Analyse Numérique Appliquée
Notes de cours

Basile Sauvage

2019

UFR de mathématique et d'informatique
Université de Strasbourg

Table des matières

1 Introduction **5**

Rappels et compléments 7

2 Rappels d'algèbre **9**
2.1 Espaces vectoriels . . . 9
2.2 Familles et bases . . . 9
2.3 Espace euclidien, produit scalaire et norme . . . 11
2.4 Applications . . . 12

3 Rappels sur la complexité **13**
3.1 Complexité asymptotique . . . 13
3.2 Ordres de grandeur . . . 13

4 Rappels de statistiques **15**

Cours 17

5 Les matrices **19**
5.1 Définitions . . . 19
5.2 Algorithmes standards sur les vecteurs et les matrices . . . 20
5.3 Représentations informatiques des matrices . . . 21

6 Résolution de systèmes linéaires **23**
6.1 Position du problème . . . 23
6.2 Exemple . . . 24
6.3 Cas des systèmes triangulaires . . . 24
6.4 Méthode du pivot de Gauss . . . 25
6.5 Stabilité numérique . . . 27

7 Approximation au sens des moindres carrés **29**
7.1 Position du problème . . . 29
7.2 Résolution par projection orthogonale . . . 29
7.3 Résolution par minimisation de la distance quadratique . . . 30
7.4 Régression linéaire . . . 30
7.5 Approximation polynômiale . . . 31
7.6 Moindres carrés pondérés, une question de confiance . . . 32
7.7 Changements de variable . . . 32

8 Applications linéaires **33**
8.1 Notations . . . 33
8.2 Espaces vectoriels et espaces affines . . . 33
8.3 Applications linéaires . . . 33
8.4 Changement de base . . . 36

9 Éléments propres **39**
9.1 Définitions . . . 39
9.2 Calcul de l'espace propre associé à une valeur propre réelle connue . . . 39
9.3 Diagonalisation . . . 41
9.4 Calcul approché des valeurs propres et vecteurs propres . . . 43

Annexes **45**

A Comment éviter les erreurs ? **47**

B Questions de cours **49**
B.1 Rappels d'algèbre . . . 49
B.2 Rappels sur la complexité . . . 49
B.3 Les matrices . . . 49
B.4 Résolution de systèmes linéaires . . . 49
B.5 Approximation au sens des moindres carrés . . . 49
B.6 Applications linéaires . . . 50
B.7 Éléments propres . . . 50

C Annales **51**
C.1 Les matrices . . . 51
C.2 Résolution de systèmes linéaires . . . 52
C.3 Approximation au sens des moindres carrés . . . 53
C.4 Applications linéaires . . . 54
C.5 Éléments propres . . . 54

Chapitre 1

Introduction

Pourquoi ce cours ?

Certains domaines de l'informatique, dont ceux enseignés dans nos Master, nécessitent de savoir *utiliser* et *programmer* un certain nombre d'outils mathématiques standards. Pour cela il faut les *comprendre* et les *manipuler*, en un mot les démystifier.

Dans ce cours nous reviendrons entre autres sur des notions mathématiques connues. Nous ferons peu de démonstrations, pour nous concentrer sur les algorithmes, les calculs et les applications. L'analyse numérique est parfois preçue comme mathématique par les informaticiens, et comme informatique par les mathématiciens : c'est qu'il s'agit d'écrire et de programmer des algorithmes pour faire des calculs mathématiques, en tenant compte des difficultés liées aux ordinateurs (imprécisions numériques, temps de calcul, et occupation mémoire).

Quel contenu ?

Les sujets abordés sont les suivants :

- Espaces vectoriels et matrices.
- Résolution de systèmes linéaires d'équations (calcul direct, factorisation).
- Algorithmes d'approximation : moindres carrés, régression linéaire, régression polynomiale, et transformation de variable.
- Applications linéaires et changements de base.
- Calcul de valeurs propres, vecteurs propres, diagonalisation.
- Applications sur un logiciel de calcul numérique.

Quelles compétences visées ?

À l'issue de cet enseignement, vous saurez :

- Utiliser des algorithmes pour résoudre des systèmes linéaires et diagonaliser des matrices.
- Effectuer des changements de bases.
- Mener des régressions avec transformation des variables.
- Utiliser un logiciel de calcul numérique.
- Tester expérimentalement la complexité asymptotique d'un algorithme.
- Observer numériquement la convergence d'un algorithme itératif.

Quelles modalités pédagogiques ?

Ce module est composé de
- 12h de CM, où nous chercherons à comprendre les notions essentielles, à l'aide d'exemples et de schémas qui compléteront ces notes de cours.
- 6h de TD, où vous réaliserez des calculs et écrirez des algorithmes ;
- 6h de TP, où vous utiliserez les algorithmes pour faire des mesures sur des données réelles.

Ce module sera évalué par
- 2 contrôles sur table, où nous évaluerons les aspects mathématiques, calculatoires et algorithmiques ;
- 1 contrôle sur machine, où nous évaluerons votre capacité à appliquer des algorithmes à l'aide d'un logiciel de calcul numérique, pour analyser et observer des données.

Quels pré-requis ?

Nous nous appuierons d'une part sur des connaissances mathématiques (espaces vectoriels, bases, matrices, systèmes linéaires, applications linéaires) abordées en particulier dans les cours d'algèbre de première année. D'autre part, vous utiliserez vos compétences informatiques (algorithmes itératifs, complexité, tableaux, listes) acquises en première année (algorithmique et programmation) et en deuxième année (structures de données et algorithmes).

Rappels et compléments

Chapitre 2

Rappels d'algèbre

2.1 Espaces vectoriels

Définition Soit K un corps commutatif ; ses éléments sont appelés scalaires. Un espace vectoriel sur le corps K (aussi appelé K-espace vectoriel) est un ensemble $\mathbf{E}$ non vide d'éléments (appelés vecteurs), muni d'une loi interne (appelée addition) et une loi externe (multiplication par un scalaire), tels que :

- L'addition est une loi interne ($\mathbf{u}+\mathbf{v} \in \mathbf{E}, \forall \mathbf{u}, \mathbf{v} \in \mathbf{E}$) vérifiant les propriétés suivantes :
  1. elle est commutative : $\mathbf{u}+\mathbf{v} = \mathbf{v}+\mathbf{u}, \forall \mathbf{u}, \mathbf{v} \in \mathbf{E}$;
  2. elle est associative : $(\mathbf{u}+\mathbf{v})+\mathbf{w} = \mathbf{u}+(\mathbf{v}+\mathbf{w}), \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{E}$;
  3. elle admet un élément neutre (noté $\mathbf{0}$) : $\mathbf{u}+\mathbf{0} = \mathbf{u}, \forall \mathbf{u} \in \mathbf{E}$;
  4. chaque vecteur $\mathbf{u} \in \mathbf{E}$ admet un opposé (noté $-\mathbf{u}$) : $\mathbf{u}+(-\mathbf{u}) = \mathbf{0}$.
- La multiplication par un scalaire est une loi externe ($c\mathbf{u} \in \mathbf{E}, \forall c \in K, \forall \mathbf{u} \in \mathbf{E}$) vérifiant les propriétés suivantes :
  1. elle est distributive sur l'addition des vecteurs : $c(\mathbf{u}+\mathbf{v}) = c\mathbf{u}+c\mathbf{v}, \forall c \in K, \forall \mathbf{u}, \mathbf{v} \in \mathbf{E}$;
  2. elle est distributive sur l'addition des scalaires : $(c+d)\mathbf{u} = c\mathbf{u}+d\mathbf{u}, \forall c, d \in K, \forall \mathbf{u} \in \mathbf{E}$;
  3. $c(d\mathbf{u}) = (cd)\mathbf{u}, \forall c, d \in K, \forall \mathbf{u} \in \mathbf{E}$;
  4. $1.\mathbf{u} = \mathbf{u}, \forall \mathbf{u} \in \mathbf{E}$;

Il en découle certaines propriétés, telles que, pour $c \in K$ et $\mathbf{u} \in \mathbf{E}$: $0\mathbf{u} = \mathbf{0}$; $c\mathbf{0} = \mathbf{0}$; $-\mathbf{u} = (-1)\mathbf{u}$.

Exemple 2.1

- $\mathbf{E} = \mathbb{R}^d$ est un e.v. sur $K = \mathbb{R}$ (en particulier pour $d = 1$).
- $\mathbf{E} = \mathbb{C}^d$ est un e.v. sur $\mathbb{R}$ ou sur $\mathbb{C}$ (en particulier pour $d = 1$).
- L'ensemble $\mathbf{E} = \mathbb{R}_d[X]$ des polynômes réels de degré inférieur ou égal à d est un e.v. sur $\mathbb{R}$.

2.2 Familles et bases

Définition. Soient deux vecteurs $\mathbf{u}$ et $\mathbf{v} \in \mathbf{E}$. Ils sont colinéaires (i.e. ils ont la *même direction*) si et seulement si $\exists c \in K,\ \mathbf{v} = c\mathbf{u}$.
Si $c > 0$ ils ont le *même sens*.

Définition. Soient $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n \in \mathbf{E}$ et $c_1, c_2, \ldots, c_n \in K$.

$$\sum_{i=1}^{n} c_i \mathbf{u}_i \text{ est appelée combinaison linéaire des } \mathbf{u}_i$$

Définition. Des vecteurs $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n \in \mathbf{E}$ sont linéairement indépendants (i.e. ils forment une famille libre) si et seulement si

$$\sum_{i=1}^{n} c_i \mathbf{u}_i = \mathbf{0} \quad \Rightarrow \quad c_i = 0, \forall i$$

Définition. Un espace vectoriel est de dimension d si on peut trouver d vecteurs linéairement indépendants, mais que $d+1$ vecteurs sont toujours linéairement dépendants.

Remarque. Un espace vectoriel peut être de dimension infinie.

Définition. Soient $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ des vecteurs de $\mathbf{E}$. L'ensemble de toutes les combinaisons linaires

$$\mathcal{L}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\} = \left\{ \sum_{i=1}^{n} c_i \mathbf{u}_i, \quad \forall c_1, c_2, \ldots, c_n \in K \right\}$$

est un sous-espace vectoriel de $\mathbf{E}$, appelé sous-espace *engendré* par cette famille.

Remarque. Grâce à sa stabilité par combinaison linéaire, ce sous-espace "hérite" de la structure d'espace vectoriel de $\mathbf{E}$. Il est donc facile de construire des espaces vectoriels de cette manière.

Définition. Une base de $\mathbf{E}$ est une famille libre qui engendre tout $\mathbf{E}$.
Si $\mathbf{E}$ est de dimension finie d, une base $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$ contient d vecteurs et $\mathcal{L}\{\mathbf{e}_1, \ldots, \mathbf{e}_d\} = \mathbf{E}$

Exemple 2.2

- $\mathbb{R}^d$ est de dimension d, et admet comme base

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} ; \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} ; \quad \ldots \quad ; \quad \mathbf{e}_d = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

- $\mathbb{R}_d[X]$ est de dimension $d+1$, et admet comme base

$$\left\{1, X, X^2, \ \ldots \ , X^d\right\}.$$

Remarque. Ces bases usuelles sont dites canoniques.

Ecriture dans une base. Soit $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$ une base de $\mathbf{E}$. Tout vecteur $\mathbf{u} \in \mathbf{E}$ se décompose de manière unique dans cette base, i.e. il existe des uniques scalaires $u_1, u_2, \ldots, u_d$ (appelés coordonnées de $\mathbf{u}$) tels que

$$\mathbf{u} = \sum_{i=1}^{d} u_i \mathbf{e}_i.$$

Exemple 2.3

- Dans $\mathbb{R}^3$ muni de sa base canonique,

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

  Donc $\mathbf{u}$ a pour coordonnées $(2, 3, 1)$.

- Dans $\mathbb{R}^3$ muni de la base

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}, \ \mathbf{e}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{e}_3 = \begin{bmatrix} 0 \\ 1 \\ -5 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix} + 0 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + (-1) \begin{bmatrix} 0 \\ 1 \\ -5 \end{bmatrix}.$$

  Donc le même vecteur $\mathbf{u}$ a pour coordonnées $(2, 0, -1)$.

- Dans $\mathbb{R}_3[X]$ muni de sa base canonique $\{1, X, X^2, X^3\}$,

$$P(X) = X^3 + 2X^2 - 3X - 3$$

a pour coordonnées $(-3, -3, 2, 1)$.
- Dans $\mathbb{R}_3[X]$ muni de la base $\{1, (X+1), X^2, (X^3+X^2)\}$, le même polynôme

$$P(X) = X^3 + 2X^2 - 3X - 3 = (X^3 + X^2) + X^2 - 3(X+1) + 0$$

a pour coordonnées $(0, -3, 1, 1)$.

Important. L'application qui à $\mathbf{u} \in \mathbf{E}$ associe ses coordonnées $(u_1, u_2, \ldots, u_d) \in K^d$ est linéaire et bijective. Pour cette raison, nous allons nous intéresser aux espaces $\mathbf{E} = K^d$, en particulier $\mathbb{R}^d$: on peut toujours s'y ramener, à condition de fixer une base de $\mathbf{E}$, et d'exprimer notre problème dans cette base.

2.3 Espace euclidien, produit scalaire et norme

Dans toute cette section, on se place dans l'espace vectoriel $\mathbb{R}^n$.
Soient deux vecteurs

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \text{ et } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

On définit un produit scalaire

$$\langle\ \mathbf{u}\ ,\ \mathbf{v}\ \rangle = \sum_{i=1}^{n} u_i v_i = \mathbf{u}^T \mathbf{v} = [u_1, u_2, \ldots, u_n] \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

On dit que $\mathbb{R}^n$ est *muni* de ce produit scalaire. C'est un espace *euclidien*.

Exemple 2.4
Calculer

$$\left\langle \begin{bmatrix} 2 \\ -5 \\ -1 \end{bmatrix}\ ,\ \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix} \right\rangle = 2 * 3 + (-5) * 2 + (-1) * (-3) = -1$$

Le produit scalaire vérifie les propriétés suivantes :
- $\langle\ \mathbf{u}\ ,\ \mathbf{v}\ \rangle = \langle\ \mathbf{v}\ ,\ \mathbf{u}\ \rangle$ (commutativité).
- $\langle\ \mathbf{u} + \mathbf{v}\ ,\ \mathbf{w}\ \rangle = \langle\ \mathbf{u}\ ,\ \mathbf{w}\ \rangle + \langle\ \mathbf{v}\ ,\ \mathbf{w}\ \rangle$ (distributivité).
- $\langle\ c\mathbf{u}\ ,\ \mathbf{v}\ \rangle = c\langle\ \mathbf{u}\ ,\ \mathbf{v}\ \rangle$, où $c \in \mathbb{R}$.
- $\langle\ \mathbf{u}\ ,\ \mathbf{u}\ \rangle \geq 0$ et $\langle\ \mathbf{u}\ ,\ \mathbf{u}\ \rangle = 0$ si et seulement si $u = 0$.

Définition. La norme euclidienne (ou longueur) d'un vecteur $\mathbf{u}$ est $\|\mathbf{u}\| = \sqrt{\langle\ \mathbf{u}\ ,\ \mathbf{u}\ \rangle}$.

On dit que $\mathbb{R}^n$ est *muni* de la norme euclidienne. C'est un *espace vectoriel normé*.

Définition. La distance entre deux vecteurs $\mathbf{u}$ et $\mathbf{v}$ est $\|\mathbf{u} - \mathbf{v}\|$.
Définition. Deux vecteurs $\mathbf{u}$ et $\mathbf{v}$ sont dits orthogonaux si $\langle\ \mathbf{u}\ ,\ \mathbf{v}\ \rangle = 0$.

Définition. Un vecteur $\mathbf{u}$ est dit normé (ou unitaire) si $\|\mathbf{u}\| = 1$.
Définition. Une base $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ est dite orthogonale si

$$\langle\ \mathbf{e}_i\ ,\ \mathbf{e}_j\ \rangle = 0 \quad \text{pour tout } i \neq j.$$

Définition. Une base $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ est dite orthonormale si pour tout $1 \leq i, j \leq n$

$$\langle\ \mathbf{e}_i\ ,\ \mathbf{e}_j\ \rangle = \delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Une base orthonormale est donc une base orthogonale de vecteurs normés.

2.4 Applications

Soit f une application d'un espace de départ $\mathbf{E}$ dans un espace d'arrivée $\mathbf{F}$.
Définition. f est dite injective si et seulement si
tout élément de $\mathbf{F}$ a *au plus* un antécédant
$\Leftrightarrow\ \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbf{E},\ f(\mathbf{u}_1) = f(\mathbf{u}_2) \Rightarrow \mathbf{u}_1 = \mathbf{u}_2$
$\Leftrightarrow\ \forall \mathbf{u}_1, \mathbf{u}_2 \in \mathbf{E},\ \mathbf{u}_1 \neq \mathbf{u}_2 \Rightarrow f(\mathbf{u}_1) \neq f(\mathbf{u}_2)$.

Définition. f est dite surjective si et seulement si
tout élément de $\mathbf{F}$ a *au moins* un antécédant
$\Leftrightarrow\ \forall \mathbf{v} \in \mathbf{F},\ \exists \mathbf{u} \in \mathbf{E} / f(\mathbf{u}) = \mathbf{v}$.

Définition. f est dite bijective si et seulement si
tout élément de $\mathbf{F}$ a *exactement* un antécédant
$\Leftrightarrow$ f est injective et surjective
$\Leftrightarrow\ \forall \mathbf{v} \in \mathbf{F},\ \exists! \mathbf{u} \in \mathbf{E} / f(\mathbf{u}) = \mathbf{v}$.

Chapitre 3

Rappels sur la complexité

Définition. Un algorithme est une suite finie d'opérations élémentaires constituant un schéma de calcul ou de résolution d'un problème.

Définition. La compléxité en temps d'un algorithme est la mesure du nombre d'opérations élémentaires qu'il effectue sur un jeu de données. Elle est exprimée comme une fonction (notée T) de la taille du jeu de données (notée n).

L'unité de mesure sera ici l'opération flottante (flop). On sépare parfois les additions/soustractions des multiplications/divisions, ces dernières coûtant plus cher.

3.1 Complexité asymptotique

Quand les flop ne sont pas dénombrées exactement, on calcule une complexité asymptotique, qui caractérise le comportement de la comlexité *quand la taille des données devient grande*. Rappelons les notations de Landau (vues en SDA1 et SDA2) :

- $T(n) = 0(g(n))$ signifie $\exists c, n_0$ positives telles que

  $$\forall n \geq n_0,\ 0 \leq T(n) \leq cg(n)$$

  On dit que g est une borne asymptotique supérieure.
- $T(n) = \Omega(g(n))$ signifie $\exists c, n_0$ positives telles que

  $$\forall n \geq n_0,\ 0 \leq cg(n) \leq T(n)$$

  On dit que g est une borne asymptotique inférieure.
- $T(n) = \Theta(g(n))$ signifie $\exists c_1, c_2, n_0$ positives telles que

  $$\forall n \geq n_0,\ 0 \leq c_1 g(n) \leq T(n) \leq c_2 g(n)$$

  On dit que g borne asymptotiquement T.

3.2 Ordres de grandeur

$T(n)$	$\log_2(n)$	n	$n\log_2(n)$	n^2	n^3	2^n
$T(10^1)$	3	10	33	100	1 000	1 000
$T(10^2)$	7	100	660	10 000	10^{06}	[d]$1,3.10^{30}$
$T(10^3)$	10	1 000	10 000	10^{06}	10^{09}	∞
$T(10^4)$	13	10 000	130 000	10^{08}	[b]10^{12}	∞
$T(10^5)$	17	100 000	$1,7.10^6$	10^{10}	10^{15}	∞
$T(10^6)$	20	1 000 000	[a]20.10^6	[b]10^{12}	[c]10^{18}	∞

Sur une machine qui fait 1 milliard d'opération par seconde (1 giga-flops), les temps d'exécution :

[a] 20 ms.
[b] 17 minutes.
[c] 32 années.
[d] 40 milliards d'années.

Ordres de grandeur pour n :

- Téléphonie mobile (France) : plusieurs 10^4 antennes émettrices, plusieurs 10^7 abonnés.
- Web mondial : 10^{14} pages web référencées.

Chapitre 4

Rappels de statistiques

Une variable. Soit $\mathbf{x} = \{x_1, \ldots, x_n\} \in \mathbb{R}$ un ensemble de n valeurs (appelé *échantillon*).
La moyenne de $\mathbf{x}$ vaut $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{n} x_i$.
La variance de $\mathbf{x}$ vaut $Var(\mathbf{x}) = \frac{1}{N-1} \sum_{i=1}^{n} (x_i - \bar{\mathbf{x}})^2$.
La variance mesure la dispersion de l'échantillon autour de la moyenne : $Var(\mathbf{x}) = 0$ si tous les x_i sont égaux, et elle augmente quand les x_i s'éloignent les uns des autres. On utilise aussi l'écart type de $\mathbf{x}$, qui vaut $\sqrt{Var(\mathbf{x})}$.

Deux variables. Soient $\mathbf{x} = \{x_1, \ldots, x_n\}$ et $\mathbf{y} = \{x_1, \ldots, x_n\}$ deux échantillons réels.
La covariance de $\mathbf{x}$ et $\mathbf{y}$ vaut $Cov(\mathbf{x}, \mathbf{y}) = \frac{1}{N-1} \sum_{i=1}^{n} (x_i - \bar{\mathbf{x}})(y_i - \bar{\mathbf{y}})$.
La covariance mesure la façon dont $\mathbf{x}$ et $\mathbf{y}$ varient l'un par rapport à l'autre : la covariance est positive s'ils on tendance à croître et décroître ensemble ; elle est négative si l'un a tendance à croître quand l'autre décroît ; elle est nulle s'il n'y a pas de lien net entre $\mathbf{x}$ et $\mathbf{y}$.
Le coefficient de corrélation entre $\mathbf{x}$ et $\mathbf{y}$ vaut $Cor(\mathbf{x}, \mathbf{y}) = \frac{Cor(\mathbf{x}, \mathbf{y})}{\sqrt{Var(\mathbf{x})}\sqrt{Var(\mathbf{y})}}$.
Le coefficient de corrélation donne une information similaire à la covariance, sauf que sa valeur est comprise dans $[-1; 1]$. La valeur 1 indique une corrélation linéaire parfaite et positive ; -1 indique une corrélation linéaire parfaite et négative ; 0 indique une absence de corrélation.

Estimateurs et biais. Les formules ci-dessus sont appelés des *estimateurs*, car elles permettent d'estimer des grandeurs (moyenne, variance, et covariance) à partir d'un échantillon. Il peut exister plusieurs estimateurs pour une même grandeur. Dans les formules de variance et covariance, attention au numérateur, qui est bien $(N-1)$ et non N, ce qui rend ces estimateurs non *biaisés*.

Cours

Chapitre 5

Les matrices

5.1 Définitions

Une matrice réelle A à n lignes et p colonnes s'écrit

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & \dots & a_{2p} \\ \vdots & & & & \vdots \\ a_{n1} & \dots & \dots & \dots & a_{np} \end{bmatrix}, \quad \text{avec } a_{ij} \in \mathbb{R}.$$

Transposition. La matrice notée A^T, transposée de A, a p lignes égales aux colonnes de A et n colonnes égales aux lignes de A.

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a_{1p} & \dots & \dots & a_{np} \end{bmatrix}$$

En notant $B = A^T$, on a donc $b_{ji} = a_{ij}$.
La transposition vérifie $(A+B)^T = A^T + B^T$ et $(AB)^T = B^T A^T$.
On remarque la cohérence des dimensions :

- A : n lignes et p colonnes ;
- B : p lignes et q colonnes ;
- AB : n lignes et q colonnes ;
- A^T : p lignes et n colonnes ;
- B^T : q lignes et p colonnes ;
- $B^T A^T$: q lignes et n colonnes.

Exemple 5.1

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Matrices carrées. Une matrice est dite carrée si et seulement si $n = p$. Une matrice carrée peut avoir des propriétés particulières :

- Elle est **diagonale** si seuls ses éléments diagonaux sont non nuls. Exemple :

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Elle est **symétrique** si $A = A^T$, c'est à dire si $a_{i,j} = a_{j,i}$ pour tout $1 \leq i, j \leq n$. Exemple :

$$\begin{bmatrix} 2 & -1 & 5 & 0 \\ -1 & 3 & 0 & -2 \\ 5 & 0 & 2 & 0 \\ 0 & -2 & 0 & 1 \end{bmatrix}$$

- Elle est **triangulaire** supérieure (resp. inférieure) si tous les éléments en dessous (resp. dessus) de la diagonale sont nuls. Exemple :

$$\begin{bmatrix} 2 & -2 & 5 & 3 \\ 0 & 3 & 2 & -1 \\ 0 & 0 & 2 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Elle est **inversible** s'il existe une matrice notée A^{-1}, appelée "inverse de A", qui vérifie $AA^{-1} = A^{-1}A = I$.
- Elle est **définie positive** (positive definite) si $x^T Ax > 0$ pour tout $x \in \mathbb{R}^n$, $x \neq 0$.
- Elle est **semi-définie positive** si $x^T Ax \geq 0$ pour tout $x \in \mathbb{R}^n$.

5.2 Algorithmes standards sur les vecteurs et les matrices

- La multiplication d'un vecteur de $\mathbb{R}^n$ par un scalaire coûte $n = \Theta(n)$ multiplications.

$$y = \lambda x = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{bmatrix}$$

pour $i = 1$ à n **faire**
  $y(i) \leftarrow \lambda * x(i)$
fin pour

  Exemple : opérations sur les lignes dans la résolution d'un système.
- La multiplication d'une matrice $n \times p$ par un scalaire coûte $np = \Theta(np)$ multiplications, donc $\Theta(n^2)$ si la matrice est carrée.

$$B = \lambda A = \lambda \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & & & \vdots \\ a_{n1} & \dots & \dots & a_{np} \end{bmatrix}$$

pour $i = 1$ à n **faire**
  pour $j = 1$ à p **faire**
    $b(i,j) \leftarrow \lambda * a(i,j)$
  fin pour
fin pour

- Une addition de vecteurs de $\mathbb{R}^n$ coûte n additions, i.e. $\Theta(n)$ flops.

$$z = x + y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

pour $i = 1$ à n **faire**
  $z(i) \leftarrow x(i) + y(i)$
fin pour

  Exemples : opérations sur les lignes dans la résolution d'un système.
- Un produit scalaire dans $\mathbb{R}^n$ coûte n multiplications et $n - 1$ additions, i.e. $\Theta(n)$ flops.

$$x \cdot y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

$s \leftarrow 0$
pour $i = 1$ à n **faire**
  $s \leftarrow s + x(i) * y(i)$
fin pour

Exemples : calcul de coordonnées dans une base, calcul de distances euclidiennes, de normes.

- Le produit d'une matrice $n \times p$ par un vecteur de $\mathbb{R}^p$ coûte $\Theta(np)$ flops.

$$y = Ax, \quad y_i = \sum_{j=1}^{p} a_{ij} x_j$$

pour $i = 1$ à n **faire**
  $y(i) \leftarrow 0$
  pour $j = 1$ à p **faire**
    $y(i) \leftarrow y(i) + a(i,j) * x(j)$
  fin pour
fin pour

- Le produit d'une matrice $n \times p$ par une matrice $p \times q$ coûte $\Theta(npq)$ flops, donc $\Theta(n^3)$ si les matrices sont carrées.

$$C = AB, \quad c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

pour $i = 1$ à n **faire**
  pour $j = 1$ à q **faire**
    $c(i,j) \leftarrow 0$
    pour $k = 1$ à p **faire**
      $c(i,j) \leftarrow c(i,j) + a(i,k) * b(k,j)$
    fin pour
  fin pour
fin pour

5.3 Représentations informatiques des matrices

Dans la section précédente, pour calculer la complexité des algorithmes sur les matrices, nous avons négligé le temps nécessaire pour accéder aux données, en ne comptant que les opérations arithmétiques. Cette approximation est valide tant que deux hypothèses sont vérifiées :

1. La structure de données est à accès aléatoire, c'est à dire que l'accès aux données se fait en temps constant. Par exemple le temps d'accès à un élément a_{ij} d'une matrice A de taille $n \times n$ ne doit pas dépendre de la valeur de n, ni de i, ni de j.
2. Les données tiennent en mémoire centrale (mémoire vive, RAM = Random Access Memory). Ceci dépend essentiellement de la taille des données. Nous supposerons qu'elles sont suffisamment petites pour que cette deuxième hypothèse soit vérifiée.

Quand les matrices sont pleines (*dense* en anglais), les matrices sont habituellement codées dans des tableaux 1D ou 2D. Ceci assure un accès aléatoire, car l'adresse de a_{ij} se calcule directement à partir de i, j, et n.

Dans de nombreux problèmes les matrices contiennent beaucoup d'éléments nuls. On dit qu'elles sont creuses (*sparse* en anglais). On peut alors économiser de la place en ne stockant pas les éléments nuls (on parle de "formats creux"), ce qui conduit à des algorithmes plus rapides mais souvent plus complexes. Supposons que A, de taille $n \times n$ n'ait que m éléments non nuls, m étant petit devant n^2. L'idée est de stocker m éléments (seulement les $a_{ij} \neg 0$) au lieu de n^2. en contrepartie, il peut être nécessaire de stocker les indices i et j, et il arrive que l'on perde la propriété d'accès aléatoire.

Chapitre 6

Résolution de systèmes linéaires

Les systèmes linéaires apparaissent souvent en informatique, en mathématiques, et en physique. Les problèmes d'optimisation, par exemple, utilisent beaucoup de résolutions de systèmes linéaires.

Il existe des méthodes variées pour résoudre un système linéaire, selon la forme de la matrice (pleine, creuse, symétrique, etc.), selon qu'on veut résoudre le problème une seule ou plusieurs fois, selon qu'on a besoin d'un résultat précis, rapide ou stable. C'est un sujet important d'analyse numérique. Nous revoyons ici la méthode de Gauss, vue en Algèbre 1.

6.1 Position du problème

On veut résoudre le système d'équations linéaires de n équations à n inconnues suivant :

$$\begin{cases} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & & & & & \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{cases} \tag{6.1}$$

Il s'écrit sous forme matricielle $AX = B$ avec

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

- A est une matrice (n lignes et n colonnes) contenant les coefficients du système linéaire.
- X est le vecteur des inconnues.
- B est le second membre, ou membre de droite.

On suppose que le système est inversible, c'est à dire que $det(A) \neq 0$, c'est à dire que $rang(A) = n$, c'est à dire qu'il possède une seule solution.

Dans tous les algorithmes, on supposera que A, X et B sont des tableaux passés en paramètre par variable (donc modifiés par effet de bord).

On note la i-ème équation L_i : $\sum_{j=1}^{n} a_{ij}x_j = b_j$.

6.2 Exemple

On résout le système

$$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 2 & -3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ -10 \end{bmatrix} \quad \Leftrightarrow \quad \left\{ \begin{array}{rcrcrcr} x_1 & - & x_2 & + & 2x_3 & = & 5 \\ 3x_1 & + & 2x_2 & + & x_3 & = & 10 \\ 2x_1 & - & 3x_2 & - & 2x_3 & = & -10 \end{array} \right.$$

$$\Leftrightarrow \quad \begin{bmatrix} 1 & -1 & 2 \\ 0 & 5 & -5 \\ 0 & -1 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \\ -20 \end{bmatrix} \qquad \text{par} \left\{ \begin{array}{ll} L_2 & \leftarrow L_2 - 3L_1 \\ L_3 & \leftarrow L_3 - 2L_1 \end{array} \right.$$

$$\Leftrightarrow \quad \begin{bmatrix} 1 & -1 & 2 \\ 0 & 5 & -5 \\ 0 & 0 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -5 \\ -21 \end{bmatrix} \qquad \text{par } L_3 \leftarrow L_3 + \frac{1}{5} L_2$$

On en déduit le valeurs de x_3 puis x_2 puis x_1 :

$$\begin{aligned} x_3 &= \frac{-21}{-7} = 3 \\ x_2 &= \frac{-5 + 5x_3}{5} = 2 \\ x_1 &= \frac{5 + x_2 - 2x_3}{1} = 1 \end{aligned}$$

On a mis le sytème sous forme triangulaire (avec des 0 sous la diagonale), puis on a calculé les x_i en “remontant”.

6.3 Cas des systèmes triangulaires

On se donne le système triangulaire supérieur

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

La solution s'obtient facilement par une remontée : calcul de x_n avec la ligne n, puis de x_{n-1} avec la ligne $n-1$, etc :

$$
\begin{aligned}
x_n &= \frac{b_n}{a_n} \\
x_{n-1} &= \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} \\
x_{n-2} &= \frac{b_{n-2} - a_{n-2,n}x_n - a_{n-2,n-1}x_{n-1}}{a_{n-2,n-2}} \\
&\vdots \\
x_i &= \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j \right) / a_{i,i} \\
&\vdots \\
x_1 &= \left(b_1 - \sum_{j=2}^{n} a_{1j}x_j \right) / a_{1,1}
\end{aligned}
$$

ALGO. **6.1** $remontee(A, B, X)$. Algorithme de remontée.

pour i descendant de n à 1 **faire**
  $X(i) \leftarrow B(i)$
  pour j de $i+1$ à n **faire**
    $X(i) \leftarrow X(i) - A(i,j)X(j)$
  fin pour
  $X(i) \leftarrow X(i)/A(i,i)$
fin pour

Exercice 6.1
Écrire l'algorithme de descente pour résoudre un système triangulaire inférieur.

Remarque. Cet algorithme suppose que tous les éléments diagonaux (a_{ii}) sont non nuls. Ceci est vrai car $det(A) = \prod_{i=1}^{n} a_{ii} \neq 0$ (voir le chapitre sur les déterminants).

6.4 Méthode du pivot de Gauss

L'objectif est de transformer un système *quelconque* en système *triangulaire supérieur* équivalent, puis de faire une remontée. Pour cela partons du système $AX = B$ (6.1).

Nous aurons besoin d'un algorithme qui additionne à une équation du multiple d'une autre : $L_i \leftarrow L_i + c * L_j$ (ALGO. (6.2)).

ALGO. **6.2** $addmultiple(A, B, i, j, c)$ Algorithme d'addition d'un multiple $L_i \leftarrow L_i + c * L_j$.

pour k de 1 à n **faire**
  $A(i,k) \leftarrow A(i,k) + c * A(j,k)$
fin pour
$B(i) \leftarrow B(i) + c * B(j)$

Pour obtenir un système triangulaire supérieur, on procède en $n-1$ étapes qui mettent des 0 sous la diagonale grâce à des combinaisons de lignes :

- Première étape : on fait apparaître des 0 dans la première colonne de A sous la diagonale. En supposant que $a_{11} \neq 0$, il faut soustraire $\frac{a_{j1}}{a_{11}} L_1$ à L_j, pour $2 \leq j \leq n$.
  Ceci donne l'algorithme
  pour j de 2 à n **faire**
  $addmultiple\,(A, B, j, 1, -A(j,1)/A(1,1))$
  fin pour
- Deuxième étape : on fait apparaître des 0 dans la deuxième colonne de A sous la diagonale. Ceci donne l'algorithme
  pour j de 3 à n **faire**
  $addmultiple\,(A, B, j, 2, -A(j,2)/A(2,2))$
  fin pour
- Et ainsi de suite, jusqu'à la colonne $n-1$. L'étape i fait apparaître des 0 sous la diagonale dans la colonne i :
  pour j de $i+1$ à n **faire**
  $addmultiple\,(A, B, j, i, -A(j,i)/A(i,i))$
  fin pour

À la i-ème étape on divise par $A(i,i)$, appelé "pivot". Donc il faut que $A(i,i) \neq 0$. Ce n'est pas toujours le cas, mais il existe toujours une valeur non nulle parmi $A(i,i), A(i+1,i), \ldots, A(n,i)$ (car A est inversible). Donc si $A(i,i)$ est nul, on échange d'abord L_i avec L_j tel que $j > i$ et $A(j,i) \neq 0$, de façon à ce que le pivot soit non nul. Il nous faut donc deux autres algorithmes élémentaires :

1. La recherche du premier $j \geq i$ tel que $A(j,i) \neq 0$ (ALGO. (6.3)).
2. L'échange de deux équations L_i et L_j (ALGO. (6.4)).

ALGO. **6.3** $choixPivot(A, i)$ Algorithme de recherche du premier $j \geq i$ tel que $A(j,i) \neq 0$.

pour j de i à n **faire**
  si $A(j,i) \neq 0$ **alors**
    renvoyer j
  fin si
fin pour
renvoyer (-1)

ALGO. **6.4** $echangeLigne(A, B, i, j)$ Algorithme d'échange de L_i et L_j.

pour k de 1 à n **faire**
  $tmp \leftarrow A(i,k)$
  $A(i,k) \leftarrow A(j,k)$
  $A(j,k) \leftarrow tmp$
fin pour
$tmp \leftarrow B(i)$
$B(i) \leftarrow B(j)$
$B(j) \leftarrow tmp$

On obtient donc l'algorithme qui rend le système triangulaire (ALGO. (6.5)). L'algorithme complet de résolution est ALGO. (6.6). L'analyse de complexité donne un nombre total d'opérations arithmétiques est $\Theta(n^3)$.

ALGO. **6.5** $triangulaire(A, B)$ Algorithme qui rend le système triangulaire.

pour i de 1 à $n-1$ **faire**
  $j \leftarrow choixPivot(A, i)$
  $echangeLigne(A, B, i, j)$
  pour j de $i+1$ à n **faire**
    $addmultiple\,(A, B, j, i, -A(j, i)/A(i, i))$
  fin pour
fin pour

ALGO. **6.6** $gauss(A, B, X)$ Algorithme résolution de $AX = B$.

$triangulaire(A, B)$
$remontee(A, B, X)$

6.5 Stabilité numérique

Dans la pratique, il se peut que le pivot $A(i, i)$ soit non nul mais très petit. Comme on divise par ce pivot pour rendre le système triangulaire, cela peut provoquer des erreurs numériques importantes.

Exemple 6.2
Supposons qu'on est en train de mettre 0 dans la colonne i à la ligne j : $L_j \leftarrow L_j - a_{ji}/a_{ii}L_i$. Donc à la colonne k :

$$a_{jk} \leftarrow a_{jk} - \frac{a_{ji}}{a_{ii}}a_{ik}$$

Supposons que a_{ii} soit 1000 fois plus petit que a_{ji} et que a_{ik} soit entâché d'une erreur ϵ, alors

$$a_{jk} - \frac{a_{ji}}{a_{ii}}(a_{ik} + \epsilon) = a_{jk} - \frac{a_{ji}}{a_{ii}}a_{ik} - 1000\epsilon$$

L'erreur sur a_{jk} a été multipliée par 1000.

ALGO. **6.7** $choixPivotPartiel(A, i)$ Algorithme de recherche du pivot partiel.

$p \leftarrow i$
$v \leftarrow abs(A(i, i))$
pour j de $i+1$ à n **faire**
  si $abs(A(j, i)) > v$ **alors**
    $p \leftarrow j$
    $v \leftarrow abs(A(j, i))$
  fin si
fin pour
renvoyer p

La méthode dite du *pivot partiel* permet de limiter ces erreurs (améliorer la stabilité numérique), en choisissant le plus grand pivot (en valeur absolue) de la colonne sous la diagonale (ALGO. (6.7)).

Il existe un algorithme dit du "pivot total" encore plus stable. Le pivot est choisi parmi les lignes *et* les colonnes entre i et n. Cet algorithme nécessite des manipulations sur les colonnes, ce qui modifie l'ordre des inconnues.

Dans la pratique on implémente toujours au moins le pivot partiel.

Exercice 6.3
Écrire l'algorithme du pivot total. Faire attention à l'ordre des inconnues.

Chapitre 7

Approximation au sens des moindres carrés

7.1 Position du problème

Au chapitre 6 nous avons étudié la résolution d'un système linéaire $A\mathbf{x} = \mathbf{b}$ carré et inversible. Il se peut que le système ne soit pas carré (par exemple plus/moins d'équations que d'inconnues) ou non inversible (pas de solution, ou plusieurs solutions). Nous nous intéressons ici au cas où le système n'a pas de solution exacte. Cette situation est fréquente quand il y a plus d'équations que d'inconnues.

Le système a m équations à n inconnues, c'est à dire :

- A est de taille $m \times n$
- le vecteur d'inconnues $\mathbf{x}$ est dans $\mathbb{R}^n$
- le membre de droite $\mathbf{b}$ est dans $\mathbb{R}^m$

Comme il n'y a pas de $\mathbf{x}$ tel que $A\mathbf{x} = \mathbf{b}$, nous allons chercher une solution $\hat{\mathbf{x}}$ telle que $A\hat{\mathbf{x}}$ se rapproche le plus possible de $\mathbf{b}$, c'est à dire telle que la distance $\|A\hat{\mathbf{x}} - \mathbf{b}\|$ est la plus petite possible.

On travaille dans $\mathbb{R}^n$ et $\mathbb{R}^m$ munis du produit scalaire classique et de la norme euclidienne.

Définition. Une *solution au sens des moindres carrés* à $A\mathbf{x} = \mathbf{b}$ est un $\hat{\mathbf{x}}$ tel que pour tout $\mathbf{x} \in \mathbb{R}^n$

$$\|A\hat{\mathbf{x}} - \mathbf{b}\| \leq \|A\mathbf{x} - \mathbf{b}\|$$

Remarques.

- On parle de *moindres carrés* car la norme utilisée est la norme quadratique, qui somme des carrés des erreurs.
- Il s'agit plus précisément de moindres carrés *linéaires* car le problème est linéaire en $\mathbf{x}$ (les inconnues).

7.2 Résolution par projection orthogonale

Notons $\mathbf{a_i}$ la i-ième colonne de $A = \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \cdots & \mathbf{a_n} \end{bmatrix}$

On remarque que, quel que soit $\mathbf{x}$, le vecteur $A\mathbf{x}$ appartient toujours au sous-espace de $\mathbb{R}^m$ engendré par les colonnes de A :

$$A\mathbf{x} = x_1\mathbf{a_1} + x_2\mathbf{a_2} + \ldots + x_n\mathbf{a_n} \qquad \in \mathcal{L}\{\mathbf{a_1}, \ldots, \mathbf{a_n}\} = Col(A)$$

On veut que $A\hat{\mathbf{x}}$ soit dans $Col(A)$ le vecteur le plus proche de $\mathbf{b}$: c'est la projection orthogonale de $\mathbf{b}$ sur $Col(A)$. Cela implique que le vecteur $A\hat{\mathbf{x}} - \mathbf{b}$ est orthogonal à tous les vecteurs de $Col(A)$, c'est à dire à tous les $\mathbf{a_i}$:

$$\langle\ \mathbf{a_i}\ ,\ A\hat{\mathbf{x}} - \mathbf{b}\ \rangle = 0 \quad \text{pour tout } i \qquad \Leftrightarrow \qquad \mathbf{a_i}^T(A\hat{\mathbf{x}} - \mathbf{b}) = 0 \quad \text{pour tout } i$$

Comme $\mathbf{a_i}^T$ est une ligne de A^T, ceci équivaut à

$$A^T(A\hat{\mathbf{x}} - \mathbf{b}) = \mathbf{0} \quad \Leftrightarrow \quad A^TA\hat{\mathbf{x}} = A^T\mathbf{b}$$

Le système linéaire $A^TA\hat{\mathbf{x}} = A^T\mathbf{b}$ est appelé *équations normales* de $A\mathbf{x} = \mathbf{b}$. On montre que les solutions des équations normales coïncident avec l'ensemble des solutions au sens des moindres carrés de $A\mathbf{x} = \mathbf{b}$.

La matrice A^TA est symétrique et ses valeurs propres sont positives ou nulles. Si elle est inversible (pas de valeur propre nulle), alors il y a une unique solution au sens des moindres carrés : $\hat{\mathbf{x}} = \left(A^TA\right)^{-1}A^T\mathbf{b}$.

7.3 Résolution par minimisation de la distance quadratique

Voici un autre éclairage. Minimiser $\|A\mathbf{x} - \mathbf{b}\|$ est la même chose que minimiser son carré :

$$E(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|^2 = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = (\mathbf{x}^TA^T - \mathbf{b}^T)(A\mathbf{x} - \mathbf{b}) = \mathbf{x}^TA^TA\mathbf{x} - 2\mathbf{x}^TA^T\mathbf{b} + \mathbf{b}^T\mathbf{b}$$

La fonction E (de $\mathbb{R}^n$ dans $\mathbb{R}$) est convexe et son gradient est

$$\nabla E(\mathbf{x}) = 2A^TA\mathbf{x} - 2A^T\mathbf{b}$$

Les valeurs minimales sont atteintes quand le gradient s'annule, c'est à dire $\nabla E(\mathbf{x}) = \mathbf{0}$. On retrouve les équations normales.

7.4 Régression linéaire

Supposons qu'on mesure expérimentalement des couples de valeurs (x_i, y_i). On observe que ces points sont à peu près alignées sur une droite. Quelle droite est la plus proche des mesures ?

Donnons nous la droite d'équation $y = \alpha x + \beta$. Cette droite est-elle proche des observations ?

Pour chaque x_i, on dispose d'une valeur *observée* y_i, et d'une valeur *prédite* $\alpha x_i + \beta$. La différence entre ces deux valeurs est appelée *résidu*. Quelle est l'erreur entre les valeurs prédites et les valeurs observées ?

Une façon courante de calculer cette erreur est de faire la somme des carrés des résidus :

$$E(\alpha, \beta) = \sum_{i=1}^{m} \left((\alpha x_i + \beta) - y_i\right)^2$$

La droite qui minimise cette erreur est appelée *droite des moindres carrés*, ou *droite de régression*. Comment la calculer ? C'est-à-dire comment calculer α et β ?

Idéalement, on voudrait que les données prédites soient égales aux données observées :

$$\begin{cases} \alpha x_1 + \beta = y_1 \\ \alpha x_2 + \beta = y_2 \\ \vdots \\ \alpha x_m + \beta = y_m \end{cases} \quad \Leftrightarrow \quad \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_m & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \Leftrightarrow \quad A \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{y} \tag{7.1}$$

Ce système n'a généralement pas de solution exacte, mais il a une solution au sens des moindres carrés. Cette solution coïncide avec la droite des moindres carrés, qui minimise

$$E\left(\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right) = \left\| A\begin{bmatrix}\alpha\\ \beta\end{bmatrix} - \mathbf{y}\right\|^2$$

Bilan. Pour trouver une droite des moindres carrés il faut :
- Écrire le système (7.1).
- Écrire les équations normales.
- Résoudre les équations normales pour trouver α et β.

Remarque. On parle de régression *linéaire* car l'équation d'une droite est linéaire en x. Il ne faut pas confondre avec la linéarité en α et β (les inconnues) : des régressions *non-linéaires* peuvent se résoudre par des moindres carrés *linéaires*.

Exercice 7.1
Calculer la droite des moindres carrés pour les observations $(2,1)$, $(5,2)$, $(7,3)$ et $(8,3)$.

Exercice 7.2
Reprendre l'exemple 9.6 : simuler l'évolution des populations pour un $\mathbf{u}_0$ quelconque, puis faire une regression sur R_k en fonction de C_k. On doit retrouver le ratio rats/chouettes d'un vecteur propre. Observer le résultat de la régression pour des périodes plus ou moins longues. Essayer aussi en négligeant les premiers mois.

Faire deux régressions linéaires séparées sur C_k (fonction de C_{k-1}) et sur R_k (fonction de R_{k-1}). On doit retrouver le taux de croissance de 2%.

7.5 Approximation polynômiale

Dans la section précédente, nous avons approximé les données $(\mathbf{x}, \mathbf{y})$ par une droite. Si les données ne sont pas allignées sur une droite, mais sur une courbe plus complexe, il est possible de les approximer par une fonction plus complexe, par exemple un polynôme $P(x)$, en faisant le même raisonnement. Cherchons un polynôme P de degré $k < n$ tel que $P(x_i) \approx y_i$ pour tout i. La question essentielle est de définir ce que signifie "$P(x_i) \approx y_i$ pour tout i". On définit l'erreur quadratique comme

$$E = \sum_{i=1}^{n} (P(x_i) - y_i)^2$$

On cherche le polynôme P qui minimise E. Pour cela, écrivons P dans la base des monômes :

$$P(x) = c_0 + c_1 x + \ldots + c_j x^j + \ldots + c_k x^k = [1, x, x^2, \ldots, x^k] \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_k \end{bmatrix}$$

Donc

$$A\mathbf{c} - \mathbf{y} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^k \\ 1 & x_2 & x_2^2 & \ldots & x_2^k \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^k \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_k \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} P(x_1) - y_1 \\ P(x_2) - y_2 \\ \vdots \\ P(x_n) - y_n \end{bmatrix}$$

D'où la reformulation de l'erreur

$$E(\mathbf{c}) = \|A\mathbf{c} - \mathbf{y}\|^2,$$

dont le minimum est atteint pour $\mathbf{c}$ vérifiant les équations normales

$$A^T A\mathbf{c} = A^T\mathbf{y}.$$

Les équations normales sont un système linéaire, de taille $(k+1)\times(k+1)$ qui est symétrique semi-défini positif.

7.6 Moindres carrés pondérés, une question de confiance

Supposons maintenant qu'on a plus confiance dans certains points que dans d'autres. On donne un poids $0 \leq w_i \leq 1$ à chaque point (x_i, y_i). Ce poids est un indice de confiance :

- $w_i = 0$ signifie qu'on n'a pas du tout confiance en (x_i, y_i).
- $w_i = 1$ signifie qu'on a une grande confiance en (x_i, y_i).

On ré-écrit l'erreur en accordant plus d'importance aux points avec un fort indice de confiance :

$$E = \sum_{i=1}^{n} w_i^2 (P(x_i) - y_i)^2 = \|WAC - WY\|^2$$

avec W matrice $n \times n$ diagonale :

$$W = \begin{bmatrix} w_1 & & 0 & \\ & w_2 & & \\ & & \ddots & \\ 0 & & & w_n \end{bmatrix}$$

Les équations normales s'écrivent donc :

$$(WA)^T(WA)C = (AW)^T WY \quad \Leftrightarrow \quad A^T W W A C = A^T W W Y$$

7.7 Changements de variable

Pour analyser le lien entre $\mathbf{x}$ et $\mathbf{y}$, il peut être utile de faire un changement de variable : on observe le lien entre $\mathbf{x}'$ et $\mathbf{y}'$, obtenues depuis $\mathbf{x}$ et $\mathbf{y}$ par un changement de variable.

Exemple 7.3

Si l'on pense que nos variables sont liées par un loi exponentielle $\mathbf{y} = \alpha^{\mathbf{x}}$, il est utile d'observer

$$\mathbf{y}' = \log(\mathbf{y}) \text{ en fonction de } \mathbf{x}' = \mathbf{x}$$

car la relation devient linéaire :

$$\mathbf{y}' = \log(\alpha^{\mathbf{x}}) = \log(\alpha)\mathbf{x} = \log(\alpha)\mathbf{x}'.$$

Une régression linéaire sur $(\mathbf{x}', \mathbf{y}')$ permet d'estimer le coefficient d'exponentiation α.

Cette technique revient à observer $(\mathbf{x}, \mathbf{y})$ en échelle semi-logarithmique, très classique dans les outils de visualisation. On pourrait dire que cette échelle "transforme les exponentielles en droites".

Exemple 7.4

Si l'on pense que nos variables sont liées par un loi de puissance $\mathbf{y} = \mathbf{x}^p$, il est utile d'observer

$$\mathbf{y}' = \log(\mathbf{y}) \text{ en fonction de } \mathbf{x}' = \log(\mathbf{x})$$

car la relation devient linéaire :

$$\mathbf{y}' = \log(\mathbf{x}^p) = p\log(\mathbf{x}) = p\mathbf{x}'.$$

Une régression linéaire sur $(\mathbf{x}', \mathbf{y}')$ permet d'estimer le puissance p.

Cette technique revient à observer $(\mathbf{x}, \mathbf{y})$ en échelle logarithmique, dite "log-log", elle aussi très classique dans les outils de visualisation.

Chapitre 8

Applications linéaires

8.1 Notations

Soient $\mathbf{E}$ un espace vectoriel sur $\mathbb{R}$, muni d'une base $e = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. Tout vecteur $\mathbf{u}$ est représenté de manière unique dans cette base par son vecteur de coordonnées dans $\mathbb{R}^n$ noté

$$\mathbf{u}^{(e)} = \begin{bmatrix} u_1^{(e)} \\ \vdots \\ u_n^{(e)} \end{bmatrix} \quad \text{tel que} \quad \mathbf{u} = \sum_{i=1}^{n} u_i^{(e)} \mathbf{e}_i$$

Attention : dans beaucoup d'exemples $\mathbf{E} = \mathbb{R}^n$ et les vecteurs se confondent avec leurs coordonnées dans la base canonique.

Exemple 8.1

Soit $\mathbf{E} = \mathbb{R}^2$ muni de sa base canonique $e = \left\{ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$.

Le vecteur $\mathbf{u} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \in \mathbf{E}$ s'écrit $\mathbf{u} = 2\mathbf{e}_1 + (-1)\mathbf{e}_2$ et donc les coordonnées $\mathbf{u}^{(e)} = \begin{bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}^{(e)}$ sont confondues avec le vecteur $\mathbf{u}$ lui-même.

8.2 Espaces vectoriels et espaces affines

On fait habituellement de la géométrie dans un espace *affine*, qui dispose de points en plus des vecteurs. On y définit des *repères*, constitués d'une base de vecteurs et d'une origine. Il est alors possible de définir des applications affines (par exemple les translations), des droites passant par des points, etc.

Dans un espace *vectoriel*, on ne dispose que de vecteurs. On peut y définir des applications linéaires, mais pas affines. Il existe des droites et des plans dits *vectoriels*, qui sont des sous-espaces vectoriels. On les représente passant par l'origine.

Exemple 8.2

Soit $\mathcal{D}$ une droite vectorielle, de vecteur directeur $\mathbf{d}$.
Par définition, c'est le sous-espace engendré par $\mathbf{d}$: $\mathcal{D} = \mathcal{L}\{\mathbf{d}\}$.

8.3 Applications linéaires

Définition. Soient $\mathbf{E}$ et $\mathbf{F}$ deux espaces vectoriels sur $\mathbb{R}$. Une application l de $\mathbf{E}$ dans $\mathbf{F}$ est dite linéaire si

- $l(\mathbf{u}+\mathbf{v}) = l(\mathbf{u}) + l(\mathbf{v})$ pour tout $\mathbf{u}, \mathbf{v} \in \mathbf{E}$.
- $l(c\mathbf{u}) = c\, l(\mathbf{u})$ pour tout $\mathbf{u} \in \mathbf{E}$ et $c \in \mathbb{R}$.

Remarque. On en déduit que $l(\mathbf{0}) = \mathbf{0}$

Représentation dans une base.

Soient n et p les dimensions respectives de $\mathbf{E}$ et $\mathbf{F}$. Soient $e = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ une base de $\mathbf{E}$, et $f = \{\mathbf{f}_1, \ldots, \mathbf{f}_p\}$ une base de $\mathbf{F}$. Une application linéaire $l : \mathbf{E} \to \mathbf{F}$ s'écrit

$$l(\mathbf{u}) = l\left(\sum_{i=1}^{n} u_i^{(e)} \mathbf{e}_i\right) = \sum_{i=1}^{n} u_i^{(e)} l(\mathbf{e}_i) = \sum_{i=1}^{n} u_i^{(e)} \left(\sum_{j=1}^{p} l_{ji} \mathbf{f}_j\right) = \sum_{j=1}^{p} \left(\sum_{i=1}^{n} u_i^{(e)} l_{ji}\right) \mathbf{f}_j = \sum_{j=1}^{p} v_j^{(f)} \mathbf{f}_j$$

Notons L la matrice des l_{ji} (de taille $p \times n$). Si on représente
$\mathbf{u} \in \mathbf{E}$ par ses coordonnées $\mathbf{u}^{(e)} \in \mathbb{R}^n$ dans la base e, et
$\mathbf{v} = l(\mathbf{u}) \in \mathbf{F}$ par ses coordonnées $\mathbf{v}^{(f)} \in \mathbb{R}^p$ dans la base f, l'application l s'écrit $l^{(e,f)} : \mathbb{R}^n \to \mathbb{R}^p$

$$\mathbf{v}^{(f)} = l^{(e,f)}(\mathbf{u}^{(e)}) = L\mathbf{u}^{(e)} = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1n} \\ l_{21} & l_{22} & \ldots & l_{2n} \\ \vdots & & & \vdots \\ l_{p1} & \ldots & \ldots & l_{pn} \end{bmatrix} \begin{bmatrix} u_1^{(e)} \\ \vdots \\ u_n^{(e)} \end{bmatrix} = \begin{bmatrix} v_1^{(f)} \\ \vdots \\ v_p^{(f)} \end{bmatrix}$$

Remarques :

- Ne pas confondre l et $l^{(e,f)}$.
- Attention, la matrice L dépend des bases e et f.
- La $i-$ème colonne de L contient les coordonnées (dans la base f) l'image $l(\mathbf{e}_i)$ du vecteur $\mathbf{e}_i$:

$$l^{(e,f)}(\mathbf{e}_i^{(e)}) = l^{(e,f)}\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^{(e)}\right) = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1n} \\ l_{21} & l_{22} & \ldots & l_{2n} \\ \vdots & & & \vdots \\ l_{p1} & \ldots & \ldots & l_{pn} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^{(e)} = \begin{bmatrix} l_{1i} \\ l_{2i} \\ \vdots \\ l_{pi} \end{bmatrix}^{(f)}$$

- Donc $l(\mathbf{u})$ dans la base f est une combinaison linéaire des colonnes de L :

$$l^{(e,f)}(\mathbf{u}^{(e)}) = \sum_{i=1}^{n} u_i^{(e)} l(\mathbf{e}_i^{(e)}) = \sum_{i=1}^{n} u_i^{(e)} \begin{bmatrix} l_{1i} \\ l_{2i} \\ \vdots \\ l_{pi} \end{bmatrix}^{(f)} . \tag{8.1}$$

Exemple 8.3 projection sur une droite dans $\mathbb{R}^2$.

On se place dans $\mathbb{R}^2$ muni du produit scalaire euclidien. Soit $\mathcal{D} = \mathcal{L}\{\mathbf{d}\}$ une droite vectorielle, de vecteur directeur $\mathbf{d}$ unitaire ($\|\mathbf{d}\| = 1$). La projection orthogonale sur $\mathcal{D}$ est l'application :

$$\begin{aligned} p : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \mathbf{u} &\mapsto p(\mathbf{u}) = \langle\, \mathbf{u}\, ,\, \mathbf{d}\, \rangle\, \mathbf{d} \end{aligned}$$

Prenons $\mathbf{d} = \dfrac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. Dans la base canonique :

$$p(\mathbf{u}) = \left\langle \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} , \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\rangle \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \frac{1}{5}(2u_1 + u_2) \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 4u_1 + 2u_2 \\ 2u_1 + u_2 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Donc $L = \frac{1}{5}\begin{bmatrix}4 & 2\\ 2 & 1\end{bmatrix}$.
La projection de $\begin{bmatrix}1\\3\end{bmatrix}$ est $p\left(\begin{bmatrix}1\\3\end{bmatrix}\right) = \frac{1}{5}\begin{bmatrix}4 & 2\\ 2 & 1\end{bmatrix}\begin{bmatrix}1\\3\end{bmatrix} = \frac{1}{5}\begin{bmatrix}10\\5\end{bmatrix} = \begin{bmatrix}2\\1\end{bmatrix}$

Exemple 8.4 projection sur une droite dans $\mathbb{R}^3$.
Identique à la projection dans $\mathbb{R}^2$.

Exemple 8.5 projection sur un plan dans $\mathbb{R}^3$.
On se place dans $\mathbb{R}^3$ muni du produit scalaire euclidien. Soit $\mathcal{P}$ un plan vectoriel de normale $\mathbf{n}$ unitaire. Donc $\langle\ \mathbf{u}\ ,\ \mathbf{n}\ \rangle = 0$ pour tout $\mathbf{u} \in \mathcal{P}$, et réciproquement. La projection orthogonale sur $\mathcal{P}$ est l'application $p : \mathbb{R}^3 \to \mathbb{R}^3$:

$$p(\mathbf{u}) = \mathbf{u} - \langle\ \mathbf{u}\ ,\ \mathbf{n}\ \rangle\, \mathbf{n}$$

Le projeté $p(\mathbf{u})$ appartient à $\mathcal{P}$, car $\langle\ p(\mathbf{u})\ ,\ \mathbf{n}\ \rangle = 0$ (exercice).
La projection est une application linéaire (exercice) :

- Pour tout $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$, $p(\mathbf{u}+\mathbf{v}) = p(\mathbf{u}) + p(\mathbf{v})$.
- Pour tout $\mathbf{u} \in \mathbb{R}^3$ et $c \in \mathbb{R}$, $p(c\mathbf{u}) = cp(\mathbf{u})$.

Noyau.
Le noyau (kernel) de l, noté $Ker(l)$, est le sous-espace vectoriel de $\mathbf{E}$ des vecteurs d'image nulle :

$$Ker(l) = \{\mathbf{u} \in \mathbf{E} / l(\mathbf{u}) = \mathbf{0}\}$$

Image.
L'image de l, noté $Im(l)$ est le sous-espace vectoriel de $\mathbf{F}$ des vecteurs image :

$$Im(l) = \{\mathbf{v} \in \mathbf{F} / \exists \mathbf{u} \in \mathbf{E}, \mathbf{v} = l(\mathbf{u})\}$$

Remarques :

- Théorème du rang : $dim(Ker(l)) + dim(Im(l)) = dim(\mathbf{E}) = n$.
- $Im(l)$ est l'espace engendré par les colonnes de L, vues comme des vecteurs de coordonnées dans la base f (confer Eq. (8.1)) :

$$Im(l) = \mathcal{L}\left\{\begin{bmatrix}l_{1,1}\\ \vdots \\ l_{p,1}\end{bmatrix}^{(f)}, \ldots, \begin{bmatrix}l_{1,n}\\ \vdots \\ l_{p,n}\end{bmatrix}^{(f)}\right\}$$

Exemple 8.6 projection sur une droite dans $\mathbb{R}^2$.
Quels sont le noyau et l'image de la projection de l'exemple 8.3 ?
Attention ici $\mathbf{E} = \mathbf{F} = \mathbb{R}^2$ sont confondus.

$$\mathbf{u} \in Ker(p) \Leftrightarrow p(\mathbf{u}) = \mathbf{0} \Leftrightarrow \langle\ \mathbf{u}\ ,\ \mathbf{d}\ \rangle = 0$$

Donc $Ker(p)$ est l'ensemble des vecteurs orthogonaux à $\mathbf{d}$. C'est une droite vectorielle dans $\mathbf{E}$: *la* droite orthogonale à $\mathcal{D}$. Sur l'exemple numérique

$$\begin{bmatrix}u_1\\u_2\end{bmatrix} \in Ker(p) \quad \Leftrightarrow \quad \left\langle \begin{bmatrix}u_1\\u_2\end{bmatrix}\ ,\ \begin{bmatrix}2\\1\end{bmatrix} \right\rangle = 0 \quad \Leftrightarrow \quad 2u_1 + u_2 = 0$$

L'image est l'espace (dans $\mathbf{F}$) engendré par les colonnes de L :

$$Im(l) = \mathcal{L}\left\{\frac{1}{5}\begin{bmatrix}4\\2\end{bmatrix}, \frac{1}{5}\begin{bmatrix}2\\1\end{bmatrix}\right\} = \mathcal{L}\left\{\begin{bmatrix}4\\2\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}\right\} = \mathcal{L}\left\{\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}\right\} = \mathcal{L}\left\{\begin{bmatrix}2\\1\end{bmatrix}\right\}$$

C'est la droite $\mathcal{D}$ elle-même (considérée dans $\mathbf{F}$).

Exemple 8.7 projection sur un plan dans $\mathbb{R}^3$.

Quels sont le noyau et l'image de la projection de l'exemple 8.5 ?
On a $\mathbf{u} \in Ker(p) \Leftrightarrow p(\mathbf{u}) = \mathbf{0} \Leftrightarrow \mathbf{u} = \langle\, \mathbf{u}\ ,\ \mathbf{n}\, \rangle\, \mathbf{n}$. Donc il faut que $\mathbf{u}$ et $\mathbf{n}$ soient colinéaires (condition nécessaire). C'est aussi une condition suffisante (exercice), donc $Ker(p) = \mathcal{L}\{\mathbf{n}\}$ est le sous-espace engendré par $\mathbf{n}$.

Lien avec la résolution de systèmes linéaires.
Dans le chapitre 6 nous cherchions à résoudre un système $AX = B$, avec X et B dans $\mathbb{R}^n$. Notons $l : \mathbb{R}^n \to \mathbb{R}^n$ l'application linéaire qui à X associe $l(X) = AX$. Le système s'écrit alors $l(X) = B$. Donc résoudre le système revient à chercher le(s) vecteur(s) dont l'image par l est B.

8.4 Changement de base

Soient $e = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ et $e' = \{\mathbf{e'}_1, \ldots, \mathbf{e'}_n\}$ deux bases d' un espace vectoriel $\mathbf{E}$. Tout vecteur $\mathbf{u} \in \mathbf{E}$ a des vecteurs coordonnées $\mathbf{u}^{(e)}$ et $\mathbf{u}^{(e')}$ dans ces bases. Il existe une unique matrice $P_{e\to e'}$, appelée matrice de passage de la base e dans la base e', telle que

$$\mathbf{u}^{(e')} = P_{e\to e'}\mathbf{u}^{(e)}$$

Propriétés.

- Les colonnes de cette matrice sont les coordonnées dans e' des vecteurs de la base e :

$$P_{e\to e'} = \left[\mathbf{e}_1^{(e')}, \mathbf{e}_2^{(e')}, \ldots, \mathbf{e}_n^{(e')}\right]$$

- La matrice $P_{e\to e'}$ est carrée et inversible.
- La matrice de passage de e' dans e est son inverse :

$$P_{e'\to e} = (P_{e\to e'})^{-1}$$

- La matrice $P_{e\to e'}$ représente l'identité (application linéaire) de $\mathbf{E}$ muni de e dans $\mathbf{E}$ muni de e'.

Exemple 8.8

Plaçons nous dans $\mathbb{R}^2$ muni des bases $e = \left\{\mathbf{e}_1 = \begin{bmatrix}1\\0\end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix}0\\1\end{bmatrix}\right\}$ et $e' = \left\{\mathbf{e'}_1 = \begin{bmatrix}2\\1\end{bmatrix}, \mathbf{e'}_2 = \begin{bmatrix}1\\-1\end{bmatrix}\right\}$ Ces bases peuvent être vues comme des grilles pour repérer les vecteurs de $\mathbb{R}^2$.
Les matrices de passage sont

$$P_{e'\to e} = \left[\mathbf{e'}_1^{(e)}, \mathbf{e'}_2^{(e)}\right] = \begin{bmatrix}2 & 1\\1 & -1\end{bmatrix} \quad \text{et} \quad P_{e\to e'} = \left[\mathbf{e}_1^{(e')}, \mathbf{e}_2^{(e')}\right] = \frac{1}{3}\begin{bmatrix}1 & 1\\1 & -2\end{bmatrix}$$

Le vecteur $\mathbf{u}$ a pour coordonnées $\mathbf{u}^{(e)} = \begin{bmatrix}3\\3\end{bmatrix}$ donc $\mathbf{u}^{(e')} = P_{e\to e'}\mathbf{u}^{(e)} = \frac{1}{3}\begin{bmatrix}1 & 1\\1 & -2\end{bmatrix}\begin{bmatrix}3\\3\end{bmatrix} = \begin{bmatrix}2\\-1\end{bmatrix}$.
Le vecteur $\mathbf{v}$ a pour coordonnées $\mathbf{v}^{(e')} = \begin{bmatrix}1\\1\end{bmatrix}$ donc $\mathbf{v}^{(e)} = P_{e'\to e}\mathbf{v}^{(e')} = \begin{bmatrix}2 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}3\\0\end{bmatrix}$.

Exemple 8.9

Reprenons l'exemple du chapitre 2. Dans $\mathbf{E} = \mathbb{R}^3$ muni de la base e

$$\mathbf{e}_1 = \begin{bmatrix}1\\2\\-2\end{bmatrix},\ \mathbf{e}_2 = \begin{bmatrix}1\\1\\0\end{bmatrix},\ \mathbf{e}_3 = \begin{bmatrix}0\\1\\-5\end{bmatrix}$$

et de sa base canonique e'.

$$P_{e\to e'} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 1 \\ -2 & 0 & -5 \end{bmatrix}$$

Le vecteur $\mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$ a pour coordonnées $\mathbf{u}^{(e)} = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$ et $\mathbf{u}^{(e')} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$.

On vérifie que

$$\mathbf{u}^{(e')} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = P_{e\to e'}\mathbf{u}^{(e)} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 1 \\ -2 & 0 & -5 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$$

Changement de base dans les application linéaires.
On peut maintenant exprimer une même application linéaire dans différentes bases. Soit $l : \mathbf{E} \to \mathbf{F}$, avec les bases e, e', f et f'. On peut exprimer l de e' dans f' par

- $l^{(e',f')} = Id^{(f,f')} \circ l^{(e,f)} \circ Id^{(e',e)}$
- $L^{(e',f')} = P_{f\to f'} L^{(e,f)} P_{e'\to e}$

Exemple 8.10 projection sur une droite dans $\mathbb{R}^2$.
Prolongeons l'exemple 8.3 et 8.6. Soit e la base canonique de $\mathbb{R}^2$ et $e' = \left\{\mathbf{e'}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathbf{e'}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}\right\}$. On a

$$P_{(e'\to e)} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \quad \text{et} \quad P_{(e\to e')} = \frac{1}{5}\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$$

l est la projection sur la droite vectorielle $\mathcal{D} = \mathcal{L}\left\{\mathbf{d} = \frac{1}{\sqrt{5}}\begin{bmatrix} 2 \\ 1 \end{bmatrix}\right\}$. Soit $\mathbf{v} = l(\mathbf{u})$. Écrivons l dans différentes bases, en l'illustrant avec

$$\mathbf{u} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \text{donc} \quad \mathbf{u}^{(e)} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}^{(e)} \quad \text{et} \quad \mathbf{u}^{(e')} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}^{(e)}$$

- De e dans e : on sait que $\mathbf{v}^{(e)} = l^{(e,e)}(\mathbf{u}^{(e)}) = L^{(e,e)}\mathbf{u}^{(e)}$

  $$L^{(e,e)} = \frac{1}{5}\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}$$

  Par exemple

  $$l^{(e,e)}\left(\begin{bmatrix} 1 \\ 3 \end{bmatrix}^{(e)}\right) = \frac{1}{5}\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix}^{(e)} = \frac{1}{5}\begin{bmatrix} 10 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^{(e)}$$

- De e' dans e' : $\mathbf{v}^{(e')} = l^{(e',e')}(\mathbf{u}^{(e')}) = L^{(e',e')}\mathbf{u}^{(e')}$ avec

  $$L^{(e',e')} = P_{e\to e'} L^{(e,e)} P_{e'\to e} = \frac{1}{25}\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

  Par exemple

  $$l^{(e',e')}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}^{(e')}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}^{(e')} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{(e')}$$

- De e dans e' : $\mathbf{v}^{(e')} = l^{(e,e')}(\mathbf{u}^{(e)}) = L^{(e,e')}\mathbf{u}^{(e)}$ avec

  $$L^{(e,e')} = P_{e\to e'} L^{(e,e)} = \frac{1}{25}\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}$$

Qui s'écrit aussi

$$L^{(e,e')} = L^{(e',e')} P_{e \to e'} = \frac{1}{5} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$$

Par exemple

$$l^{(e,e')} \left(\begin{bmatrix} 1 \\ 3 \end{bmatrix}^{(e)} \right) = \frac{1}{5} \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix}^{(e)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{(e')}$$

- De e' dans e : $\mathbf{v}^{(e)} = l^{(e',e)}(\mathbf{u}^{(e')}) = L^{(e',e)} \mathbf{u}^{(e')}$ avec

$$L^{(e',e)} = P_{e' \to e} L^{(e',e')} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix}$$

Par exemple

$$l^{(e',e)} \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}^{(e')} \right) = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^{(e')} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^{(e)}$$

Chapitre 9

Éléments propres

9.1 Définitions

Vecteurs propres et valeurs propres.
Soit A une matrice réelle carrée de taille $n \times n$.
Un vecteur propre $\mathbf{v}$ associé à une valeur propre λ (scalaire) est un vecteur non nul tel que $A\mathbf{v} = \lambda\mathbf{v}$.

Cette équation se réécrit $(A - \lambda I)\mathbf{v} = \mathbf{0}$.
Par définition ce système admet une solution $\mathbf{v} \neq \mathbf{0}$.
Comme $\mathbf{0}$ est aussi solution, ce système admet plusieurs solutions.
Donc $(A - \lambda I)$ n'est pas inversible.
Donc $det(A - \lambda I) = 0$.

Polynôme caractéristique.
$P_A(\lambda) = det(A - \lambda I)$ est un polynôme en λ de degré n, appelé polynôme caractéristique de A.
Les valeurs propres de A sont les racines du polynôme caractéristique.

Attention : même si A est réelle, il se peut que certaines valeurs propres soient complexes.

Exemple 9.1
Prenons $A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$, on a

$$det(A - \lambda I) = \begin{vmatrix} 3-\lambda & -2 \\ 1 & -\lambda \end{vmatrix} = (3-\lambda)(-\lambda) + 2 = \lambda^2 - 3\lambda + 2 = (\lambda - 1)(\lambda - 2)$$

Donc A a deux valeurs propres $\lambda = 2$ et $\lambda = 1$.

Remarque.
Cette méthode de calcul des valeurs propres (calculer le polynôme caractéristique et en trouver les racines) ne fonctionne que pour les petites matrices, car il n'existe pas de formule pour trouver les racines d'un polynôme de degré n supérieur ou égal à 5.

9.2 Calcul de l'espace propre associé à une valeur propre réelle connue

Espace propre.
Soit λ une valeur propre de A. L'espace propre associé à λ est l'ensemble E_λ de tous les vecteurs

propres associés à cette valeur propre :

$$E_\lambda = \{\mathbf{v} \in \mathbb{R}^n \ / \ A\mathbf{v} = \lambda\mathbf{v}\}$$

C'est un sous-espace vectoriel de $\mathbb{R}^n$.

Soit λ une valeur propre réelle de A. Posons $B_\lambda = A - \lambda I$. L'espace propre E_λ est donc l'ensemble des solutions du système linéaire $B_\lambda \mathbf{v} = \mathbf{0}$. Nous allons calculer E_λ en deux étapes.

La **première étape** consiste à mettre B_λ sous forme échelonnée $\widetilde{B}_\lambda$, c'est à dire de la forme

$$\widetilde{B}_\lambda = \begin{bmatrix} \mathbf{x} & \times & \times & \times & \times & \times \\ 0 & 0 & \mathbf{x} & \times & \times & \times \\ \vdots & & 0 & \mathbf{x} & \times & \times \\ \vdots & & & 0 & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 \end{bmatrix} \quad \text{telle que} \quad \widetilde{B}_\lambda \mathbf{v} = \mathbf{0} \ \Leftrightarrow \ B_\lambda \mathbf{v} = \mathbf{0}$$

Le premier élément de chaque échelon (en gras) est appelé pivot.
Cette transformation se fait grâce à des opérations sur les lignes, par un algorithme similaire à celui de mise sous forme triangulaire des systèmes inversibles.

Exemple 9.2

Reprenons $A = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}$ avec $\lambda = 1$, donc $B_1 = \begin{bmatrix} 2 & -2 \\ 1 & -1 \end{bmatrix}$. Alors

$$B_1\mathbf{v} = \mathbf{0} \quad \Leftrightarrow \quad \begin{bmatrix} 2 & -2 \\ 1 & -1 \end{bmatrix} \mathbf{v} = \mathbf{0} \quad \Leftrightarrow \quad \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix} \mathbf{v} = \mathbf{0} \qquad \text{par } L_2 \leftarrow L_2 - \frac{1}{2} L_1$$

Donc $\widetilde{B}_1 = \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix}$.

Exemple 9.3

$A = \begin{bmatrix} 4 & -1 & 6 \\ 2 & 1 & 6 \\ 2 & -1 & 8 \end{bmatrix}$ admet $\lambda_1 = 9$ et $\lambda_2 = 2$ comme valeurs propres.

$$B_9 = \begin{bmatrix} -5 & -1 & 6 \\ 2 & -8 & 6 \\ 2 & -1 & -1 \end{bmatrix} \quad \text{devient} \quad \begin{bmatrix} -5 & -1 & 6 \\ 0 & -42 & 42 \\ 0 & -7 & 7 \end{bmatrix} \quad \text{devient} \quad \begin{bmatrix} -5 & -1 & 6 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \widetilde{B}_9$$

$$B_2 = \begin{bmatrix} 2 & -1 & 6 \\ 2 & -1 & 6 \\ 2 & -1 & 6 \end{bmatrix} \quad \text{devient} \quad \begin{bmatrix} 2 & -1 & 6 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \widetilde{B}_2$$

La **deuxième étape** consiste à exprimer l'ensemble de des solutions de $\widetilde{B}_\lambda \mathbf{v} = \mathbf{0}$ sous forme d'une combinaison linéaire, c'est-à-dire à trouver une base de E_λ.
On observe que $\widetilde{B}_\lambda \mathbf{v} = \mathbf{0}$ possède un nombre $m_\lambda < n$ d'équations linéaires indépendantes (m_λ est le nombre de lignes non nulles de $\widetilde{B}_\lambda$). On choisit alors m_λ variables dites liées (une par équation, par exemple correspondant au pivot) et $(n - m_\lambda)$ variables dites libres (les autres). On exprime alors les variables liées en fonction des variables libres (par un algorithme similaire à l'algorithme de remontée dans la résolution de systèmes), puis on exprime les solutions de $\widetilde{B}_\lambda \mathbf{v} = \mathbf{0}$ comme une combinaison linéaire dont les coefficients sont les variables libres.

Exemple 9.4

Reprenons l'exemple 9.3.
Pour $\lambda_1 = 9$ on a $m_9 = 2$ équations indépendantes. On choisit les variables liées v_1 et v_2, et la variable libre v_3. Alors

$$\widetilde{B}_9\mathbf{v} = \mathbf{0} \quad \Leftrightarrow \left\{ \begin{array}{rcrcrcl} -5v_1 & - & v_2 & + & 6v_3 & = & 0 \\ & - & v_2 & + & v_3 & = & 0 \end{array} \right. \quad \Leftrightarrow \left\{ \begin{array}{rcl} -5v_1 & = & v_2 - 6v_3 \\ v_2 & = & v_3 \end{array} \right. \quad \Leftrightarrow \left\{ \begin{array}{rcl} v_1 & = & v_3 \\ v_2 & = & v_3 \end{array} \right.$$

Donc toute solution s'écrit $\mathbf{v} = v_3 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ et $E_9 = \mathcal{L}\left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$.
On vérifie par exemple pour $v_3 = 2$ que $A \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 18 \\ 18 \\ 18 \end{bmatrix}$.
Les vecteurs de E_9 subissent une mise à l'échelle d'un facteur 9.

Pour $\lambda_2 = 2$ on a $m_2 = 1$ équations indépendantes. On choisit la variable liée v_1, et les variables libres v_2 et v_3. Alors

$$\widetilde{B}_2\mathbf{v} = \mathbf{0} \quad \Leftrightarrow \quad 2v_1 - v_2 + 6v_3 = 0 \quad \Leftrightarrow \quad v_1 = \frac{1}{2}v_2 - 3v_3$$

Donc toute solution s'écrit $\mathbf{v} = v_2 \begin{bmatrix} 1/2 \\ 1 \\ 0 \end{bmatrix} + v_3 \begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix}$ et $E_2 = \mathcal{L}\left\{ \begin{bmatrix} 1/2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix} \right\}$.
On vérifie par exemple pour $v_2 = 2$ et $v_3 = 1$ que $A \begin{bmatrix} -2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 2 \end{bmatrix}$.
Les vecteurs de E_2 subissent une mise à l'échelle d'un facteur 2.

On remarque que $dim(E_9) = 1 = n - m_9$ et $dim(E_2) = 2 = n - m_2$. De manière générale $dim(E_\lambda) = n - m_\lambda$.

Remarque.
Cette méthode de calcul des espaces propres fonctionne pour les calculs à la main, mais elle n'est pas fiable numériquement. Les erreurs d'arrondi peuvent mener à une forme échelonnée erronée.

9.3 Diagonalisation

Comme nous le verrons dans l'exemple 9.6, il est parfois utile de savoir calculer la puissance d'une matrice. Dans le cas où cette matrice est diagonale, ce calcul est simple :

$$D = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix} \quad \Rightarrow \quad D^2 = \begin{bmatrix} 5^2 & 0 \\ 0 & 3^2 \end{bmatrix} \quad \Rightarrow \quad D^3 = \begin{bmatrix} 5^3 & 0 \\ 0 & 3^3 \end{bmatrix} \quad \Rightarrow \quad D^k = \begin{bmatrix} 5^k & 0 \\ 0 & 3^k \end{bmatrix}$$

Prenons maintenant $A = \begin{bmatrix} 7 & 2 \\ -4 & 1 \end{bmatrix}$ qui s'écrit $A = PDP^{-1}$ avec $P = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}$ et $D = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}$. Alors

$$A^2 = (PDP^{-1})(PDP^{-1}) = PD(P^{-1}P)DP^{-1} = PD^2P^{-1}$$

$$A^3 = AA^2 = (PDP^{-1})(PD^2P^{-1}) = PD^3P^{-1}$$

En fait pour tout $k \geq 1$,

$$A^k = PD^kP^{-1} = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} 5^k & 0 \\ 0 & 3^k \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix}$$

Donc si on sait écrire A sous la forme PDP^{-1} avec D une diagonale et P inversible, alors on peut facilement calculer ses puissances.

Matrices semblables. On dit que deux matrices A et B sont semblables si et seulement si il existe une matrice P inversible telle que $B = PAP^{-1}$.

Diagonalisation. Une matrice est dite diagonalisable si elle est semblable à une matrice diagonale, ce qui est vrai si et seulement si elle a n vecteurs propres linéairement indépendants.
Dans l'écriture $A = PDP^{-1}$, les éléments diagonaux de D sont les valeurs propres de A et les colonnes de P sont les vecteurs propres correspondants.

Remarques.
- Toutes les matrices ne sont pas diagonalisables.
- Une matrice qui a n valeurs propres réelles distinctes est toujours diagonalisable.

Exemple 9.5

Reprenons les exemples 9.3 et 9.4 : $A = \begin{bmatrix} 4 & -1 & 6 \\ 2 & 1 & 6 \\ 2 & -1 & 8 \end{bmatrix}$ et $P = \begin{bmatrix} 1 & 1 & -3 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

On a

$$A = P \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} P^{-1}$$

donc

$$A^k = P \begin{bmatrix} 9^k & 0 & 0 \\ 0 & 2^k & 0 \\ 0 & 0 & 2^k \end{bmatrix} P^{-1}$$

Exemple 9.6 Système proie/prédateur.

Dans les forêts de séquoias californiennes, les chouettes se nourrissent essentiellement de rats. Notons C_k le nombre de chouettes, et R_k le nombre milliers de rats au k-ième mois. On suppose que

$$\begin{cases} C_k &= & 0,5\, C_{k-1} &+ & 0,4\, R_{k-1} \\ R_k &= & -0,104\, C_{k-1} &+ & 1,1\, R_{k-1} \end{cases}$$

Ce système signifie que :
- En l'absence de rats, seules la moitié des chouettes survivent chaque mois.
- Si les rats sont abondants, ils permettent aux chouettes de vivre et de se reproduire à raison de $1000/0,4 = 2500$ rats pour 1 chouette.
- En l'absence de chouettes, les rats prolifèrent de 10% chaque mois.
- Si les chouettes sont là, elles mangent 104 rats chacune par mois.

En posant $\mathbf{u}_k = \begin{bmatrix} C_k \\ R_k \end{bmatrix}$ et $A = \begin{bmatrix} 0,5 & 0,4 \\ -0,104 & 1,1 \end{bmatrix}$, ce système s'écrit

$$\mathbf{u}_k = A\mathbf{u}_{k-1} = A^2\mathbf{u}_{k-2} = \ldots = A^k\mathbf{u}_0$$

Or la matrice A admet les éléments propres $\lambda_1 = 1,02$ associé à $\mathbf{v}_1 = \begin{bmatrix} 10 \\ 13 \end{bmatrix}$ et $\lambda_2 = 0,58$ associé à $\mathbf{v}_2 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$. Donc on peut calculer la population au mois k en fonction du mois 0 :

$$\mathbf{u}_k = P \begin{bmatrix} 1,02^k & 0 \\ 0 & 0,58^k \end{bmatrix} P^{-1}\mathbf{u}_0$$

On cherche maintenant à savoir comment se comporte “globalement” les populations à long terme. Pour cela, écrivons $\mathbf{u_0} = u_1\mathbf{v}_1 + u_2\mathbf{v}_2$ dans la base de vecteurs propres. On a alors

$$\begin{aligned}\mathbf{u}_k &= A^k(u_1\mathbf{v}_1 + u_2\mathbf{v}_2)\\ &= u_1A^k\mathbf{v}_1 + u_2A^k\mathbf{v}_2\\ &= u_1(1,02)^k\mathbf{v}_1 + u_2(0,58)^k\mathbf{v}_2\end{aligned}$$

Le temps passant, k devient grand, $(0,58)^k$ se rapproche rapidement de 0, et

$$\mathbf{u}_k \approx u_1(1,02)^k \begin{bmatrix}10\\13\end{bmatrix} \approx 1,02\ \mathbf{u}_{k-1}.$$

Donc les populations tendent à croître au rythme de 2% par mois, avec un ratio de 10 chouettes pour $13\,000$ rats.

Écriture dans une base de vecteurs propres et lien avec les applications linéaires.
Supposons que la matrice A, représente une application linéaire l dans la base canonique e de $\mathbb{R}^n$:

$$\begin{aligned}l^{(e,e)} : \mathbb{R}^n &\longrightarrow \mathbb{R}^n\\ \mathbf{u} &\longmapsto A\mathbf{u}\end{aligned}$$

Si A est diagonalisable, elle admet une base $v = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ de vecteurs propres associés à des valeurs propres $\{\lambda_1, \ldots, \lambda_n\}$. Soit $P = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ la matrice dont les colonnes sont les vecteurs de v exprimés dans la base canonique.

Donc P est la matrice de passage de la base v dans la base e : $P = P_{v\to e}$. Comme $A = PDP^{-1}$, la matrice $D = P^{-1}AP = P_{e\to v}AP_{v\to e}$ représente l dans la base v :

$$l^{(v,v)}\left(\mathbf{v}^{(v)}\right) = D\mathbf{v}^{(v)} \quad \text{avec} \quad D = \begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix}$$

Cette expression montre que dans la base de vecteurs propres, l’application linéaire multiplie simplement la coordonnée sur chaque vecteur propre $\mathbf{v}_i$ par la valeur propre λ_i.

Exemple 9.7 projection sur une droite dans $\mathbb{R}^2$.
Nous avons vu dans le chapitre précédent (exemples 8.3, 8.6, 8.10) que la projection sur la droite vectorielle $\mathcal{D} = \mathcal{L}\left\{\begin{bmatrix}2\\1\end{bmatrix}\right\}$ est représentée :

- Dans la base canonique par la matrice $A = \frac{1}{5}\begin{bmatrix}4 & 2\\2 & 1\end{bmatrix}$.
- Dans la base $\left\{\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}-1\\2\end{bmatrix}\right\}$ par la matrice $D = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$. Il s’agit de l’écriture dans une base de vecteurs propres : la composante parallèle à la droite est inchangée (valeur propre 1) alors que la composante perpendiculaire est annulée (valeur propre 0).

9.4 Calcul approché des valeurs propres et vecteurs propres

Nous avons signalé à la section 9.1 que le calcul des valeurs propres par le polynôme caractéristique est impossible pour les grandes matrices. Nous allons voir ici comment approximer itérativement la plus grande valeur propre ainsi qu’un vecteur propre associé, à l’aide de la “méthode de la puissance”.

Soit A une matrice $n \times n$ diagonalisable avec une valeur propre strictement dominante (plus grande que toutes les autres en valeur absolue). On range les valeurs propres $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq$

$\ldots \geq |\lambda_n|$ avec une base de vecteurs propres $v = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. Soit $\mathbf{u}$ un vecteur qui s'écrit $\mathbf{u} = u_1\mathbf{v}_1 + \ldots + u_n\mathbf{v}_n$ dans la base v. Comme dans l'exemple du système proie-prédateur, on peut écrire :

$$A^k\mathbf{u} = u_1(\lambda_1)^k\mathbf{v}_1 + u_2(\lambda_2)^k\mathbf{v}_2 + \ldots + u_n(\lambda_n)^k\mathbf{v}_n$$

Donc en divisant par λ_1^k on a

$$\frac{1}{\lambda_1^k}A^k\mathbf{u} = u_1\mathbf{v}_1 + u_2(\frac{\lambda_2}{\lambda_1})^k\mathbf{v}_2 + \ldots + u_n(\frac{\lambda_n}{\lambda_1})^k\mathbf{v}_n$$

Comme les fractions $\frac{\lambda_2}{\lambda_1}, \ldots, \frac{\lambda_n}{\lambda_1}$ sont strictement inférieures à 1 en module, leurs puissances tendent vers 0, et donc

$$\mathbf{u}_k = \frac{1}{\lambda_1^k}A^k\mathbf{u} = \frac{1}{\lambda_1}A\mathbf{u}_{k-1} \to u_1\mathbf{v}_1 \text{ quand } k \to \infty$$

Donc $\mathbf{u}_k$ se rapproche de $u_1\mathbf{v}_1$ qui est lui même un vecteur propre associé à λ_1, à condition que $u_1 \neq 0$. Le problème pour faire ce calcul est qu'il faudrait déjà connaître λ_1. Au lieu de normaliser par λ_1, nous allons normaliser par μ_k la plus grande composante (en valeur absolue) de $A\mathbf{u}_{k-1}$:

$$\mathbf{u}_k = \frac{1}{\mu_k}A\mathbf{u}_{k-1}$$

Il se trouve que cette suite converge vers un multiple de $\mathbf{v}_1$, dont la plus grande composante est égale à 1, et donc μ_k est proche de λ_1. L'algorithme ALGO. (9.1) calcule cette suite. Il permet donc d'approximer en même temps λ_1 (par la valeur de μ) et un vecteur propre (par la valeur de $\mathbf{u}$).

ALGO. **9.1** $approximation\_vp(A)$.

---

choisir $\mathbf{u}$ dont la plus grande composante est 1
tant que l'approximation n'est pas assez bonne **faire**
  $\mathbf{u} \leftarrow A\mathbf{u}$
  $\mu \leftarrow$ la composante de $\mathbf{u}$ de plus grande valeur absolue
  $\mathbf{u} \leftarrow \mathbf{u}/\mu$
fin tant que
renvoyer μ et $\mathbf{u}$

Remarques.

- Condition d'arrêt de la boucle : on choisit d'arrêter l'itération quand μ et $\mathbf{u}$ n'évoluent presque plus.
- Cette méthode peut être adaptée pour calculer la plus petite valeur propre, ou bien pour affiner d'autres valeurs propres dont on connaît une approximation grossière.
- Il existe des méthodes plus complexes pour approximer toutes les valeurs propres à la fois.

Annexes

Annexe A

Comment éviter les erreurs ?

Qu'il s'agisse de faire un calcul, écrire un algorithme ou un programme, la solution est souvent difficile à trouver mais facile à vérifier. Votre objectif doit être avant tout de ne pas faire d'erreurs. Donc prenez le temps de vérifier vos résultats ! Voici quelques conseils, faciles et rapides à mettre en œuvre.

Calculs. Prenons l'exemple la résolution d'un système dans la section 6.2. Il y a des risques de se tromper dans le calcul, mais vérifier que $\begin{bmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 2 & -3 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ est bien égal au membre de droite, c'est rapide et presque sans risque. Il en va de même pour les changements de base (exemple 8.9), les calculs d'éléments propres (exemple 9.3), etc. Un conseil pour que ça fonctionne : ne calculez pas "de tête" en vous disant "ça a l'air bon", mais faites *vraiment* le calcul au brouillon, *sans regarder le résultat*.

Algorithmes. Plusieurs alternatives sont possibles :

- Déroulez un exemple à la main. Un conseil pour que ce soit efficace : déroulez les instructions avec minutie, et *écrivez* les résultats au fur et à mesure.
- Observez des indices comme :
  Est-ce que la complexité est celle attendue ?
  Est-ce que toutes les données sont parcourues ?
  Est-ce que les dimensions de matrices et vecteurs sont cohérentes ?
- Programmez l'algorithme et testez le code.

Programmes. Faites des tests, c'est à dire exécutez votre programme avec plusieurs entrées et vérifiez les sorties. Pour qu'un test soit fiable il ne suffit pas d'observer quelques sorties en se disant "ça a l'air bon" :

- Prévoyez des entrées variées qui couvrent un maximum de situations.
- Écrivez les résultats attendus.
- Seulement *ensuite*, exécuter le programme.

Annexe B

Questions de cours

Pour vous entraîner et auto-évaluer votre maîtrise du cours, voici des questions de cours classiques, organisées par chapitres.

B.1 Rappels d'algèbre

- Qu'est-ce qu'une base ?
- Qu'est-ce que la dimension d'un espace vectoriel ?
- Pourquoi peut-on généralement travailler dans $\mathbb{R}^n$ ou $\mathbb{C}^n$?

B.2 Rappels sur la complexité

- Que mesure la complexité asymptotique ?
- Que (ne) peut-on (pas) faire avec ?
- Quelles complexités sont faibles ? fortes ? ingérables ?

B.3 Les matrices

- La transposée est-elle distributive sur le produit matriciel ?
- L'inversion de matrice est-elle distributive sur le produit matriciel ?
- Quelle est la complexité du produit matrice-matrice ?
- Selon quels critères choisir une structure de données pour les matrices ?

B.4 Résolution de systèmes linéaires

- Quelles sont les conditions pour pouvoir appliquer la méthode du pivot de Gauss ?
- Quelles sont les principales étapes de cette méthode ?
- Quelle est la complexité de cette méthode ?
- Qu'est-ce que la méthode du pivot partiel ? Quel est son intérêt ?

B.5 Approximation au sens des moindres carrés

- D'où vient le terme *moindres carrés* ?
- Que sont les équations normales ? Comment les calculer ?

- Comment connaître la précision d'une régression ?
- Comment gérer des données peu fiables ?

B.6 Applications linéaires

- À quoi sert une matrice de passage ?
- Soient e et e', deux bases de $\mathbb{R}^n$ et soit $P_{e\to e'}$, la matrice de passage de e vers e'. Que contient la i^{eme} colonne de $P_{e\to e'}$?

B.7 Éléments propres

- Qu'est-ce que le polynôme caractéristique d'une matrice ?
- Qu'est-ce qu'un espace propre ? Comment le calculer ?
- Quelles matrices sont diagonalisables ?
- Quel est le lien entre le noyau d'une application linéaire et les éléments propres ?
- Comment calculer les puissances d'une matrice ?

Annexe C

Annales

Pour vous entraîner et préparer les évaluations, voici des exercices posés les années précédentes, organisés par chapitres. Les durées indicatives contiennent le temps de rédaction ou de recopie au propre.

C.1 Les matrices

Exercice C.1 Matrices circulantes (durée 50 minutes).

Une matrice A carrée est circulante si elle a la forme suivante :

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \\ a_n & a_1 & a_2 & \dots & a_{n-1} \\ a_{n-1} & a_n & a_1 & \dots & a_{n-2} \\ \vdots & & & & \vdots \\ a_2 & a_3 & \dots & a_n & a_1 \end{bmatrix}. \quad \text{On note } \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \text{ le vecteur associé à } A.$$

Le vecteur $\mathbf{a}$ suffit à reconstruire la matrice A, ce qui permet un stockage compact : au lieu de stocker A entièrement, on stocke uniquement $\mathbf{a}$. Nous allons faire différents calculs matriciels avec $\mathbf{a}$ au lieu de A.

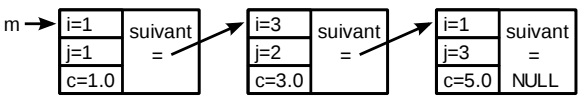
1. Calculez le produit de A par le vecteur ne contenant que des 1. Qu'en déduisez-vous ?
2. La matrice $B = A^T$ est circulante. Que vaut le vecteur $\mathbf{b}$ associé à B ?
   Écrivez un algorithme qui calcule $\mathbf{b}$ en fonction de $\mathbf{a}$.
3. Écrivez un algorithme qui construit la matrice A à partir du vecteur $\mathbf{a}$.
4. Écrivez un algorithme qui calcule le vecteur $\mathbf{v} = A\mathbf{u}$, en fonction de $\mathbf{a}$ et $\mathbf{u}$ (sans construire A).

Exercice C.2 Matrices creuses (durée 20 minutes).

On représente des matrices creuses (*i.e.* contenant beaucoup de 0) par une liste chaînée des coefficients non nuls avec leurs indices $1 \leq i \leq N$ et $1 \leq j \leq N$:

```
struct elem {int i; int j; float c; struct elem * suivant;};
typedef struct elem * matrice;
```

Par exemple la matrice $m = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$ peut être représentée par la liste :



Les matrices sont carrées, et leur taille est définie par une constante `N` globale. Les vecteurs sont représentés par des tableaux de flottants.

Codez une fonction de profil `float* produit (matrice m, float* v)` qui alloue, calcule et renvoie le résultat du produit `mv`.

Exercice C.3 Algorithme d'orthonormalisation de Gram-Schmidt (durée 1h).

À partir d'une base quelconque $\mathbf{a} = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ de $\mathbb{R}^n$, l'algorithme de Gram-Schmidt calcule une base orthonormale $\mathbf{e} = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ définie ainsi :

$$
\begin{aligned}
\mathbf{e_1} &= \frac{\mathbf{b_1}}{\|\mathbf{b_1}\|} && \text{avec } \mathbf{b_1} = \mathbf{a_1} \\
\mathbf{e_2} &= \frac{\mathbf{b_2}}{\|\mathbf{b_2}\|} && \text{avec } \mathbf{b_2} = \mathbf{a_2} - p(\mathbf{e_1}, \mathbf{a_2}) \\
\mathbf{e_3} &= \frac{\mathbf{b_3}}{\|\mathbf{b_3}\|} && \text{avec } \mathbf{b_3} = \mathbf{a_3} - p(\mathbf{e_1}, \mathbf{a_3}) - p(\mathbf{e_2}, \mathbf{a_3}) \\
&\vdots \\
\mathbf{e_k} &= \frac{\mathbf{b_k}}{\|\mathbf{b_k}\|} && \text{avec } \mathbf{b_k} = \mathbf{a_k} - \sum_{j=1}^{k-1} p(\mathbf{e_j}, \mathbf{a_k}) \\
&\vdots
\end{aligned}
$$

où p représente la projection $p(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}; \mathbf{v} \rangle \, \mathbf{u}$.
Pour les algorithmes, les bases d'entrée et de sortie seront stockées comme colonnes de matrices

$$A = \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \cdots & \mathbf{a_n} \end{bmatrix} \text{ et } E = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \cdots & \mathbf{e_n} \end{bmatrix}$$

La taille n est supposée fixée globalement.

1. Dérouler à la main l'algorithme pour la base de $\mathbb{R}^3$ suivante :
$$\mathbf{a_1} = \begin{bmatrix} 3 \\ 0 \\ 4 \end{bmatrix}, \quad \mathbf{a_2} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{a_3} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$
2. Écrire l'algorithme de Gram-Schmidt.
Indication : on pourra définir des algorithmes intérmédiaires
ProduitScalaire(A, j, B, k) qui renvoie le produit scalaire des colonnes $A(., j)$ et $B(., k)$; Normalise(A, k) qui normalise la colonne $A(., k)$.
3. Quelle est la complexité de cet algorithme ?
4. Une matrice dont les colonnes sont orthonormales (comme E) est appelée *matrice orthogonale*.
Que vaut $E^T E$? Que peut valoir le déterminant de E ? Justifiez vos réponses.

C.2 Résolution de systèmes linéaires

Exercice C.4 Systèmes linéaires (durée 20 minutes).

Calculer $det(A)$, puis résoudre $Ax = b$ pour

$$A = \begin{bmatrix} 0 & 3 & 3 \\ -2 & 2 & 1 \\ 3 & -1 & -1 \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} 9 \\ 9 \\ -9 \end{bmatrix}$$

Exercice C.5 Systèmes linéaires (durée 20 minutes).
Calculer $det(A)$, puis résoudre $Ax = b$ pour

$$A = \begin{bmatrix} 0 & 2 & -1 \\ 2 & 1 & 3 \\ -4 & 2 & -6 \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} -7 \\ 13 \\ -34 \end{bmatrix}$$

Exercice C.6 Factorisation LU (durée 25 minutes).
Supposons qu'on sache factoriser une matrice carrée A de taille $n \times n$ sous la forme $A = LU$, où L (respectivement U) est une matrice triangulaire inférieure (respectivement supérieure). Cette factorisation est calculée en temps $\Theta(n^3)$.

1. Proposez une méthode de résolution du système $AX = B$ s'appuyant sur cette décomposition.
2. Quelle est sa complexité ?
3. On souhaite résoudre le système pour plusieurs membres de droite B. Quel est l'avantage de cette méthode par rapport au pivot de Gauss ?

Exercice C.7 Pivot de Gauss (durée 45 minutes).
On veut ré-écrire la méthode du pivot de Gauss partiel en utilisant une matrice triangulaire inférieure au lieu de supérieure.

1. Écrivez un algorithme qui transforme un système quelconque en système triangulaire inférieur équivalent.
2. Écrivez un algorithme "de descente" qui résout un système triangulaire inférieur.

C.3 Approximation au sens des moindres carrés

Exercice C.8 Droite des moindres carrés (durée 25 minutes).
Calculez la droite des moindres carrés pour les observations $(1, 3)$, $(3, 3)$, $(5, 2)$ et $(7, 0)$. Faites un dessin. Calulez les résidus.

Exercice C.9 Droite des moindres carrés (durée 20 minutes).
Calculer la droite des moindres carrés puis les résidus pour les observations suivantes :

$i =$	1	2	3	4	5
$x_i =$	-2	0	3	4	5
$y_i =$	0	2	4	4	3

Exercice C.10 Moindres carrés : algorithme général (durée 35 minutes).

1. Écrire un algorithme `resout` de résolution de système linéaire de taille 2×2 :
   - paramètres : A et $\mathbf{b}$ (où A est supposée inversible) ;
   - résultat : le vecteur $\mathbf{x}$ qui vérifie $A\mathbf{x} = \mathbf{b}$.

   Indication : on pourra utiliser la formule $A^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{det(A)} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$.
2. Écrire un algorithme `regression` qui calcule la droite des moindres carrés :
   - paramètres : deux vecteurs $\mathbf{x}$ et $\mathbf{y}$ (les observations), un entier n (la taille des vecteurs) ;
   - résultat : le vecteur $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ des coefficients de la droite des moindres carrés $y = \alpha x + \beta$.
3. Quelle est la complexité de `regression` ?

Période $i =$	1	2	3	4
Température moyenne $t_i =$	-5	0	10	15
Quantité de gaz consommé $g_i =$	30	12	8	7

TABLE C.1 – Mesures de la température moyenne et de la consommation de gaz.

Exercice C.11 Chauffage au gaz (durée 40 minutes).

Un distributeur de gaz observe la quantité de gaz que ses clients consomment pendant plusieurs périodes (TABLE C.1). Il constate que plus il fait froid, plus les clients consomment de gaz. Pour bien s'approvisionner, il voudrait prévoir la quantité de gaz g en fonction de la température t.

1. On essaie une prévision par une droite des moindres carrés $g = at + b$. Calculez cette droite. Faites un dessin. Quelle quantité de gaz prévoyez vous pour $t = 5$?
2. On veut une prévision quadratique plus précise $g = at^2 + bt + c$. Proposez une méthode pour calculer cette prévision (le calcul sur l'exemple numérique n'est pas demandé).

C.4 Applications linéaires

Exercice C.12 Bases et applications linéaires (durée 45 minutes).

On se place dans l'espace E des polynômes de degré inférieur ou égal à 3. On définit les deux bases :

$$e = \{e_1(X) = 1,\ e_2(X) = X,\ e_3(X) = X^2,\ e_4(X) = X^3\}$$
$$e' = \{e'_1(X) = X,\ e'_2(X) = 1 - X,\ e'_3(X) = X^3,\ e'_4(X) = (1 - X)^3\}$$

On appelle d l'application qui dérive un polynôme :

$$\begin{array}{rccc} d: & E & \longrightarrow & E \\ & \mathbf{p}(X) & \longmapsto & \mathbf{p}'(X) \end{array}$$

1. Montrer que d est une application linéaire de E dans E.
2. Calculer la matrice D qui représente d dans la base e.
3. Calculer $Ker(d)$, le noyau de d.
4. Calculer $P_{e' \to e}$ la matrice de passage de e' dans e.
5. Calculer la matrice D' qui représente d dans la base e'.

Exercice C.13 Changement de base (durée 20 minutes).

On se donne deux bases de $\mathbb{R}^2$: e la base canonique, et $e' = \left\{ \mathbf{e'}_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \mathbf{e'}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \right\}$.

Soit $l : \mathbb{R}^2 \to \mathbb{R}^2$ l'application linéaire représentée de la base e' dans la base e' par la matrice $\begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}$.

Calculez la matrice qui représente l de la base e dans la base e (détailler les calculs).

C.5 Éléments propres

Exercice C.14 Élements propres, puissance d'une matrice (durée 45 minutes).

Soit la matrice A suivante :

$$A = \begin{bmatrix} 0 & -1 \\ 3 & 4 \end{bmatrix}$$

1. Calculer les valeurs propres de la matrice A.
2. Calculer les vecteurs propres correspondants.
3. En déduire la diagonalisation de la matrice A.
4. Calculer A^k.

Exercice C.15 Élements propres (durée 15 à 20 minutes).
Soit l l'application linéaire de $\mathbb{R}^2$ dans $\mathbb{R}^2$ ayant pour éléments propres :

$$\lambda_1 = -1,\ \mathbf{v}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix},\ \lambda_2 = 2,\ \mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Calculez la matrice A représentant l dans la base canonique.

Exercice C.16 Élements propres (durée 20 minutes).
On se donne la matrice $A = \begin{bmatrix} -4 & 0 & 4 \\ 3 & 1 & -3 \\ -2 & 0 & 2 \end{bmatrix}$ et le vecteur $\mathbf{v} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$.

1. Calculer $det(A)$.
2. Qu'en déduisez-vous (plusieurs réponses) ?
3. Calculer $A\mathbf{v}$.
4. Qu'en déduisez vous ?