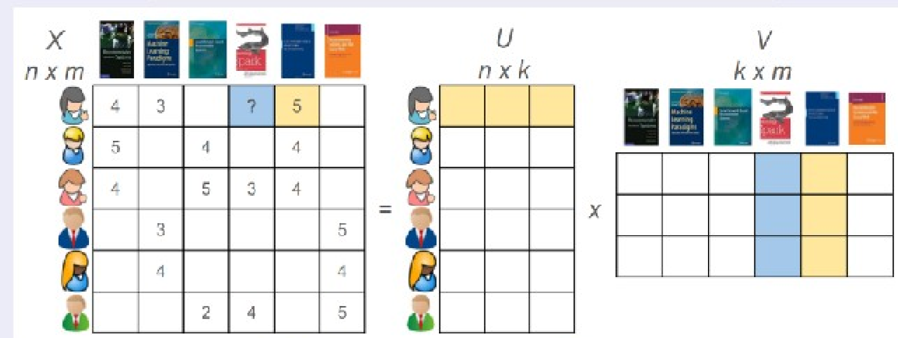
Utilisation de Matrices I

Informatique

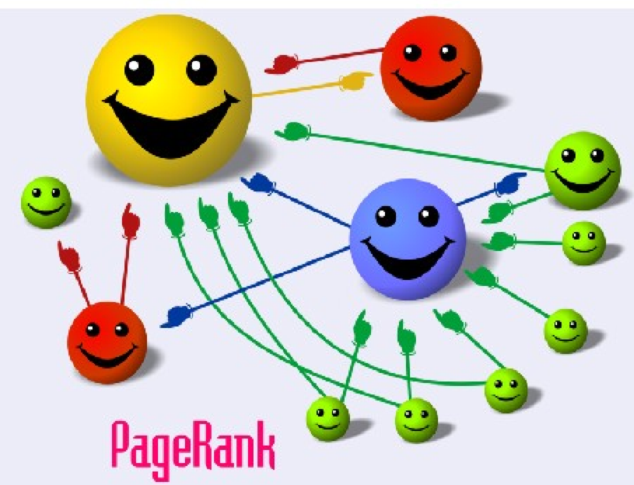
- Traitement d'Images



- Système de recommandations : Netflix, Amazon



- Algorithme PageRank de Google



Utilisation de Matrices II

Mathématiques

- Linéaire = simple

$$A\mathbf{x} = \mathbf{b}$$

- Une matrice A est associée à une application linéaire.
- Exemple 2 :

$$\left\{\begin{array}{ccccc} 4\mathbf{x}_1 & + & \mathbf{x}_2 & + & 2\mathbf{x}_3 = 8 \\ 2\mathbf{x}_1 & & & + & 4\mathbf{x}_3 = 2 \\ \mathbf{x}_1 & + & 5\mathbf{x}_2 & + & 4\mathbf{x}_3 = 1 \end{array}\right. \Leftrightarrow \begin{pmatrix} 4 & 1 & 2 \\ 2 & 0 & 4 \\ 1 & 5 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \\ 1 \end{pmatrix}$$

Utilisation de Matrices III

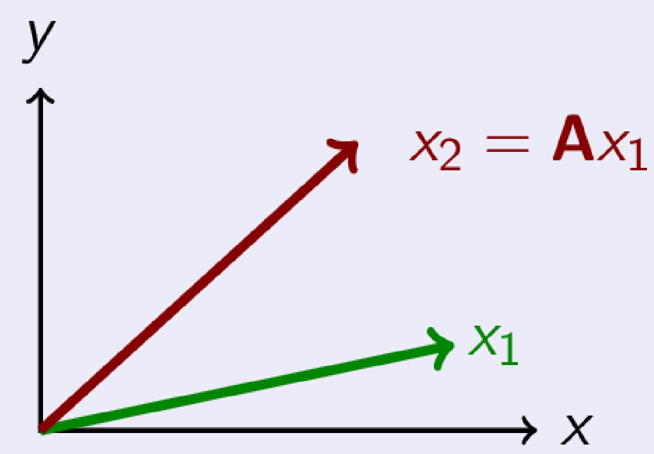
Mathématiques II

- Linéaire = simple

$$A\mathbf{x} = \mathbf{b}$$

- Une matrice A est associée à une application linéaire
- Exemple 1 : $\mathbf{A}$ matrice de rotation

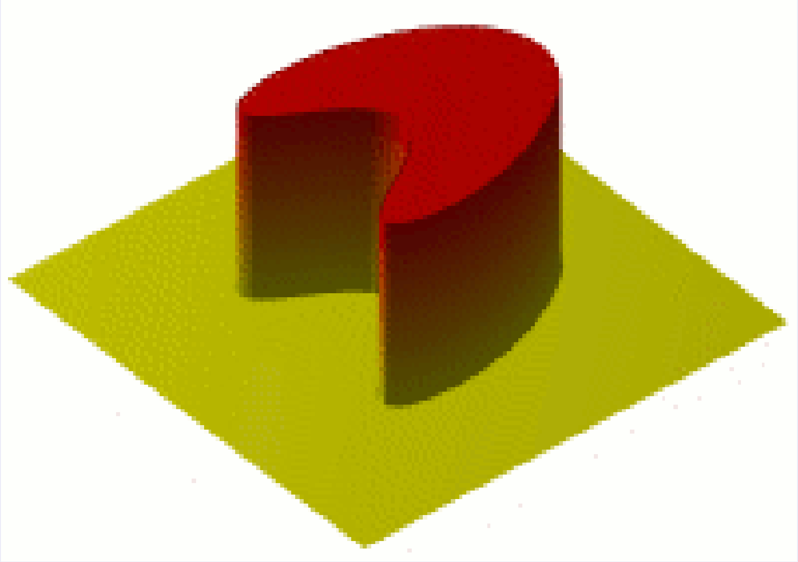
$$\mathbf{A} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$



Utilisation de Matrices IV

Matrice et physique

- Equation de la chaleur
- Circuit électrique
- Matrices aléatoires : dans les années 1950, Wigner a suggéré de remplacer l'opérateur hamiltonien du noyau par une matrice aléatoire (cf wikipedia)
  - n équations à p inconnues



Notations et définitions

Définition

- Matrice A
- $A \in \mathbb{R}^{m \times n}$, $A = \begin{pmatrix} a_{1,1} & \dots & a_{1,p} \\ \vdots & & \vdots \\ a_{n,1} & \dots & a_{n,p} \end{pmatrix}$
- n lignes
- p colonnes (si $p = 1$, c'est un vecteur)
- a_{ij} : terme général

Notations

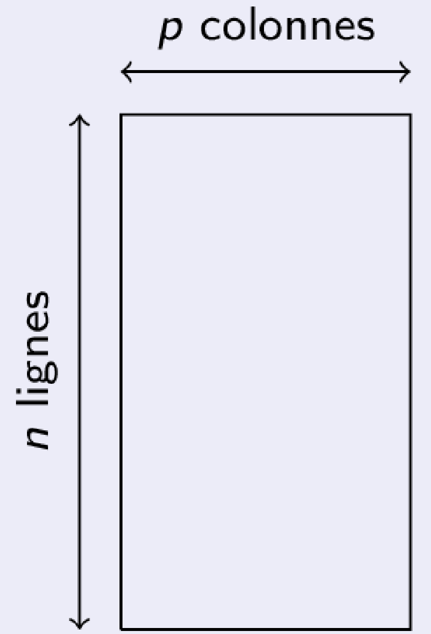
- Vecteur : $\mathbf{v} \in \mathbb{R}^n$;
- Transposition $\mathbf{v}^\top = (v_1, ..., v_i, ..., v_n)$
- Matrice $A \in \mathbb{R}^{n \times p}$ de terme général a_{ij}
- vecteur colonne extrait de A :

$$A(:, j) = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{pmatrix}$$

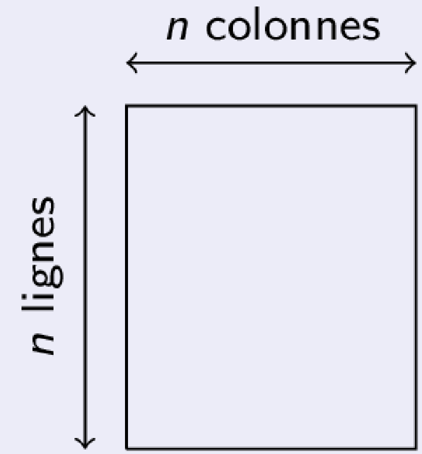
- Vecteur ligne extrait de A : $A(i, :) = (a_{i1}, ..., a_{ij}, \ldots, a_{ip})$,
- NB : cette notation est la même dans Matlab.

Différentes tailles de matrices :

Rectangulaire, $n \neq p$



Carrée, $n = p$



Type de Matrices II

Différentes structures :

- Pleine, $a_{i,j} \neq 0, \forall i, j$
- Creuse, $a_{i,j} = 0$, pour beaucoup d'indices.
- Triangulaire : $\begin{pmatrix} a_{1,1} & \dots & a_{1,p} \\ 0 & \ddots & \vdots \\ 0 & 0 & a_{n,p} \end{pmatrix}$
- Diagonale : $\begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_n \end{pmatrix}$
- par blocs : $\begin{pmatrix} a_{11} & \cdots & a_{1n} & 0 & \cdots & 0 \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ a_{m1} & \cdots & a_{mn} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_{11} & \cdots & b_{1q} \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 0 & \cdots & 0 & b_{p1} & \cdots & b_{pq} \end{pmatrix}$

Opérations vectorielles I

Matrice et vecteur

- $\mathbf{v} \in \mathbb{R}^{1\times n}$: Vecteur ligne
- $\mathbf{u} \in \mathbb{R}^{n\times 1}$: Vecteur colonne (vecteur par défaut)

$\rightarrow$ Cas particuliers de matrices

Opérations I

- Transposition : *vecteur ligne* $\leftrightarrow$ *vecteur colonne*

$$\mathbf{u} = \mathbf{v}^{\mathsf{T}}$$

- Addition :

$$\mathbf{w} = \mathbf{v} + \mathbf{x} \Leftrightarrow w_i = v_i + x_i, \forall i = 1, \ldots, n$$

- Multiplication par un scalaire :

$$\mathbf{w} = \alpha \mathbf{v} \Leftrightarrow w_i = \alpha v_i$$

Opérations vectorielles II

Opérations II

- Produit scalaire :

$$w = \mathbf{v}^\top \mathbf{x} \Leftrightarrow w = \sum_{i=1}^{n} v_i x_i$$

- Produit extérieur (*outer product*)

$$C = \mathbf{x}\mathbf{v}^\top \Leftrightarrow c_{ij} = x_i v_j$$

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} * \begin{pmatrix} v_1 & v_2 & \dots & v_p \end{pmatrix} = \begin{pmatrix} x_1 v_1 & x_1 v_2 & \cdots & x_1 v_p \\ x_2 v_1 & x_2 v_2 & \cdots & x_2 v_p \\ \vdots & \vdots & \ddots & \vdots \\ x_n v_1 & x_n v_2 & \cdots & x_n v_p \end{pmatrix}$$

BLAS

- Basic Linear Algebra Library Subprograms

- The BLAS functionality is divided into three levels:
  - **Level 1:** contains *vector operations* of the form:

    $$\mathbf{y} \leftarrow \alpha \mathbf{x} + \mathbf{y}$$

    as well as scalar *dot products* and *vector norms*

  - **Level 2:** contains *matrix-vector operations* of the form

    $$\mathbf{y} \leftarrow \alpha A \mathbf{x} + \beta \mathbf{y}$$

    as well as *Tx* = *y* solving for **x** with *T* being triangular

  - **Level 3:** contains *matrix-matrix operations* of the form

    $$C \leftarrow \alpha AB + \beta C$$

    as well as solving $B \leftarrow \alpha T^{-1} B$ for triangular matrices *T*. This level contains the widely used *General Matrix Multiply* operation.

BLAS

- Several implementations for different languages exist
  - Reference implementation (F77 and C)
  - ATLAS, highly optimized for particular processor architectures
  - A generic C++ template class library providing BLAS functionality: uBLAS
  - Several vendors provide libraries optimized for their architecture (AMD, HP, IBM, Intel, NEC, NViDIA, Sun)

BLAS: F77 naming conventions

- Each routine has a name which specifies the operation, the type of matrices involved and their precisions.
  Names are in the form: PMMOO
  - Some of the most common operations (OO):
    - **DOT** scalar product, x^T y
      AXPY vector sum, α x + y
      MV matrix-vector product, A x
      SV matrix-vector solve, inv(A) x
      MM matrix-matrix product, A B
      SM matrix-matrix solve, inv(A) B
  - The types of matrices are (MM)
    - **GE** general
      GB general band
      SY symmetric
      SB symmetric band
      SP symmetric packed
      HE hermitian
      HB hermitian band
      HP hermitian packed
      TR triangular
      TB triangular band
      TP triangular packed
  - Each operation is defined for four precisions (P)
    - **S** single real
      D double real
      C single complex
      Z double complex
- Examples

  SGEMM stands for “single-precision general matrix-matrix multiply”

  DGEMM stands for “double-precision matrix-matrix multiply”.

BLAS Level 1 routines

- Vector operations (xROT, xSWAP, xCOPY etc.)
- Scalar dot products (xDOT etc.)
- Vector norms (IxAMX etc.)

Level 1 BLAS

	dim	scalar	vector	vector	scalars	5-element array	
SUBROUTINE xROTG (					A, B, C, S)		Generate plane rotation
SUBROUTINE xROTMG(					D1, D2, A, B,	PARAM)	Generate modified plane rotation
SUBROUTINE xROT (	N,		X, INCX,	Y, INCY,	C, S)		Apply plane rotation
SUBROUTINE xROTM (	N,		X, INCX,	Y, INCY,		PARAM)	Apply modified plane rotation
SUBROUTINE xSWAP (	N,		X, INCX,	Y, INCY)			$x \leftrightarrow y$
SUBROUTINE xSCAL (	N,	ALPHA,	X, INCX)				$x \leftarrow \alpha x$
SUBROUTINE xCOPY (	N,		X, INCX,	Y, INCY)			$y \leftarrow x$
SUBROUTINE xAXPY (	N,	ALPHA,	X, INCX,	Y, INCY)			$y \leftarrow \alpha x + y$
FUNCTION xDOT (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^T y$
FUNCTION xDOTU (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^T y$
FUNCTION xDOTC (	N,		X, INCX,	Y, INCY)			$dot \leftarrow x^H y$
FUNCTION xxDOT (	N,		X, INCX,	Y, INCY)			$dot \leftarrow \alpha + x^T y$
FUNCTION xNRM2 (	N,		X, INCX)				$nrm2 \leftarrow \|\|x\|\|_2$
FUNCTION xASUM (	N,		X, INCX)				$asum \leftarrow \|\|re(x)\|\|_1 + \|\|im(x)\|\|_1$
FUNCTION IxAMAX(	N,		X, INCX)				$amax \leftarrow 1^{st} k \ni \|re(x_k)\| + \|im(x_k)\| = max(\|re(x_i)\| + \|im(x_i)\|)$

BLAS Level 2 routines

- Matrix-vector operations (xGEMV, xGBMV, xHEMV, xHBMV etc.)
- Solving *Tx* = *y* for x, where T is triangular (xGER, xHER etc.)

Level 2 BLAS

	options	dim	b-width	scalar	matrix	vector	scalar	vector	
xGEMV (	TRANS,	M, N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y, y \leftarrow \alpha A^T x + \beta y, y \leftarrow \alpha A^H x + \beta y, A - m \times n$
xGBMV (	TRANS,	M, N,	KL, KU,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y, y \leftarrow \alpha A^T x + \beta y, y \leftarrow \alpha A^H x + \beta y, A - m \times n$
xHEMV (	UPLO,	N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xHBMV (	UPLO,	N,	K,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xHPMV (	UPLO,	N,		ALPHA,	AP,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xSYMV (	UPLO,	N,		ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xSBMV (	UPLO,	N,	K,	ALPHA,	A, LDA,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xSPMV (	UPLO,	N,		ALPHA,	AP,	X, INCX,	BETA,	Y, INCY)	$y \leftarrow \alpha Ax + \beta y$
xTRMV (	UPLO, TRANS, DIAG,	N,			A, LDA,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$
xTBMV (	UPLO, TRANS, DIAG,	N,	K,		A, LDA,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$
xTPMV (	UPLO, TRANS, DIAG,	N,			AP,	X, INCX)			$x \leftarrow Ax, x \leftarrow A^T x, x \leftarrow A^H x$
xTRSV (	UPLO, TRANS, DIAG,	N,			A, LDA,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$
xTBSV (	UPLO, TRANS, DIAG,	N,	K,		A, LDA,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$
xTPSV (	UPLO, TRANS, DIAG,	N,			AP,	X, INCX)			$x \leftarrow A^{-1}x, x \leftarrow A^{-T}x, x \leftarrow A^{-H}x$

	options	dim	scalar	vector	vector	matrix	
xGER (		M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)	$A \leftarrow \alpha xy^T + A, A - m \times n$
xGERU (		M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)	$A \leftarrow \alpha xy^T + A, A - m \times n$
xGERC (		M, N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)	$A \leftarrow \alpha xy^H + A, A - m \times n$
xHER (	UPLO,	N,	ALPHA,	X, INCX,		A, LDA)	$A \leftarrow \alpha xx^H + A$
xHPR (	UPLO,	N,	ALPHA,	X, INCX,		AP)	$A \leftarrow \alpha xx^H + A$
xHER2 (	UPLO,	N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)	$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$
xHPR2 (	UPLO,	N,	ALPHA,	X, INCX,	Y, INCY,	AP)	$A \leftarrow \alpha xy^H + y(\alpha x)^H + A$
xSYR (	UPLO,	N,	ALPHA,	X, INCX,		A, LDA)	$A \leftarrow \alpha xx^T + A$
xSPR (	UPLO,	N,	ALPHA,	X, INCX,		AP)	$A \leftarrow \alpha xx^T + A$
xSYR2 (	UPLO,	N,	ALPHA,	X, INCX,	Y, INCY,	A, LDA)	$A \leftarrow \alpha xy^T + \alpha yx^T + A$
xSPR2 (	UPLO,	N,	ALPHA,	X, INCX,	Y, INCY,	AP)	$A \leftarrow \alpha xy^T + \alpha yx^T + A$

BLAS Level 3 routines

- Matrix-matrix operations (xGEMM etc.)
- Solving $B \leftarrow \alpha T^{-1} B$ for triangular matrices (xTRMM)
- Widely used matrix-matrix multiply (xSYMM, xGEMM)

Level 3 BLAS

	options	dim	scalar	matrix	matrix	scalar	matrix	
xGEMM (	TRANSA, TRANSB,	M, N, K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha op(A)op(B) + \beta C, op(X) = X, X^T, X^H, C - m \times n$
xSYMM (	SIDE, UPLO,	M, N,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C - m \times n, A = A^T$
xHEMM (	SIDE, UPLO,	M, N,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB + \beta C, C \leftarrow \alpha BA + \beta C, C - m \times n, A = A^H$
xSYRK (	UPLO, TRANS,	N, K,	ALPHA,	A, LDA,		BETA,	C, LDC)	$C \leftarrow \alpha AA^T + \beta C, C \leftarrow \alpha A^T A + \beta C, C - n \times n$
xHERK (	UPLO, TRANS,	N, K,	ALPHA,	A, LDA,		BETA,	C, LDC)	$C \leftarrow \alpha AA^H + \beta C, C \leftarrow \alpha A^H A + \beta C, C - n \times n$
xSYR2K(	UPLO, TRANS,	N, K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB^T + \bar{\alpha} BA^T + \beta C, C \leftarrow \alpha A^T B + \bar{\alpha} B^T A + \beta C, C - n \times n$
xHER2K(	UPLO, TRANS,	N, K,	ALPHA,	A, LDA,	B, LDB,	BETA,	C, LDC)	$C \leftarrow \alpha AB^H + \bar{\alpha} BA^H + \beta C, C \leftarrow \alpha A^H B + \bar{\alpha} B^H A + \beta C, C - n \times n$
xTRMM (	SIDE, UPLO, TRANSA, DIAG,	M, N,	ALPHA,	A, LDA,	B, LDB)			$B \leftarrow \alpha op(A)B, B \leftarrow \alpha B op(A), op(A) = A, A^T, A^H, B - m \times n$
xTRSM (	SIDE, UPLO, TRANSA, DIAG,	M, N,	ALPHA,	A, LDA,	B, LDB)			$B \leftarrow \alpha op(A^{-1})B, B \leftarrow \alpha B op(A^{-1}), op(A) = A, A^T, A^H, B - m \times n$

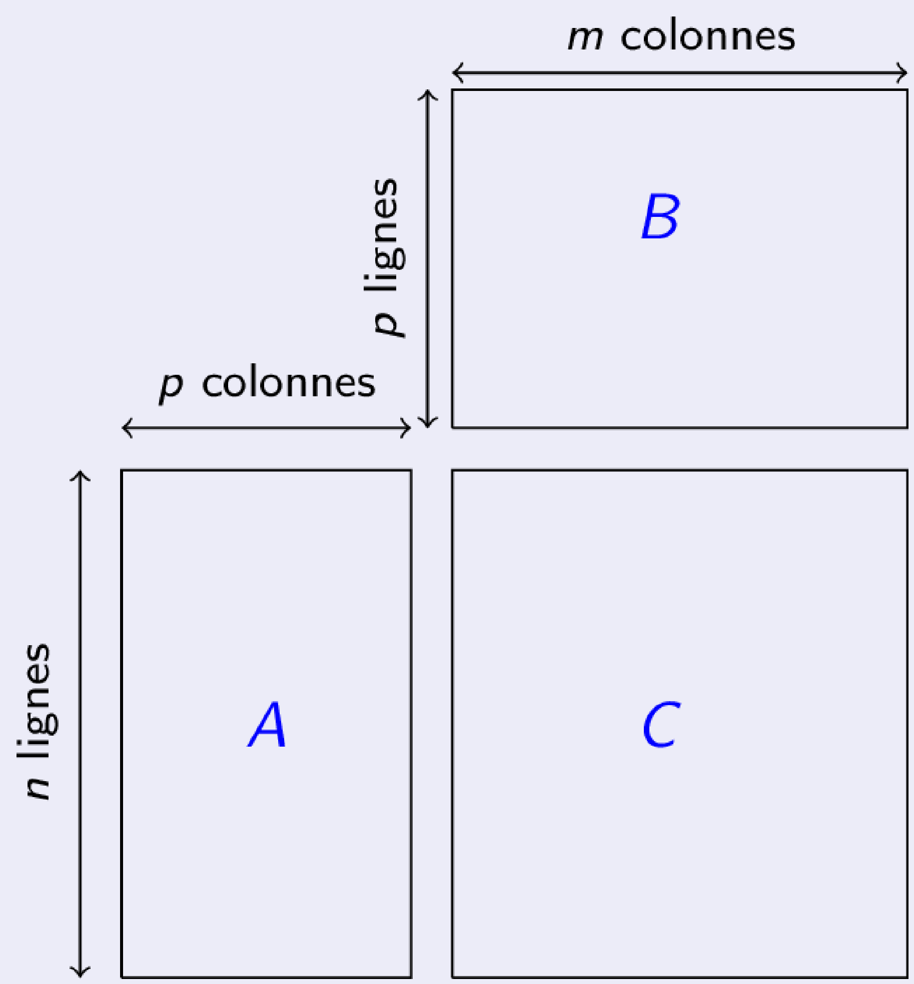
Opérations matricielles I

- Transposition
$$C = A^\top \Leftrightarrow c_{ij} = a_{ji}$$
- Addition
$$C = A + B \Leftrightarrow c_{ij} = a_{ij} + b_{ij}$$
- Multiplication par un scalaire
$$C = \alpha A \Leftrightarrow c_{ij} = \alpha a_{ij}$$
- Multiplication
$$C = AB \Leftrightarrow c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

Opérations matricielles II

Attention aux dimensions

Faire un



Gaxpy : Generalized Ax plus y

- Tiré de [Golub and Van Loan, 1996]

$$\mathbf{y} = A\mathbf{x} + \mathbf{y} \qquad \Leftrightarrow \qquad y_i = \sum_{j=1}^{p} a_{ij} x_j + y_i \qquad i = 1, n$$

- Deux versions :

Version **ligne** :

Fonction $\mathbf{y} \leftarrow \texttt{GAxpy}(A, \mathbf{x}, \mathbf{y})$

for $i = 1, n$
$\quad y_i = A(i, :) * x + y_i$
end

Version **colonne** :

Fonction $\mathbf{y} \leftarrow \texttt{GAxpy}(A, \mathbf{x}, \mathbf{y})$

for $j = 1, p$
$\quad \mathbf{y} = A(:, j) * x_j + \mathbf{y}$
end

Gaxpy : exemple

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} + \begin{pmatrix} 7 \\ 8 \end{pmatrix} =$$

Gaxpy : exemple

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}\begin{pmatrix}5\\ 6\end{pmatrix}+\begin{pmatrix}7\\ 8\end{pmatrix} = \begin{array}{lcl} & \begin{pmatrix}5\\ 6\end{pmatrix} & \\ \begin{pmatrix}1 & 2\end{pmatrix} & & +\;7\\ \begin{pmatrix}3 & 4\end{pmatrix} & & +\;8\end{array}$$

$$= \begin{pmatrix}1*5+2*6+7\\ 3*5+4*6+8\end{pmatrix} \qquad \text{(Version ligne)}$$

$$= 5\begin{pmatrix}1\\ 3\end{pmatrix}+6\begin{pmatrix}2\\ 4\end{pmatrix}+\begin{pmatrix}7\\ 8\end{pmatrix} \qquad \text{(Version colonne)}$$

Produit extérieur I

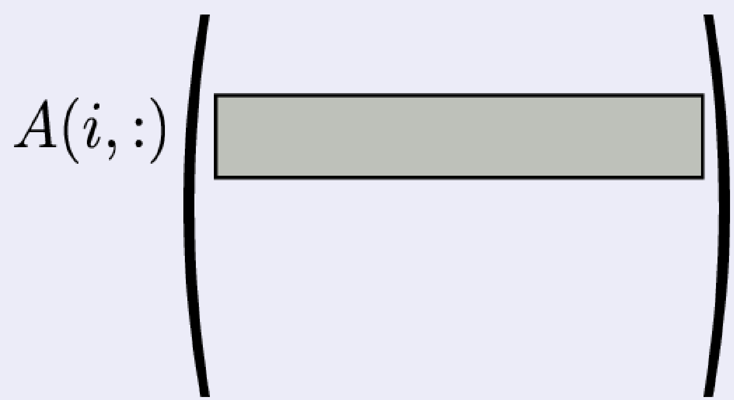
Outer product $A = \mathbf{vw}^T$

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} \begin{pmatrix} w_1 & w_2 & w_3 & w_4 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix}$$

$A = A + \mathbf{vw}^\top$

- Version ligne

Fonction $A \leftarrow \texttt{outer}(A, \mathbf{v}, \mathbf{w})$
for $i = 1, n$
$\quad A(i,:) = A(i,:) + v(i) * \mathbf{w}^\top$
end

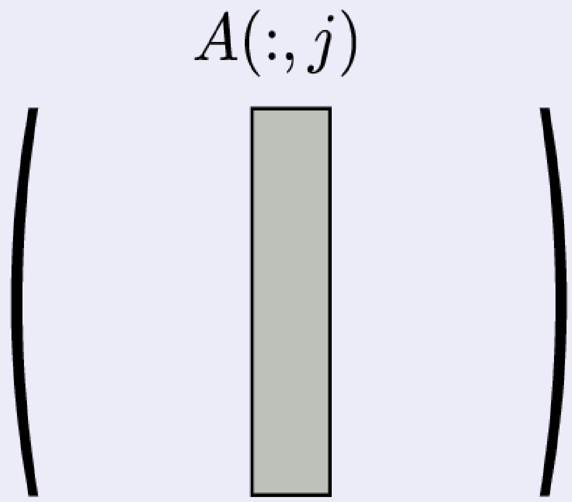


Produit extérieur II

$A = A + \mathbf{vw}^\top$

- version colonne

Fonction $A \leftarrow \texttt{outer}(A, \mathbf{v}, \mathbf{w})$
for $j = 1, m$
$\quad A(:, j) = A(:, j) + \mathbf{v}\mathbf{w}(j)$
end



Produit Matriciel

C = AB

$A \in \mathbb{R}^{n \times p}$, $B \in \mathbb{R}^{p \times n}$,

$$C = AB \quad \Leftrightarrow \quad C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

⚠ Attention aux dimensions... cf. Slide

$$A = \begin{pmatrix} 5 & 6 & 0 \\ 7 & 8 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix}$$

Produit Matriciel

C = AB

$A \in \mathbb{R}^{n \times p}$, $B \in \mathbb{R}^{p \times n}$,

$$C = AB \quad \Leftrightarrow \quad C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

Attention aux dimensions... cf. Slide

$$\begin{array}{cc} & \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 5 & 6 & 0 \\ 7 & 8 & 1 \end{pmatrix} & \begin{pmatrix} \bullet & \\ & \end{pmatrix} \end{array}$$

Produit Matriciel

C = AB

$A \in \mathbb{R}^{n \times p}$, $B \in \mathbb{R}^{p \times n}$,

$$C = AB \quad \Leftrightarrow \quad C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

Attention aux dimensions... cf. Slide

$$\begin{array}{cc}
 & \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix} \\
\begin{pmatrix} 5 & 6 & 0 \\ 7 & 8 & 1 \end{pmatrix} & \begin{pmatrix} & \bullet \\ & \end{pmatrix}
\end{array}$$

Produit Matriciel

$C = AB$

$A \in \mathbb{R}^{n \times p}, B \in \mathbb{R}^{p \times n}$,

$$C = AB \quad \Leftrightarrow \quad C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

⚠ Attention aux dimensions... cf. Slide

$$\begin{array}{cc} & \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 5 & 6 & 0 \\ 7 & 8 & 1 \end{pmatrix} & \begin{pmatrix} & \\ \bullet & \end{pmatrix} \end{array}$$

Produit Matriciel

C = AB

$A \in \mathbb{R}^{n \times p}, B \in \mathbb{R}^{p \times n},$

$$C = AB \quad \Leftrightarrow \quad C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$

Attention aux dimensions... cf. Slide

$$\begin{array}{cc} & \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 5 & 6 & 0 \\ 7 & 8 & 1 \end{pmatrix} & \begin{pmatrix} & \\ & \bullet \end{pmatrix} \end{array}$$

Plan

Huit versions du produit de deux matrices I

Comment calculer le produit de deux matrices ?

Différentes stratégies

$\Rightarrow$ Différentes performances

$\rightarrow$ TP

Huit versions du produit de deux matrices II

Une première approche : 3 boucles

Pour $i = 1, n$ **faire**
  Pour $j = 1, m$ **faire**
    $s = 0$;
    Pour $k = 1, p$ **faire**
      $s = s + A_{ik}B_{kj}$
    Fin Pour
    $C_{ij} = s$
  Fin Pour
Fin Pour

en éliminant s.

Pour $i = 1, n$ **faire**
  Pour $j = 1, m$ **faire**
    $C_{ij} = 0$;
    Pour $k = 1, p$ **faire**
      $C_{ij} = C_{ij} + A_{ik}B_{kj}$
    Fin Pour
  Fin Pour
Fin Pour

NB : L'ordre d'inclusion des boucles peut changer les performances...

Huit versions du produit de deux matrices III

Deux boucles : Approche produit scalaire

Pour $i = 1, n$ **faire**
  Pour $j = 1, m$ **faire**
    $C_{ij} = C_{ij} + A(i,:)B(:,j)$
  Fin Pour
Fin Pour

Huit versions du produit de deux matrices IV

Deux boucles : Approche ligne

Pour $i = 1, n$ **faire**
  $C(i,:) = 0$;
  Pour $k = 1, p$ **faire**
    $C(i,:) = C(i,:) + A(i,k)B(k,:)$
  Fin Pour
Fin Pour

Huit versions du produit de deux matrices V

Deux boucles : Approche colonne

Pour $j = 1, m$ **faire**
- $C(:,j) = 0$;
- **Pour** $k = 1, p$ **faire**
  - $C(:,j) = C(:,j) + A(:,k)B(k,j)$
- **Fin Pour**

Fin Pour

Simplification I

Codons $C = AB + C \Rightarrow$ plus d'initialisation.

$$\begin{pmatrix} & & \\ & B & \\ & & \end{pmatrix}$$

$$\begin{pmatrix} & & \\ & C & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & C & \\ & & \end{pmatrix} + \begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix}$$

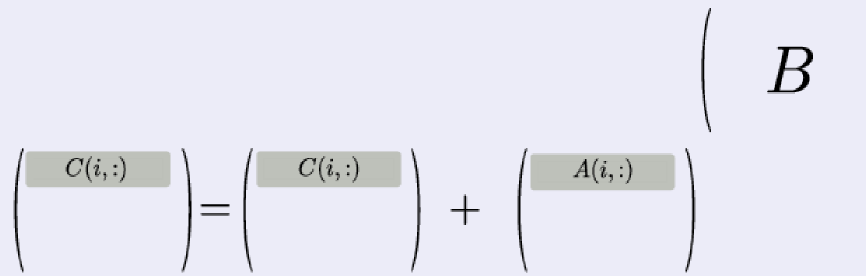
Simplification II

Version Ligne

Pour $i = 1, n$ **faire**

$\quad C(i,:) = C(i,:) + A(i,:)B$

Fin Pour

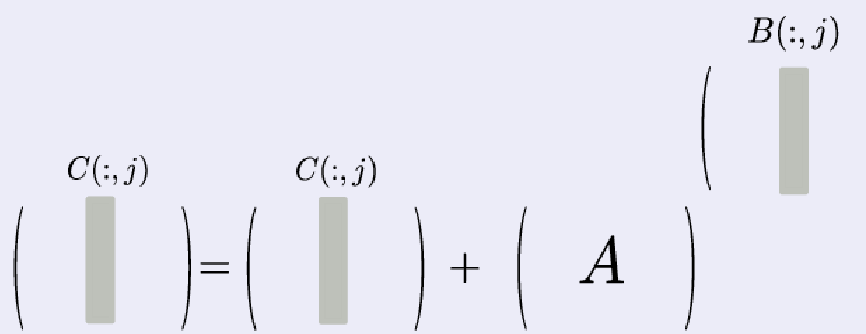


Version Colonne (Gaxpy)

Pour $j = 1, m$ **faire**

$\quad C(:,j) = C(:,j) + AB(:,j)$

Fin Pour



Simplification III

Global (outer)

Pour $k = 1, p$ **faire**

  $C = C + A(:, k)B(k, :)$ % Outer($A(:, k), B(k, :), C$)

Fin Pour

$$\begin{pmatrix} C \end{pmatrix} = \begin{pmatrix} C \end{pmatrix} + \begin{array}{c} A(:, j) \\ \begin{pmatrix} \; \end{pmatrix} \end{array} \begin{array}{c} \begin{pmatrix} B(i, :) \end{pmatrix} \\ \; \end{array}$$

Questions ?

- Quel critère peut influencer la vitesse de calcul ?
  1. Le nombre d'opérations
  2. L'organisation des matrices en mémoire
  3. L'implémentation des opérations de base

maths $\neq$ informatique

- La machine n'est pas infinie...
- $\exists\varepsilon \mid x + \varepsilon = x$
- $0 \approx 10^{-16}$ dans Matlab
- $A = \begin{pmatrix} 10^{-9} & 10^{-8} \\ 10^{-9} & 10^{-9} \end{pmatrix}$
- $\det(A) = 0$ en informatique

Calcul matriciel en informatique

- Attention aux faibles valeurs $\rightarrow$ stabilité
- Utilisation des ressources : temps CPU et espace mémoire
- Utilisation optimale : LAPACK et BLAS (via ATLAS), Matlab, ...

Complexité

Tout dépend du nombre de boucles considérées...

Calcul du produit de deux matrices $n \times n$

Pour $i = 1, n$, **faire**

//n opérations

Pour $j = 1, n$ **faire**

//n opérations

$s = 0$;

Pour $k = 1, n$ **faire**

//n opérations

$s = s + A_{ik} B_{kj}$

Fin Pour

$C_{ij} = s$

Fin Pour

Fin Pour

Complexité $\mathcal{O}(n^3)$

Conclusion

- Notion de matrice : math / informatique / physique
- $A\mathbf{x} = \mathbf{b}$
- LAPACK
- histoire de boucles