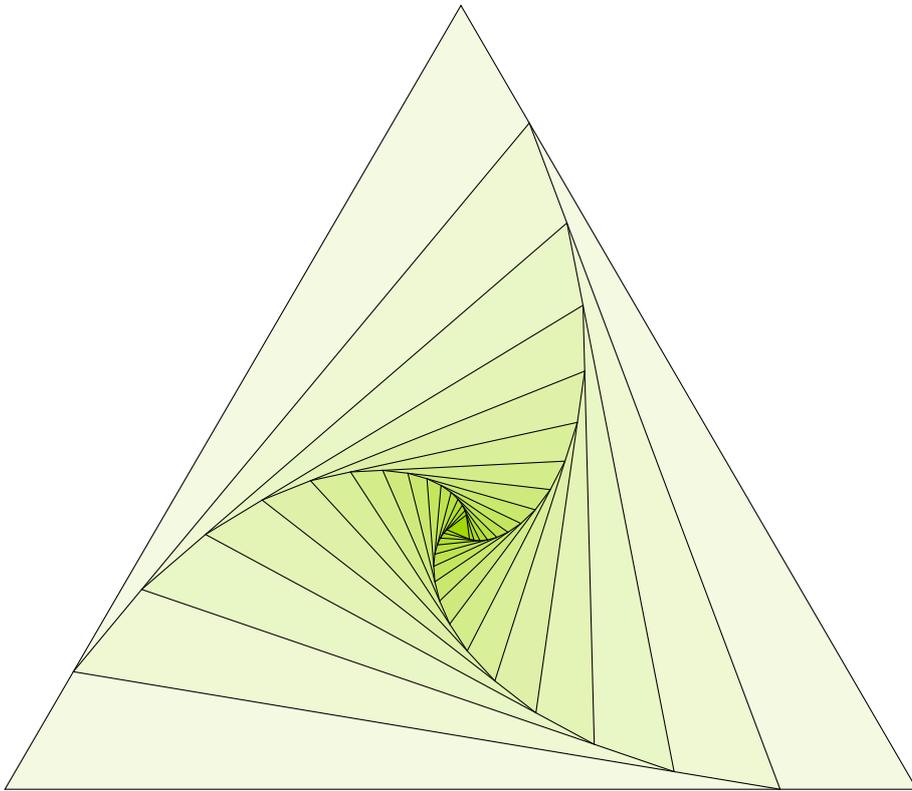


# Analyse numérique élémentaire

## *Notes de cours*

Sup Galilée, Ingénieurs MACS 1ère année & L3-MIM

Version du 2025/09/29



Francois Cuvelier  
Université Paris XIII / Institut Galilée  
L.A.G.A./Département de Mathématiques  
<http://www.math.univ-paris13.fr/~cuvelier>  
[cuvelier@math.univ-paris13.fr](mailto:cuvelier@math.univ-paris13.fr)

---



# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Représentation des nombres en machine, erreurs d'arrondis</b> | <b>1</b>  |
| 1.1      | Un exemple : calcul approché de $\pi$                            | 1         |
| 1.2      | Représentation scientifique des nombres dans différentes bases   | 2         |
| 1.2.1    | Partie entière, mantisse et exposant                             | 2         |
| 1.3      | Nombres flottants : le système IEEE 754                          | 5         |
| 1.3.1    | Simple précision   | 6         |
| 1.3.2    | Double précision   | 7         |
| 1.3.3    | Matlab   | 7         |
| 1.4      | Calculs sur les nombres flottants                                | 8         |
| 1.4.1    | Erreurs d'arrondi  | 8         |
| 1.4.2    | Associativité  | 8         |
| 1.4.3    | Monotonie  | 8         |
| 1.4.4    | Erreurs d'annulation   | 8         |
| 1.5      | Quelques catastrophes dues à l'arithmétique flottante            | 9         |
| <b>2</b> | <b>Résolution de systèmes non linéaires</b>                      | <b>13</b> |
| 2.1      | Recherche des zéros d'une fonction                               | 14        |
| 2.1.1    | Méthode de dichotomie ou de bisection                            | 14        |
| 2.2      | Points fixes d'une fonction (dimension 1)                        | 20        |
| 2.2.1    | Points fixes attractifs et répulsifs                             | 26        |
| 2.2.2    | Interprétations graphiques de la méthode du point fixe           | 27        |
| 2.2.3    | Algorithme générique du point fixe                               | 30        |
| 2.2.4    | Méthodes de points fixes pour la recherche de racines            | 32        |
| 2.2.5    | La méthode de la sécante   | 43        |
| 2.2.6    | Méthode Regula-Falsi ou fausse position                          | 44        |
| 2.3      | Résolution de systèmes non linéaires                             | 47        |
| 2.3.1    | Point fixe   | 48        |
| 2.3.2    | Méthode de Newton  | 50        |
| 2.3.3    | Exemples   | 51        |
| <b>3</b> | <b>Résolution de systèmes linéaires</b>                          | <b>57</b> |
| 3.1      | Méthodes directes  | 58        |
| 3.1.1    | Matrices particulières   | 59        |
| 3.1.2    | Exercices et résultats préliminaires                             | 63        |

|          |  |            |
|----------|--|------------|
| 3.1.3    | Méthode de Gauss-Jordan, écriture matricielle . . . . .                        | 69         |
| 3.1.4    | Factorisation LU . . . . .   | 73         |
| 3.1.5    | Factorisation LDL* . . . . .   | 83         |
| 3.1.6    | Factorisation de Cholesky . . . . .  | 84         |
| 3.1.7    | Factorisation QR . . . . .   | 90         |
| 3.2      | Normes vectorielles et normes matricielles . . . . .                           | 99         |
| 3.2.1    | Normes vectorielles . . . . .  | 99         |
| 3.2.2    | Normes matricielles . . . . .  | 100        |
| 3.2.3    | Suites de vecteurs et de matrices . . . . .                                    | 105        |
| 3.3      | Conditionnement d'un système linéaire . . . . .                                | 106        |
| 3.4      | Méthodes itératives . . . . .  | 108        |
| 3.4.1    | Principe . . . . .   | 108        |
| 3.4.2    | Présentation des méthodes usuelles . . . . .                                   | 109        |
| 3.4.3    | Etude de la convergence . . . . .  | 112        |
| 3.4.4    | Algorithmes . . . . .  | 118        |
| 3.4.5    | Algorithmes matriciels . . . . .   | 125        |
| 3.4.6    | Exercices . . . . .  | 126        |
| <b>4</b> | <b>Interpolation</b> . . . . .   | <b>135</b> |
| 4.1      | Polynôme d'interpolation de Lagrange . . . . .                                 | 136        |
| 4.1.1    | Erreur de l'interpolation . . . . .  | 140        |
| 4.1.2    | Points de Chebyshev . . . . .  | 142        |
| 4.1.3    | Stabilité . . . . .  | 142        |
| 4.2      | Polynôme d'interpolation de Lagrange-Hermite . . . . .                         | 147        |
| 4.3      | Exercices . . . . .  | 153        |
| <b>5</b> | <b>Intégration numérique</b> . . . . .   | <b>161</b> |
| 5.1      | Méthodes de quadrature élémentaires . . . . .                                  | 162        |
| 5.1.1    | Méthodes simplistes . . . . .  | 162        |
| 5.1.2    | Quelques résultats théoriques . . . . .  | 164        |
| 5.1.3    | Formules élémentaires de Newton-Cotes . . . . .                                | 175        |
| 5.1.4    | Formules élémentaires de Gauss-Legendre . . . . .                              | 179        |
| 5.2      | Méthodes de quadrature composées . . . . .                                     | 188        |
| 5.2.1    | Exemples . . . . .   | 188        |
| 5.2.2    | Erreurs des méthodes de quadrature composées . . . . .                         | 192        |
| 5.3      | Intégrales multiples . . . . .   | 197        |
| <b>A</b> | <b>Langage algorithmique</b> . . . . .   | <b>199</b> |
| A.1      | Pseudo-langage algorithmique . . . . .   | 199        |
| A.1.1    | Données et constantes . . . . .  | 199        |
| A.1.2    | Variables . . . . .  | 199        |
| A.1.3    | Opérateurs . . . . .   | 200        |
| A.1.4    | Expressions . . . . .  | 200        |
| A.1.5    | Instructions . . . . .   | 201        |
| A.1.6    | Fonctions . . . . .  | 202        |
| A.1.7    | Tableaux . . . . .   | 204        |
| A.2      | Methodologie d'élaboration d'un algorithme . . . . .                           | 206        |
| A.2.1    | Description du problème . . . . .  | 206        |
| A.2.2    | Recherche d'une méthode de résolution . . . . .                                | 206        |
| A.2.3    | Réalisation d'un algorithme . . . . .  | 206        |
| A.2.4    | Exercices . . . . .  | 207        |
| A.3      | Principes de «bonne» programmation pour attaquer de «gros» problèmes . . . . . | 209        |
| A.4      | Quelques simplifications d'écriture . . . . .                                  | 210        |

|   |            |
|---|------------|
| <b>B Annexes</b>  | <b>213</b> |
| B.1 Analyse : rappels . . . . .                               | 213        |
| B.1.1 En vrac . . . . .                                       | 213        |
| B.1.2 Espace métrique <sup>1</sup> . . . . .                  | 214        |
| B.2 Algèbre linéaire . . . . .                                | 219        |
| B.2.1 Vecteurs . . . . .                                      | 219        |
| B.2.2 Matrices . . . . .                                      | 220        |
| B.2.3 Normes vectorielles et normes matricielles . . . . .    | 227        |
| B.2.4 Réduction des matrices . . . . .                        | 230        |
| B.2.5 Suites de vecteurs et de matrices . . . . .             | 231        |
| B.3 Recueil d'exercices . . . . .                             | 231        |
| B.3.1 Algorithmique numérique . . . . .                       | 231        |
| B.3.2 Algèbre linéaire . . . . .                              | 248        |
| B.3.3 Normes . . . . .  | 266        |
| B.4 Listings . . . . .  | 284        |
| B.4.1 Codes sur la méthode de dichotomie/bissection . . . . . | 284        |

---

<sup>1</sup>En grande partie extrait du site bibmat



# Chapitre 1

## Représentation des nombres en machine, erreurs d'arrondis

### NOTE

Toute cette partie est le contenu quasi-intégral d'un document réalisé par C. Japhet

Ce chapitre est une introduction à la représentation des nombres en machine et aux erreurs d'arrondis, basé sur [5], [4].

### 1.1 Un exemple : calcul approché de $\pi$

Cet exemple est extrait de [5], [4]. Le nombre  $\pi$  est connu depuis l'antiquité, en tant que méthode de calcul du périmètre du cercle ou de l'aire du disque. Le problème de la quadrature du cercle étudié par les anciens Grecs consiste à construire un carré de même aire qu'un cercle donné à l'aide d'une règle et d'un compas. Ce problème resta insoluble jusqu'au 19<sup>ème</sup> siècle, où la démonstration de la transcendance de  $\pi$  montra que le problème ne peut être résolu en utilisant une règle et un compas.

Nous savons aujourd'hui que l'aire d'un cercle de rayon  $r$  est  $\mathcal{A} = \pi r^2$ . Parmi les solutions proposées pour approcher  $\mathcal{A}$ , une méthode consiste à construire un polygone dont le nombre de côté augmenterait jusqu'à ce qu'il devienne équivalent au cercle circonscrit. C'est Archimède vers 250 avant J-C qui appliquera cette propriété au calcul des décimales du nombre  $\pi$ , en utilisant à la fois un polygone inscrit et circonscrit au cercle. Il utilise ainsi un algorithme pour le calcul et parvient à l'approximation de  $\pi$  dans l'intervalle  $(3 + \frac{10}{71}, 3 + \frac{1}{7})$  en faisant tendre le nombre de côtés jusqu'à 96.

Regardons l'algorithme de calcul par les polygones inscrits. On considère un cercle de rayon  $r = 1$  et on note  $\mathcal{A}_n$  l'aire associée au polygone inscrit à  $n$  côtés. En notant  $\alpha_n = \frac{2\pi}{n}$ ,  $\mathcal{A}_n$  est égale à  $n$  fois l'aire du triangle  $ABC$  représenté sur la figure 1.1, c'est-à-dire

$$\mathcal{A}_n = n \cos \frac{\alpha_n}{2} \sin \frac{\alpha_n}{2},$$

que l'on peut réécrire

$$\mathcal{A}_n = \frac{n}{2} (2 \cos \frac{\alpha_n}{2} \sin \frac{\alpha_n}{2}) = \frac{n}{2} \sin \alpha_n = \frac{n}{2} \sin(\frac{2\pi}{n}).$$

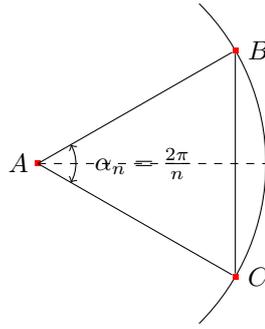


Figure 1.1: Quadrature du cercle

Comme on cherche à calculer  $\pi$  à l'aide de  $A_n$ , on ne peut pas utiliser l'expression ci-dessus pour calculer  $A_n$ , mais on peut exprimer  $A_{2n}$  en fonction de  $A_n$  en utilisant la relation

$$\sin \frac{\alpha_n}{2} = \sqrt{\frac{1 - \cos \alpha_n}{2}} = \sqrt{\frac{1 - \sqrt{1 - \sin^2 \alpha_n}}{2}}.$$

Ainsi, en prenant  $n = 2^k$ , on définit l'approximation de  $\pi$  par récurrence

$$x_k = A_{2^k} = \frac{2^k}{2} s_k, \quad \text{avec } s_k = \sin\left(\frac{2\pi}{2^k}\right) = \sqrt{\frac{1 - \sqrt{1 - s_{k-1}^2}}{2}}$$

En partant de  $k = 2$  (i.e.  $n = 4$  et  $s = 1$ ) on obtient l'algorithme suivant:

---

**Algorithm 1.1** Algorithme de calcul de  $\pi$ , version naïve

---

- |   |   |
|---|---|
| 1: $s \leftarrow 1, n \leftarrow 4$                           | ▷ Initialisations                       |
| 2: <b>Tantque</b> $s > 1e - 10$ <b>faire</b>                  | ▷ Arrêt si $s = \sin(\alpha)$ est petit |
| 3: $s \leftarrow \text{sqrt}((1 - \text{sqrt}(1 - s * s))/2)$ | ▷ nouvelle valeur de $\sin(\alpha/2)$   |
| 4: $n \leftarrow 2 * n$                                       | ▷ nouvelle valeur de $n$                |
| 5: $A \leftarrow (n/2) * s$                                   | ▷ nouvelle valeur de l'aire du polygone |
| 6: <b>Fin</b>   |   |
- 

On a  $\lim_{k \rightarrow +\infty} x_k = \pi$ . Ce n'est pourtant pas du tout ce que l'on va observer sur machine! Les résultats en Python (sous Sage) de la table 1.1 montre que l'algorithme commence par converger vers  $\pi$  puis pour  $n > 65536$ , l'erreur augmente et finalement on obtient  $A_n = 0$ !! "Although the theory and the program are correct, we obtain incorrect answers" ([5]).

Ceci résulte du codage des valeurs réelles sur un nombre fini de bits, ce que nous allons détailler dans ce chapitre.

## 1.2 Représentation scientifique des nombres dans différentes bases

Dans cette section nous introduisons les notions de mantisse, exposant, et la façon dont sont représentés les nombres sur une calculatrice ou un ordinateur.

### 1.2.1 Partie entière, mantisse et exposant

#### Exemple en base 10

La base 10 est la base naturelle avec laquelle on travaille et celle que l'on retrouve dans les calculatrices. Un nombre à virgule, ou nombre décimal, a plusieurs écritures différentes en changeant simplement la position du point décimal et en rajoutant à la fin une puissance de 10 dans l'écriture de ce nombre. La partie à gauche du point décimal est la partie entière, celle à droite avant l'exposant s'appelle la mantisse. Par exemple le nombre  $x = 1234.5678$  a plusieurs représentations :

$$x = 1234.5678 = 1234.5678 \cdot 10^0 = \mathbf{1.2345678} \cdot 10^3 = 0.0012345678 \cdot 10^6, \quad (1.1)$$

| $n$        | $A_n$            | $ A_n - \pi $ | $\sin(\alpha_n)$ |
|------------|------------------|---------------|------------------|
| 4          | 2.00000000000000 | 1.141593e+00  | 1.000000e+00     |
| 8          | 2.82842712474619 | 3.131655e-01  | 7.071068e-01     |
| 16         | 3.06146745892072 | 8.012519e-02  | 3.826834e-01     |
| 32         | 3.12144515225805 | 2.014750e-02  | 1.950903e-01     |
| 64         | 3.13654849054594 | 5.044163e-03  | 9.801714e-02     |
| 128        | 3.14033115695474 | 1.261497e-03  | 4.906767e-02     |
| 256        | 3.14127725093276 | 3.154027e-04  | 2.454123e-02     |
| 512        | 3.14151380114415 | 7.885245e-05  | 1.227154e-02     |
| 1024       | 3.14157294036788 | 1.971322e-05  | 6.135885e-03     |
| 2048       | 3.14158772527996 | 4.928310e-06  | 3.067957e-03     |
| 4096       | 3.14159142150464 | 1.232085e-06  | 1.533980e-03     |
| 8192       | 3.14159234561108 | 3.079787e-07  | 7.669903e-04     |
| 16384      | 3.14159257654500 | 7.704479e-08  | 3.834952e-04     |
| 32768      | 3.14159263346325 | 2.012654e-08  | 1.917476e-04     |
| 65536      | 3.14159265480759 | 1.217796e-09  | 9.587380e-05     |
| 131072     | 3.14159264532122 | 8.268578e-09  | 4.793690e-05     |
| 262144     | 3.14159260737572 | 4.621407e-08  | 2.396845e-05     |
| 524288     | 3.14159291093967 | 2.573499e-07  | 1.198423e-05     |
| 1048576    | 3.14159412519519 | 1.471605e-06  | 5.992115e-06     |
| 2097152    | 3.14159655370482 | 3.900115e-06  | 2.996060e-06     |
| 4194304    | 3.14159655370482 | 3.900115e-06  | 1.498030e-06     |
| 8388608    | 3.14167426502176 | 8.161143e-05  | 7.490335e-07     |
| 16777216   | 3.14182968188920 | 2.370283e-04  | 3.745353e-07     |
| 33554432   | 3.14245127249413 | 8.586189e-04  | 1.873047e-07     |
| 67108864   | 3.14245127249413 | 8.586189e-04  | 9.365235e-08     |
| 134217728  | 3.16227766016838 | 2.068501e-02  | 4.712161e-08     |
| 268435456  | 3.16227766016838 | 2.068501e-02  | 2.356080e-08     |
| 536870912  | 3.46410161513775 | 3.225090e-01  | 1.290478e-08     |
| 1073741824 | 4.00000000000000 | 8.584073e-01  | 7.450581e-09     |
| 2147483648 | 0.00000000000000 | 3.141593e+00  | 0.000000e+00     |

Table 1.1: Calcul de  $\pi$  avec l'algorithme naïf 1.1

avec

- *Partie entière* : 1234
- *Mantisse* : 0.5678 ou 1.2345678 ou 0.0012345678
- *Exposant* : 4 ou 6

Selon le décalage et l'exposant que l'on aura choisi, le couple mantisse-exposant va changer mais le nombre représenté est le même. Afin d'avoir une représentation unique, la troisième dans (1.1) sera utilisée, avec 1.2345678 pour mantisse et 3 pour exposant. Elle correspond à l'écriture avec un premier chiffre avant le point décimal non nul dans la mantisse.

### Exemple en base 2

La base 2 est la base utilisée par les ordinateurs. Les chiffres utilisables dans cette base sont 0 et 1 que l'on appelle *bit* pour *binary digit*, les ordinateurs travaillent en binaire. Par exemple

$$39 = 32 + 4 + 2 + 1 = 2^5 + 2^2 + 2^1 + 2^0 = (100111)_2,$$

$$3.625 = 2^1 + 2^0 + 2^{-1} + 2^{-3} = (11.101)_2 = (1.1101)_2 2^1$$

### Représentation d'un nombre en machine : nombres flottants

De façon générale tout nombre réel  $x$  sera représenté dans une base  $b$  ( $b = 10$  pour une calculatrice  $b = 2$  pour un ordinateur) par son signe (+ ou -), la mantisse  $m$  (appelée aussi significande), la base  $b$  et un

exposant  $e$  tel que le couple  $(m, e)$  caractérise le nombre. En faisant varier  $e$ , on fait « flotter » la virgule décimale. La limitation fondamentale est que la place mémoire d'un ordinateur est limitée, c'est-à-dire qu'il ne pourra stocker qu'un ensemble fini de nombres. Ainsi un nombre machine réel ou *nombre à virgule flottante* s'écrira :

$$\begin{aligned}\tilde{x} &= \pm m \cdot b^e \\ m &= D.D \cdots D \\ e &= D \cdots D\end{aligned}$$

où  $D \in \{0, 1, \dots, b-1\}$  représente un chiffre. Des représentations approchées de  $\pi$  sont :  $(0.031, 2)$ ,  $(3.142, 0)$ ,  $(0.003, 3)$  et on observe qu'elles ne donnent pas la même précision. Pour rendre la représentation unique et garder la meilleure précision, on utilisera une mantisse *normalisée* : le premier chiffre avant le point décimal dans la mantisse est non nul. Les nombres machine correspondants sont appelés *normalisés*. En base 2, le premier bit dans la mantisse sera donc toujours 1, et on n'écrit pas ce 1 ce qui permet d'économiser un bit. L'exposant est un nombre variant dans un intervalle fini de valeurs admissibles :  $L \leq e \leq U$  (typiquement  $L < 0$  et  $U > 0$ ). Le système est donc caractérisé par quatre entiers :

- la base  $b$  ( $b = 2$ ),
- le nombre de chiffres  $t$  dans la mantisse (en base  $b$ ),
- l'exposant minimal  $L$  et maximal  $U$ .

En mathématiques on effectue les calculs avec des nombres réels  $x$  provenant de l'intervalle continu  $x \in [-\infty, \infty]$ . A cause de la limitation ci-dessus, la plupart des réels seront approchés sur un ordinateur. Par exemple,  $\frac{1}{3}$ ,  $\sqrt{2}$ ,  $\pi$  possèdent une infinité de décimales et ne peuvent donc pas avoir de représentation exacte en machine. Le plus simple des calculs devient alors approché. L'expérience pratique montre que cette quantité limitée de nombres représentables est largement suffisante pour les calculs. Sur l'ordinateur, les nombres utilisés lors des calculs sont des nombres machine  $\tilde{x}$  provenant d'un ensemble discret de nombres machine  $\tilde{x} \in \{\tilde{x}_{min}, \dots, \tilde{x}_{max}\}$ . Ainsi, chaque nombre réel  $x$  doit être transformé en un nombre machine  $\tilde{x}$  afin de pouvoir être utilisé sur un ordinateur. Un exemple est donné sur la figure 1.2.

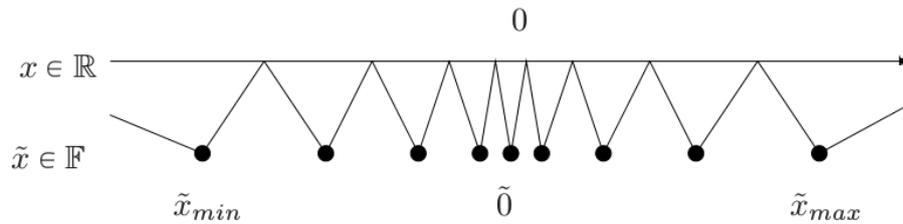


Figure 1.2: Représentation des nombres réels  $\mathbb{R}$  par les nombres machine  $\mathbb{F}$

Prenons un exemple, beaucoup trop simple pour être utilisé mais pour fixer les idées:  $t = 3$ ,  $L = -1$ ,  $U = 2$ . Dans ce cas on a 3 chiffres significatifs et 33 nombres dans le système  $\mathbb{F}$ . Ils se répartissent avec 0 d'une part, 16 nombres négatifs que l'on ne représente pas ici, et 16 nombres positifs représentés sur la figure 1.3.

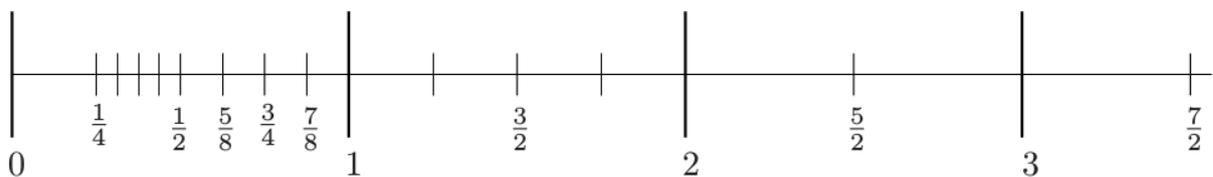


Figure 1.3: Nombres positifs de  $\mathbb{F}$  dans le cas  $t = 3$ ,  $L = -1$ ,  $U = 2$

Dans cet exemple, l'écriture en binaire des nombres entre  $\frac{1}{2}$  et 1 est

$$\begin{aligned}\frac{1}{2} &= (0.100)_2, & \frac{3}{4} &= \frac{1}{2} + \frac{1}{4} = (0.110)_2, \\ \frac{5}{8} &= \frac{1}{2} + \frac{1}{8} = (0.101)_2, & \frac{7}{8} &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = (0.111)_2.\end{aligned}$$

On obtient ensuite les autres nombres en multipliant par une puissance de 2. Le plus grand nombre représentable dans ce système est  $\frac{7}{2}$  (en particulier 4 n'est pas représentable). On remarque que les nombres ne sont pas espacés régulièrement. Ils sont beaucoup plus resserrés du côté de 0 entre  $\frac{1}{4}$  et  $\frac{1}{2}$  que entre 1 et 2 et encore plus qu'entre 2 et 3. Plus précisément, chaque fois que l'on passe par une puissance de 2, l'espacement absolu est multiplié par 2, mais l'espacement relatif reste constant ce qui est une bonne chose pour un calcul d'ingénierie car on a besoin d'une précision absolue beaucoup plus grande pour des nombres petits (autour de un millième par exemple) que des nombres très grands (de l'ordre du million par exemple). Mais la précision ou l'erreur relative sera du même ordre. L'erreur absolue ou relative est définie à section 1.4.1.

**Précision machine.** elle est décrite par le nombre machine *eps*. *eps* est le plus petit nombre machine positif tel que  $1 + eps > 1$  sur la machine. C'est la distance entre l'entier 1 et le nombre machine  $\tilde{x} \in \mathbb{F}$  le plus proche, qui lui est supérieur. Dans l'exemple précédent  $eps = 1/4$ .

### Sur une calculatrice

Le système utilisé est la base 10 ( $b = 10$ ). Typiquement, il y a 10 chiffres pour la mantisse et 2 pour l'exposant ( $L = -99$  et  $U = 99$ ).

- Le plus grand nombre machine

$$\tilde{x}_{max} = 9.999999999 \times 10^{+99}$$

- Le plus petit nombre machine

$$\tilde{x}_{min} = -9.999999999 \times 10^{+99}$$

- Le plus petit nombre machine strictement positif

$$\tilde{x}_+ = 1.000000000 \times 10^{-99}$$

Notez qu'avec des nombres dénormalisés, ce nombre serait  $0.000000001 \times 10^{-99}$ , c'est-à-dire avec seulement un chiffre significatif!

Les différences de représentation des nombres flottants d'un ordinateur à un autre obligeaient à reprendre les programmes de calcul scientifique pour les porter d'une machine à une autre. Pour assurer la compatibilité entre les machines, depuis 1985 une norme a été proposée par l'IEEE (Institute of Electrical and Electronics Engineers), c'est la norme 754.

## 1.3 Nombres flottants : le système IEEE 754

Le système IEEE 754 est un standard pour la représentation des nombres à virgule flottante en binaire. Il définit les formats de représentation des nombres à virgule flottante (signe, mantisse, exposant, nombres dénormalisés) et valeurs spéciales (infinis et NaN). Le bit de poids fort est le bit de signe. Cela signifie que si ce bit est à 1, le nombre est négatif, et s'il est à 0, le nombre est positif. Les  $N_e$  bits suivants représentent l'exposant décalé, et les  $N_m$  bits suivants représentent la mantisse.

L'exposant est décalé de  $2^{N_e-1} - 1$  ( $N_e$  représente le nombre de bits de l'exposant), afin de le stocker sous forme d'un nombre non signé.

## 1.3.1 Simple précision

C'est le format 32 bits : 1 bit de signe,  $N_e = 8$  bits d'exposant (-126 à 127), 23 bits de mantisse comme sur le tableau 1.2. L'exposant est décalé de  $2^{N_e-1} - 1 = 2^7 - 1 = 127$ .

|                |                   |                          |
|----------------|-------------------|--------------------------|
| <i>signe s</i> | <i>exposant e</i> | <i>mantisse m</i>        |
| S              | EEEEEEEE          | FFFFFFFFFFFFFFFFFFFFFFFF |
| 0              | 1 à 8             | 9 à 31                   |

Table 1.2: Représentation en simple précision

| $\tilde{x}$   | exposant $e$  | mantisse $m$ |
|---|---------------|--------------|
| $\tilde{x} = 0$ (si $S = 0$ )<br>$\tilde{x} = -0$ (si $S = 1$ )                       | $e = 0$       | $m = 0$      |
| Nombre <i>normalisé</i><br>$\tilde{x} = (-1)^S \times 2^{e-127} \times 1.m$           | $0 < e < 255$ | quelconque   |
| Nombre <i>dénormalisé</i><br>$\tilde{x} = (-1)^S \times 2^{e-126} \times 0.m$         | $e = 0$       | $m \neq 0$   |
| $\tilde{x} = \mathbf{Inf}$ (si $S = 0$ )<br>$\tilde{x} = -\mathbf{Inf}$ (si $S = 1$ ) | $e = 255$     | $m = 0$      |
| $\tilde{x} = \mathbf{NaN}$ ( <i>Not a Number</i> )                                    | $e = 255$     | $m \neq 0$   |

Table 1.3: Représentation en simple précision

- Le *plus petit nombre positif normalisé* différent de zéro, et le *plus grand nombre négatif normalisé* différent de zéro sont :  
 $\pm 2^{-126} = \pm 1,175494351 \times 10^{-38}$
- Le *plus grand nombre positif fini*, et le *plus petit nombre négatif fini* sont :  $\pm(2^{24} - 1) \times 2^{104} = \pm 3,4028235 \times 10^{38}$

| $\tilde{x}$                               | signe | exposant $e$ | mantisse $m$                 | valeur               |
|---|-------|--------------|------------------------------|----------------------|
| zéro                                      | 0     | 0000 0000    | 000 0000 0000 0000 0000 0000 | 0,0                  |
|   | 1     | 0000 0000    | 000 0000 0000 0000 0000 0000 | -0,0                 |
| 1   | 0     | 0111 1111    | 000 0000 0000 0000 0000 0000 | 1,0                  |
| Plus grand nombre normalisé               | 0     | 1111 1110    | 111 1111 1111 1111 1111 1111 | $3,4 \times 10^{38}$ |
| Plus petit $\tilde{x} \geq 0$ normalisé   | 0     | 0000 0001    | 000 0000 0000 0000 0000 0000 | $2^{-126}$           |
| Plus petit $\tilde{x} \geq 0$ dénormalisé | 0     | 0000 0000    | 000 0000 0000 0000 0000 0001 | $2^{-149}$           |
| Infini                                    | 0     | 1111 1111    | 000 0000 0000 0000 0000 0000 | <b>Inf</b>           |
|   | 1     | 1111 1111    | 000 0000 0000 0000 0000 0000 | <b>-Inf</b>          |
| NaN                                       | 0     | 1111 1111    | 010 0000 0000 0000 0000 0000 | <b>NaN</b>           |
| 2   | 0     | 1000 0000    | 000 0000 0000 0000 0000 0000 | 2,0                  |
| exemple 1                                 | 0     | 1000 0001    | 101 0000 0000 0000 0000 0000 | 6,5                  |
| exemple 2                                 | 1     | 1000 0001    | 101 0000 0000 0000 0000 0000 | -6,5                 |
| exemple 3                                 | 1     | 1000 0101    | 110 1101 0100 0000 0000 0000 | -118,625             |

Table 1.4: Exemples en simple précision

Détaillons l'exemple 3 : on code le nombre décimal -118,625 en utilisant le système IEEE 754.

- C'est un nombre négatif, le bit de signe est donc "1",
- On écrit le nombre (sans le signe) en binaire. Nous obtenons 1110110,101,
- On décale la virgule vers la gauche, en laissant seulement un 1 sur sa gauche (nombre flottant normalisé):  $1110110,101 = 1,110110101 \times 2^6$ . La mantisse est la partie à droite de la virgule, remplie de 0 vers la droite pour obtenir 23 bits. Cela donne 1101101010000000000000,
- L'exposant est égal à 6, et nous devons le convertir en binaire et le décaler. Pour le format 32-bit IEEE 754, le décalage est 127. Donc  $6 + 127 = 133$ . En binaire, cela donne 10000101.

### 1.3.2 Double précision

C'est le format 64 bits : 1 bit de signe, 11 bits d'exposant (-1022 à 1023), 52 bits de mantisse. L'exposant est décalé de  $2^{N_e-1} - 1 = 2^{10} - 1 = 1023$ .



Table 1.5: Représentation en double précision

| $\tilde{x}$   | exposant $e$   | mantisse $m$ |
|---|----------------|--------------|
| $\tilde{x} = 0$ (si $S = 0$ )<br>$\tilde{x} = -0$ (si $S = 1$ )                   | $e = 0$        | $m = 0$      |
| Nombre <i>normalisé</i><br>$\tilde{x} = (-1)^S \times 2^{e-1023} \times 1.m$      | $0 < e < 2047$ | quelconque   |
| Nombre <i>dénormalisé</i><br>$\tilde{x} = (-1)^S \times 2^{e-1022} \times 0.m$    | $e = 0$        | $m \neq 0$   |
| $\tilde{x} = \text{Inf}$ (si $S = 0$ )<br>$\tilde{x} = -\text{Inf}$ (si $S = 1$ ) | $e = 2047$     | $m = 0$      |
| $\tilde{x} = \text{NaN}$ ( <i>Not a Number</i> )                                  | $e = 2047$     | $m \neq 0$   |

Table 1.6: Représentation en double précision

- La précision machine est  $eps = 2^{-52}$ .
- Le *plus grand nombre positif fini*, et le *plus petit nombre négatif fini* sont :  $\tilde{x}_{max}^{\pm} = \pm(2^{1024} - 2^{971}) = \pm 1,7976931348623157 \times 10^{308}$
- **Overflow**: L'intervalle de calcul est  $[\tilde{x}_{max}^-, \tilde{x}_{max}^+]$ . Si un calcul produit un nombre  $x$  qui n'est pas dans cet intervalle, on dit qu'il y a *overflow*. La valeur de  $x$  est alors mise à  $\pm \text{Inf}$ . Cela peut arriver au milieu du calcul, même si le résultat final peut être représenté par un nombre machine.
- Le *plus petit nombre positif normalisé* différent de zéro, et le *plus grand nombre négatif normalisé* différent de zéro sont :  
 $\tilde{x}_{min}^{\pm} = \pm 2^{-1022} = \pm 2,22507338585072020 \times 10^{-308}$
- Le système IEEE permet les calculs avec des nombres dénormalisés dans l'intervalle  $[\tilde{x}_{min}^+ * eps, \tilde{x}_{min}^+]$ .
- **Underflow**: Si un calcul produit un nombre positif  $x$  qui est plus petit que  $\tilde{x}_{min}^+ * eps$ , on dit qu'il y a *underflow*. Néanmoins, le calcul ne s'arrête pas dans ce cas, il continue avec la valeur de  $x$  mise à zéro.

### 1.3.3 Matlab

Sous Matlab, les calculs réels sont en double précision par défaut. La fonction `single` peut être utilisée pour convertir les nombres en simple précision. Pour voir la représentation des nombres réels sous Matlab, on peut les afficher au format hexadécimal avec la commande `format hex`.

Le système hexadécimal est celui en base 16 et utilise 16 symboles : 0 à 9 pour représenter les valeurs de 0 à 9, et A, B, C, D, E, F pour représenter les valeurs de 10 à 15. La lecture s'effectue de droite à gauche. La valeur vaut la somme des chiffres affectés de poids correspondant aux puissances successives du nombre 16. Par exemple,  $5EB52_{16}$  vaut  $2 * 16^0 + 5 * 16^1 + 11 * 16^2 + 14 * 16^3 + 5 * 16^4 = 387922$ . Pour passer du binaire au format hexadécimal, c'est facile en regardant la chaîne binaire en groupe de 4 chiffres, et en représentant chaque groupe en un chiffre hexadécimal. Par exemple,

$$\begin{aligned}
 387922 &= 01011110101101010010_2 &= 0101 \ 1110 \ 1011 \ 0101 \ 0010_2 \\
 & &= 5 \ E \ B \ 5 \ 2_{16} \\
 & &= 5EB52_{16}
 \end{aligned}$$

La conversion de l'hexadécimal au binaire est le processus inverse.

## 1.4 Calculs sur les nombres flottants

### 1.4.1 Erreurs d'arrondi

Si  $\tilde{x}$  et  $\tilde{y}$  sont deux nombres machine, alors  $z = \tilde{x} \times \tilde{y}$  ne correspondra pas en général à un nombre machine puisque le produit demande une quantité double de chiffres. Le résultat sera un nombre machine  $\tilde{z}$  proche de  $z$ .

On définit l'*erreur absolue* entre un nombre réel  $x$  et le nombre machine correspondant  $\tilde{x}$  par

$$r_a = |x - \tilde{x}|.$$

L'*erreur relative* entre ces nombres (si  $x \neq 0$ ) est définie par

$$r = \frac{|x - \tilde{x}|}{|x|}.$$

**Opérations machine:** On désigne par *flop* (de l'anglais *floating operation*) une opération élémentaire à virgule flottante (addition, soustraction, multiplication ou division) de l'ordinateur. Sur les calculateurs actuels on peut s'attendre à la précision suivante, obtenue dans les opérations basiques:

$$\tilde{x} \tilde{\oplus} \tilde{y} = (\tilde{x} \oplus \tilde{y})(1 + r)$$

où  $|r| < eps$ , la précision machine, et  $\oplus$  représente l'opération exacte,  $\oplus \in \{+, -, *, /\}$  et  $\tilde{\oplus}$  représente l'opération de l'ordinateur (*flop*).

### 1.4.2 Associativité

L'associativité des opérations élémentaires comme par exemple l'addition:

$$(x + y) + z = x + (y + z),$$

n'est plus valide en arithmétique finie. Par exemple, avec 6 chiffres de précision, si on prend les trois nombres

$$x = 1.23456e-3, \quad y = 1.00000e0, \quad z = -y,$$

on obtient  $(x + y) + z = (0.00123 + 1.00000e0) - 1.00000e0 = 1.23000e-3$  alors que  $x + (y + z) = x = 1.23456e-3$ . Il est donc essentiel de considérer l'ordre des opérations et faire attention où l'on met les parenthèses.

### 1.4.3 Monotonie

Supposons que l'on a une fonction  $f$  strictement croissante sur un intervalle  $[a, b]$ . Peut-on assurer en arithmétique finie que

$$\tilde{x} < \tilde{y} \Rightarrow f(\tilde{x}) < f(\tilde{y})?$$

En général non. Dans la norme IEEE les *fonctions standard* sont implémentées de façon à respecter la monotonie (mais pas la stricte monotonie).

### 1.4.4 Erreurs d'annulation

Ce sont les erreurs dues à l'annulation numérique de chiffres significatifs, quand les nombres ne sont représentés qu'avec une quantité finie de chiffres, comme les nombres machine. Il est donc important en pratique d'être attentif aux signes dans les expressions, comme l'exemple suivant le montre :

Exemple : on cherche à évaluer sur l'ordinateur de façon précise, pour de petites valeurs de  $x$ , la fonction

$$f(x) = \frac{1}{1 - \sqrt{1 - x^2}}.$$

Pour  $|x| < \sqrt{eps}$ , on risque d'avoir le nombre  $\sqrt{1 - x^2}$  remplacé par 1, par la machine, et donc lors du calcul de  $f(x)$ , on risque d'effectuer une division par 0. Par exemple, pour  $x = \frac{\sqrt{eps}}{2}$ , on obtient :

```

> f=@(x) 1./(1-sqrt(1-x.^2));
> f(0.5*sqrt(eps))

ans =

```

Inf

On ne peut donc pas évaluer précisément  $f(x)$  sur l'ordinateur dans ce cas. Maintenant, si on multiplie dans  $f(x)$  le numérateur et le dénominateur par  $1 + \sqrt{1 - x^2}$ , on obtient

$$f(x) = \frac{1 + \sqrt{1 - x^2}}{x^2}$$

Cette fois, on peut évaluer  $f(x)$  de façon précise :

```

>> f=@(x) (1+sqrt(1-x.^2))./x.^2;
>> f(0.5*sqrt(eps))

ans =

```

3.6029e+16

C'est ce qui se passe dans le cas du calcul de  $\pi$  avec l'algorithme naïf 1.1. En utilisant la formule

$$\sin \frac{\alpha_n}{2} = \sqrt{\frac{1 - \cos \alpha_n}{2}} = \sqrt{\frac{1 - \sqrt{1 - \sin^2 \alpha_n}}{2}},$$

comme  $\sin \alpha_n \rightarrow 0$ , le numérateur à droite est de la forme

$$1 - \sqrt{1 - \varepsilon^2}, \quad \text{avec } \varepsilon = \sin \alpha_n \text{ petit,}$$

donc sujet aux erreurs d'annulation. Pour y palier, il faut reformuler les équations de façon à s'affranchir des erreurs d'annulations, par exemple en multipliant le numérateur et le dénominateur par  $(1 + \sqrt{1 - \sin^2 \alpha_n})$ .

$$\begin{aligned} \sin \frac{\alpha_n}{2} &= \sqrt{\frac{1 - \sqrt{1 - \sin^2 \alpha_n}}{2}} = \sqrt{\frac{(1 - \sqrt{1 - \sin^2 \alpha_n})(1 + \sqrt{1 - \sin^2 \alpha_n})}{2(1 + \sqrt{1 - \sin^2 \alpha_n})}} \\ &= \sqrt{\frac{1 - (1 - \sin^2 \alpha_n)}{2(1 + \sqrt{1 - \sin^2 \alpha_n})}} = \frac{\sin \alpha_n}{\sqrt{2(1 + \sqrt{1 - \sin^2 \alpha_n})}}. \end{aligned}$$

On peut alors écrire l'Algorithme 1.2 correspondant au calcul de  $\pi$  avec cette nouvelle formule. En utilisant cet algorithme on obtient les résultats donnés en Table 1.7.

---

**Algorithm 1.2** Algorithme de calcul de  $\pi$ , version stable

---

- |   |   |
|---|---|
| 1: $s \leftarrow 1, n \leftarrow 4,$                              | ▷ Initialisations                       |
| 2: <b>Tantque</b> $s > 1e - 10$ <b>faire</b>                      | ▷ Arrêt si $s = \sin(\alpha)$ est petit |
| 3: $s \leftarrow s/\text{sqrt}(2 * (1 + \text{sqrt}(1 - s * s)))$ | ▷ nouvelle valeur de $\sin(\alpha/2)$   |
| 4: $n \leftarrow 2 * n$   | ▷ nouvelle valeur de $n$                |
| 5: $A \leftarrow (n/2) * s$                                       | ▷ nouvelle valeur de l'aire du polygone |
| 6: <b>Fin</b>   |   |
- 

## 1.5 Quelques catastrophes dues à l'arithmétique flottante

Il y a un petit nombre "connu" de catastrophes dans la vie réelle qui sont attribuables à une mauvaise gestion de l'arithmétique des ordinateurs (erreurs d'arrondis, d'annulation), voir [6]. Dans le premier exemple ci-dessous, cela a abouti à la perte de vies humaines.

| $n$         | $A_n$            | $ A_n - \pi $ | $\sin(\alpha_n)$ |
|-------------|------------------|---------------|------------------|
| 4           | 2.00000000000000 | 1.141593e+00  | 1.000000e+00     |
| 8           | 2.82842712474619 | 3.131655e-01  | 7.071068e-01     |
| 16          | 3.06146745892072 | 8.012519e-02  | 3.826834e-01     |
| 32          | 3.12144515225805 | 2.014750e-02  | 1.950903e-01     |
| 64          | 3.13654849054594 | 5.044163e-03  | 9.801714e-02     |
| 128         | 3.14033115695475 | 1.261497e-03  | 4.906767e-02     |
| 256         | 3.14127725093277 | 3.154027e-04  | 2.454123e-02     |
| 512         | 3.14151380114430 | 7.885245e-05  | 1.227154e-02     |
| 1024        | 3.14157294036709 | 1.971322e-05  | 6.135885e-03     |
| 2048        | 3.14158772527716 | 4.928313e-06  | 3.067957e-03     |
| 4096        | 3.14159142151120 | 1.232079e-06  | 1.533980e-03     |
| 8192        | 3.14159234557012 | 3.080197e-07  | 7.669903e-04     |
| 16384       | 3.14159257658487 | 7.700492e-08  | 3.834952e-04     |
| 32768       | 3.14159263433856 | 1.925123e-08  | 1.917476e-04     |
| 65536       | 3.14159264877699 | 4.812807e-09  | 9.587380e-05     |
| 131072      | 3.14159265238659 | 1.203202e-09  | 4.793690e-05     |
| 262144      | 3.14159265328899 | 3.008003e-10  | 2.396845e-05     |
| 524288      | 3.14159265351459 | 7.519985e-11  | 1.198422e-05     |
| 1048576     | 3.14159265357099 | 1.879963e-11  | 5.992112e-06     |
| 2097152     | 3.14159265358509 | 4.699352e-12  | 2.996056e-06     |
| 4194304     | 3.14159265358862 | 1.174172e-12  | 1.498028e-06     |
| 8388608     | 3.14159265358950 | 2.926548e-13  | 7.490141e-07     |
| 16777216    | 3.14159265358972 | 7.238654e-14  | 3.745070e-07     |
| 33554432    | 3.14159265358978 | 1.731948e-14  | 1.872535e-07     |
| 67108864    | 3.14159265358979 | 3.552714e-15  | 9.362676e-08     |
| 134217728   | 3.14159265358979 | 0.000000e+00  | 4.681338e-08     |
| 268435456   | 3.14159265358979 | 8.881784e-16  | 2.340669e-08     |
| 536870912   | 3.14159265358979 | 1.332268e-15  | 1.170334e-08     |
| 1073741824  | 3.14159265358979 | 1.332268e-15  | 5.851672e-09     |
| 2147483648  | 3.14159265358979 | 1.332268e-15  | 2.925836e-09     |
| 4294967296  | 3.14159265358979 | 1.332268e-15  | 1.462918e-09     |
| 8589934592  | 3.14159265358979 | 1.332268e-15  | 7.314590e-10     |
| 17179869184 | 3.14159265358979 | 1.332268e-15  | 3.657295e-10     |
| 34359738368 | 3.14159265358979 | 1.332268e-15  | 1.828648e-10     |
| 68719476736 | 3.14159265358979 | 1.332268e-15  | 9.143238e-11     |

Table 1.7: Calcul de  $\pi$  avec l'algorithme stable

### Missile Patriot

En février 1991, pendant la Guerre du Golfe, une batterie américaine de missiles Patriot, à Dharan (Arabie Saoudite), a échoué dans l'interception d'un missile Scud irakien. Le Scud a frappé un baraquement de l'armée américaine et a tué 28 soldats. La commission d'enquête a conclu à un calcul incorrect du temps de parcours, dû à un problème d'arrondi. Les nombres étaient représentés en virgule fixe sur 24 bits, donc 24 chiffres binaires. Le temps était compté par l'horloge interne du système en 1/10 de seconde. Malheureusement, 1/10 n'a pas d'écriture finie dans le système binaire :  $1/10 = 0,1$  (dans le système décimal) =  $0,0001100110011001100110011\dots$  (dans le système binaire). L'ordinateur de bord arrondissait 1/10 à 24 chiffres, d'où une petite erreur dans le décompte du temps pour chaque 1/10 de seconde. Au moment de l'attaque, la batterie de missile Patriot était allumée depuis environ 100 heures, ce qui avait entraîné une accumulation des erreurs d'arrondi de 0,34 s. Pendant ce temps, un missile Scud parcourt environ 500 m, ce qui explique que le Patriot soit passé à côté de sa cible. Ce qu'il aurait fallu faire c'était redémarrer régulièrement le système de guidage du missile.

### Explosion d'Ariane 5

Le 4 juin 1996, une fusée Ariane 5, à son premier lancement, a explosé 40 secondes après l'allumage. La fusée et son chargement avaient coûté 500 millions de dollars. La commission d'enquête a rendu son

rapport au bout de deux semaines. Il s'agissait d'une erreur de programmation dans le système inertielle de référence. À un moment donné, un nombre codé en virgule flottante sur 64 bits (qui représentait la vitesse horizontale de la fusée par rapport à la plate-forme de tir) était converti en un entier sur 16 bits. Malheureusement, le nombre en question était plus grand que 32768 (overflow), le plus grand entier que l'on peut coder sur 16 bits, et la conversion a été incorrecte.

### Bourse de Vancouver

Un autre exemple où les erreurs de calcul ont conduit à une erreur notable est le cas de l'indice de la Bourse de Vancouver. En 1982, elle a créé un nouvel indice avec une valeur nominale de 1000. Après chaque transaction boursière, cet indice était recalculé et tronqué après le troisième chiffre décimal et, au bout de 22 mois, la valeur obtenue était 524,881, alors que la valeur correcte était 1098.811. Cette différence s'explique par le fait que toutes les erreurs d'arrondi étaient dans le même sens : l'opération de troncature diminuait à chaque fois la valeur de l'indice.



# Chapitre 2

## Résolution de systèmes non linéaires



(a) *Scipione del Ferro* 1465-1526, mathématicien italien



(b) *Niccolo Fontana* 1499-1557, mathématicien italien



(c) *Ludovico Ferrari* 1522-1565, mathématicien italien



(d) *Paolo Ruffini* 1765-1822, mathématicien italien



(e) *Bernard Bolzano* 1781-1848, mathématicien, logicien, philosophe et théologien de l'empire d'Autriche



(f) *Niels Henrik Abel* 1802-1829, mathématicien norvégien

Un problème *simple* comme la recherche des zéros/racines d'un polynôme n'est pas ... *simple*. Depuis tout petit, on sait trouver les racines d'un polynôme de degré 2 :  $a_2x^2 + a_1x + a_0 = 0$ . Quid des racines de polynômes de degré plus élevé?



- **degré 2** : Babyloniens en 1600 avant J.-C.
- **degré 3** : *Scipione del Ferro* (1465-1526, mathématicien italien) et *Niccolo Fontana* (1499-1557, mathématicien italien)
- **degré 4** : *Lodovico Ferrari* (1522-1565, mathématicien italien)
- **degré 5** : *Paolo Ruffini* (1765-1822, mathématicien italien) en 1799, *Niels Henrik Abel* (1802-1829, mathématicien norvégien) en 1824, montrent qu'il n'existe **pas de solution analytique**.

L'objectif de ce chapitre est de rechercher **numériquement** les zéros/racines d'une fonction ou d'un système d'équations lorsqu'ils existent :

- Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ , trouver  $x \in [a, b]$  tel que  $f(x) = 0$ , étudier en section 2.1
- Soit  $f : \Omega \subset \mathbb{K}^n \rightarrow \mathbb{K}^n$ , trouver  $\mathbf{x} \in \Omega$  tel que  $f(\mathbf{x}) = 0$  ( $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{K} = \mathbb{C}$ ). étudier en section 2.3

Avant de commencer une petite pique de rappels par la lecture de l'Annexe B.1

## 2.1 Recherche des zéros d'une fonction

Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une fonction **continue**. On cherche à déterminer les zéros de  $f$  ou plus précisément l'ensemble des  $x \in [a, b]$  tels que  $f(x) = 0$ .

Dans un premier temps, on étudiera la **méthode de dichotomie** qui est *assez naturelle*, puis on analysera plusieurs algorithmes liés à la méthode du point fixe.

### 2.1.1 Méthode de dichotomie ou de bisection

#### Principe et résultats

##### NOTE

**principe de la méthode de dichotomie** : Soit  $I$  un intervalle contenant un unique zéro de la fonction  $f$ , on le divise par son milieu en deux intervalles et on détermine lequel des deux contient le zéro. On itère ce processus sur le nouvel intervalle.

Plus précisément, on suppose que la fonction  $f$  vérifie  $f(a)f(b) < 0$ . D'après le théorème des valeurs intermédiaires ou le théorème de Bolzano, il existe alors un  $\xi \in ]a, b[$  tel que  $f(\xi) = 0$ .

On note  $I^{(0)} = ]a, b[$  et  $x_0 = (a+b)/2$ . Si  $f(x_0) = 0$  alors on a fini! Supposons  $f(x_0) \neq 0$ , alors  $\xi$  appartient à  $]a, x_0[$  si  $f(a)f(x_0) < 0$ , sinon il appartient à  $]x_0, b[$ . On vient donc de déterminer une méthode permettant de diviser par 2 la longueur de l'intervalle de recherche de  $\xi$ . On peut bien évidemment itérer ce principe en définissant les trois suites  $(a_k)_{k \in \mathbb{N}}$ ,  $(b_k)_{k \in \mathbb{N}}$  et  $(x_k)_{k \in \mathbb{N}}$  par

- $a_0 = a, b_0 = b$  et  $x_0 = \frac{a+b}{2}$ ,

- $\forall k \in \mathbb{N}$ ,

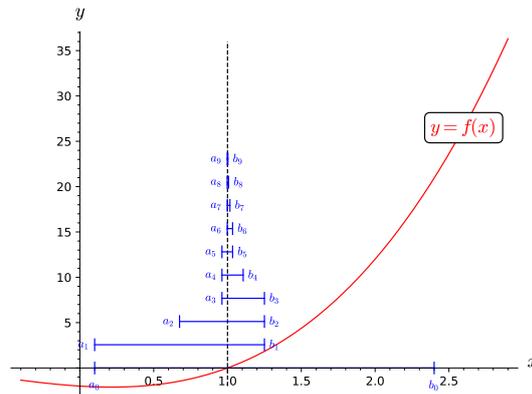
$$\begin{cases} a_{k+1} = b_{k+1} = x_k & \text{si } f(x_k) = 0, \\ a_{k+1} = x_k, b_{k+1} = b_k & \text{si } f(b_k)f(x_k) < 0, \\ a_{k+1} = a_k, b_{k+1} = x_k & \text{si } f(a_k)f(x_k) < 0. \end{cases}$$

et

$$x_{k+1} = (a_{k+1} + b_{k+1})/2.$$

Par construction on a  $a_k \leq x_k \leq b_k$ .

En Figure 2.2, on représente les intervalles successifs obtenus par la méthode de dichotomie pour  $a = 0.1$ ,  $b = 2.4$  et  $f : x \mapsto (x+2)(x+1)(x-1)$ .

Figure 2.2: Méthode de dichotomie:  $f(x) = (x + 2)(x + 1)(x - 1)$ ,**EXERCICE 2.1.1**

On suppose que la fonction  $f$  est continue sur  $[a, b]$ , vérifie  $f(a)f(b) < 0$  et qu'il existe un unique  $\alpha \in ]a, b[$  tel que  $f(\alpha) = 0$ .

**Q. 1**

- Montrer que les suites  $(a_k)$  et  $(b_k)$  convergent vers  $\alpha$ .
- En déduire que la suite  $(x_k)$  converge vers  $\alpha$ .

**Q. 2**

- Montrer que pour tout  $k \in \mathbb{N}$ ,  $|x_k - \alpha| \leq \frac{b-a}{2^{k+1}}$ .
- Soit  $\epsilon > 0$ . En déduire que si  $k \geq \frac{\log(\frac{b-a}{\epsilon})}{\log(2)} - 1$  alors  $|x_k - \alpha| \leq \epsilon$ .

**Correction 2.1.1****R. 1**

- Supposons qu'il existe  $k \in \mathbb{N}$  tel que  $f(x_k) = 0$ , (i.e.  $x_k = \alpha$  car  $x_k \in [a, b]$ ) alors par construction  $a_{k+i} = b_{k+i} = x_{k+i} = \alpha$  pour tout  $i \in \mathbb{N}^*$ . Ceci assure la convergence des 3 suites vers  $\alpha$ .  
Supposons maintenant que  $\forall k \in \mathbb{N}$ ,  $f(x_k) \neq 0$ . Par construction, nous avons  $a_k \leq a_{k+1} \leq b$ ,  $a \leq b_{k+1} \leq b_k$  et  $a_k \leq b_k$ . La suite  $(a_k)$  est convergente car elle est croissante et majorée. La suite  $(b_k)$  est décroissante et minorée : elle est donc convergente. De plus  $0 \leq b_k - a_k \leq \frac{b_{k-1} - a_{k-1}}{2}$  et donc  $0 \leq b_k - a_k \leq \frac{b-a}{2^k}$ . On en déduit que les suites  $(a_k)$  et  $(b_k)$  ont même limite. Comme par construction,  $\forall k \in \mathbb{N}$ ,  $\alpha \in [a_k, b_k]$  ceci entraîne que  $\alpha$  est la limite de ces 2 suites.
- Par construction,  $\forall k \in \mathbb{N}$ ,  $a_k \leq x_k \leq b_k$ . D'après le théorème des gendarmes, les suites  $(a_k)$  et  $(b_k)$  convergeant vers  $\alpha$ , on obtient la convergence de la suite  $(x_k)$  vers  $\alpha$ .

**R. 2**

- On a  $\forall k \in \mathbb{N}$ ,  $x_k = \frac{a_k + b_k}{2}$  et  $a_k \leq \alpha \leq b_k$  d'où  $|x_k - \alpha| \leq \frac{b_k - a_k}{2}$ . Ce qui donne

$$|x_k - \alpha| \leq \frac{b - a}{2^{k+1}}.$$

- Pour avoir  $|x_k - \alpha| \leq \epsilon$ , il suffit d'avoir  $\frac{b-a}{2^{k+1}} \leq \epsilon$ , et la fonction  $\log$  étant croissante on obtient

$$k \geq \frac{\log(\frac{b-a}{\epsilon})}{\log(2)} - 1.$$

~~~~~ Fin correction ~~~~~

On peut alors écrire la proposition suivante :

**Proposition 2.1.1** (Méthode de dichotomie/bissection). Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une fonction continue vérifiant  $f(a)f(b) < 0$  et admettant  $\alpha \in ]a, b[$  comme **unique** solution de  $f(x) = 0$ . Alors la suite  $(x_k)_{k \in \mathbb{N}}$  définie par la méthode de dichotomie converge vers  $\alpha$  et

$$|x_k - \alpha| \leq \frac{b - a}{2^{k+1}}, \quad \forall k \in \mathbb{N}.$$

On a alors  $\forall \epsilon > 0, \forall k \geq \frac{\log(\frac{b-a}{\epsilon})}{\log(2)} - 1$

$$|x_k - \alpha| \leq \epsilon.$$

A partir de ce résultat, on va proposer plusieurs variantes de l'algorithme de dichotomie.

On note tout d'abord que pour déterminer  $\alpha$  il faut calculer la limite d'une suite et donc une infinité de termes! Numériquement et algorithmiquement, on se limite donc à déterminer une approximation de  $\alpha$ . La proposition 2.1.1 permet d'obtenir une approximation  $\alpha_\epsilon$  de  $\alpha$  en un nombre fini d'itérations avec une précision  $\epsilon$  donnée: on choisit  $\alpha_\epsilon = x_{k_{\min}}$  avec  $k_{\min} = E(\frac{\log(\frac{b-a}{\epsilon})}{\log(2)})$  où  $E(\cdot)$  est la fonction *partie entière*.

### Algorithmique

Tout d'abord nous allons poser "correctement" le problème pour lever toute ambiguïté. En effet, si le problème est *rechercher une racine de  $f$  sur l'intervalle  $[a, b]$  par la méthode de dichotomie*, dois-je uniquement rechercher une approximation d'une racine ou calculer la suite  $(x_k)$  ou l'ensemble des suites  $(x_k)$ ,  $(a_k)$  et  $(b_k)$ ? C'est ce que nous appellerons le(s) **résultat(s)/objectif(s)** de l'algorithme. L'écriture d'un algorithme est fortement corrélée aux objectifs souhaités.

Sauf note contraire, on choisira par la suite comme objectif de déterminer une approximation de la racine  $\alpha$  de  $f$  :

**Résultat :**  $\alpha_\epsilon$  : un réel tel que  $|\alpha_\epsilon - \alpha| \leq \epsilon$ .

Ensuite, pour aboutir à cet objectif on détermine les données (avec hypothèses) nécessaires et suffisantes :

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  vérifiant les hypothèses de la proposition 2.1.1,  
 $\epsilon$  : un réel strictement positif.

On peut alors noter (formellement pour le moment) **Dichotomie** la fonction permettant, à partir des données  $f, a, b$ , et  $\epsilon$ , de trouver  $\alpha_\epsilon$ . Cette fonction algorithmique aura la syntaxe suivante :

$$\alpha_\epsilon \leftarrow \text{Dichotomie}(f, a, b, \epsilon)$$

Nous allons maintenant proposer une manière d'aborder l'écriture de cette fonction dans un langage algorithmique présenté au chapitre A. On doit s'affranchir de tout (ou presque) symbolismes mathématiques pour s'approcher au plus près des langages de programmation. Par exemple, la notation mathématique  $(x_k)_{k=0}^{10}$  pour noter les 11 premiers itérés d'une suite n'est pas directement disponible dans les langages de programmations : on peut par exemple passer par un tableau  $X$  de dimension 11 (au moins) pour stocker les valeurs de la suite. Suivant les langages, l'accès aux composantes d'un tableau diffère : sous Matlab/Octave l'accès au 1er élément d'un tableau  $X$  se fait par  $X(1)$  et en C/C++/Python par  $X[0]$ . Lors de l'écriture d'un algorithme, nous utiliserons l'opérateur  $()$  ou(exclusif)  $[]$  pour accéder aux différentes composantes d'un tableau :  $()$  si le 1er élément est d'indice 1 ou  $[]$  si le 1er élément est d'indice 0.

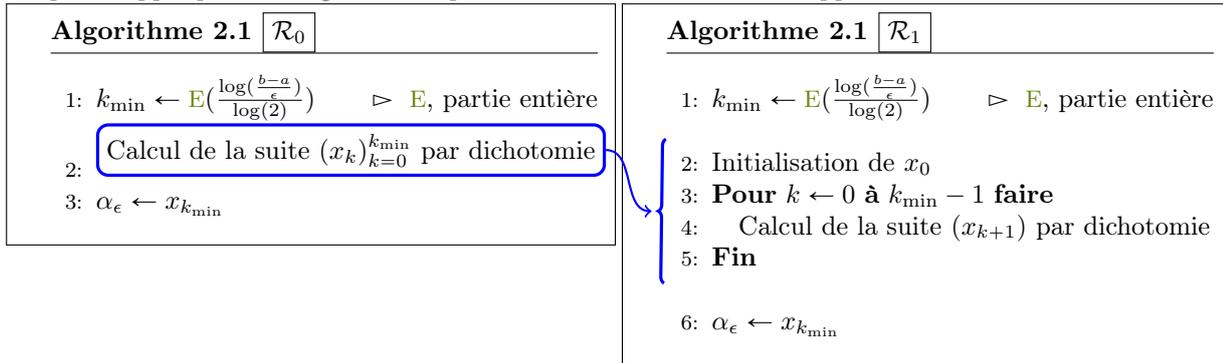
Pour écrire un algorithme non trivial, nous utiliserons une *technique de raffinements d'algorithme* (voir chapitre A) : par raffinements successifs, nous allons écrire des algorithmes **équivalents** pour aboutir au final à un algorithme n'utilisant que

- des instructions élémentaires (affectation, addition, somme, ...)
- des instructions composées (conditionnelle, boucles pour, tantque, ...)
- des fonctions usuelles, mathématiques (`cos`, `exp`, `E` partie entière, ...), entrées/sorties, ...

présentent dans tous les langages.

L'idée est donc de partir d'un algorithme formel facile à comprendre puis de le détailler de plus en plus au fur et à mesure des raffinements. Le passage entre deux raffinements successifs doit être simple.

Un petit rappel pour les algorithmes qui suivent : les données sont supposées ... données!



Entre les raffinements  $\mathcal{R}_0$  et  $\mathcal{R}_1$ , nous avons juste décrit le principe de calcul de toute suite récurrente d'ordre 1.

Pour faciliter la lecture, nous rappelons la méthode de dichotomie :

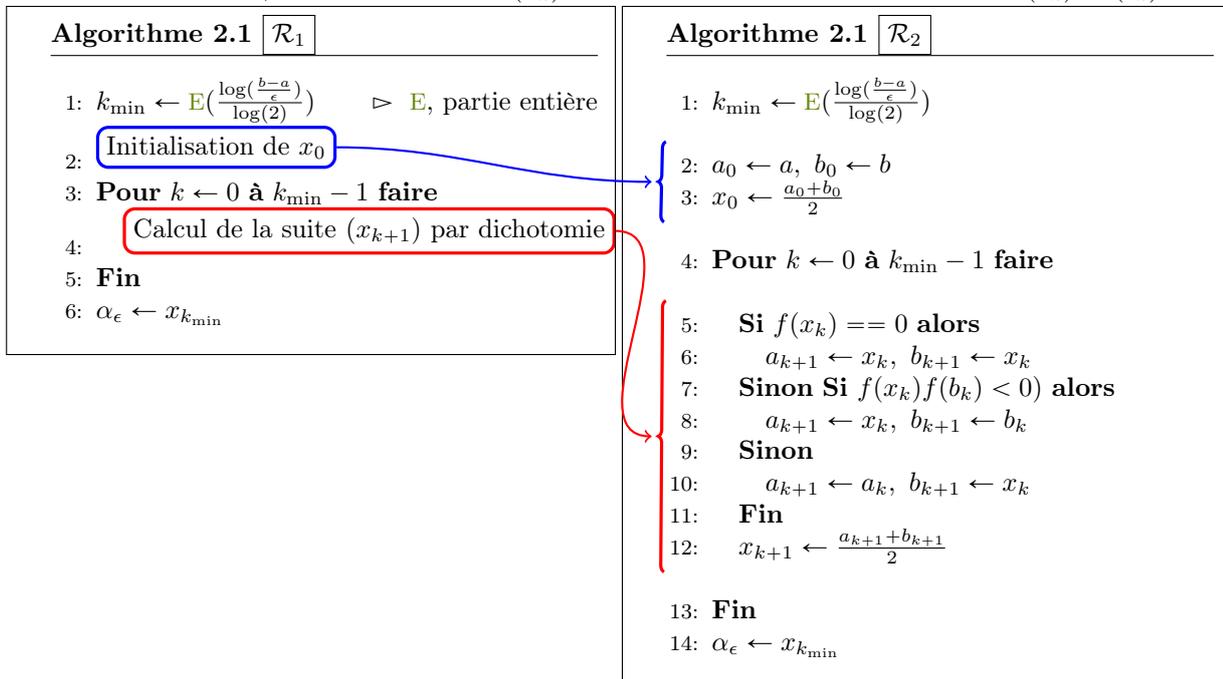
- $a_0 = a, b_0 = b$  et  $x_0 = \frac{a+b}{2}$ ,
- $\forall k \in \llbracket 0, k_{\min} - 1 \rrbracket$ ,

$$\begin{cases} a_{k+1} = b_{k+1} = x_k & \text{si } f(x_k) = 0, \\ a_{k+1} = x_k, b_{k+1} = b_k & \text{si } f(b_k)f(x_k) < 0, \\ a_{k+1} = a_k, b_{k+1} = x_k & \text{si } f(a_k)f(x_k) < 0. \end{cases}$$

et

$$x_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$$

Ecrit sous cette forme, le calcul de la suite  $(x_k)$  nécessite le calcul simultané des suites  $(a_k)$  et  $(b_k)$ .



La transcription de ce raffinement dans un langage de programmation n'est pas forcément triviale pour tout le monde. Quid de  $a_k, \epsilon, \dots$  qui sont syntaxiquement incorrectes dans un langage de programmation? Le cas du  $\epsilon$  est très simple : les lettres grecques étant prohibées, on la remplacera par exemple par eps. Pour les suites  $(a_k)$ ,  $(b_k)$  et  $(x_k)$ , tronquées à  $k_{\min}$ , on pourra utiliser des tableaux (vecteurs) de  $k_{\min} + 1$  réels notés  $\mathbf{A}$ ,  $\mathbf{B}$  et  $\mathbf{X}$  qui contiendront respectivement l'ensemble des valeurs  $a_k, b_k$  et  $x_k$  pour  $0 \leq k \leq k_{\min}$ . Plus précisément nous pourrions utiliser les opérateurs  $()$  (indice des tableaux commence à 1) ou  $[]$  (indice des tableaux commence à 0) pour accéder aux différents éléments des tableaux et nous

aurons par convention  $\mathbf{A}(k+1) = \mathbf{A}[k] = a_k$ ,  $\forall k \in \llbracket 0, k_{\min} \rrbracket$ . Par la suite nous utiliserons les opérateurs  $()$  et nous aurons alors les relations suivantes entre les notations *algorithmiques* et *mathématiques*.

$$\forall k \in \llbracket 0, k_{\min} \rrbracket, \mathbf{A}(k+1) = a_k, \mathbf{B}(k+1) = b_k \text{ et } \mathbf{X}(k+1) = x_k.$$

En utilisant ces changements de notations nous obtenons (enfin) l'Algorithme 2.1.

---

**Algorithm 2.1** Méthode de dichotomie : version 1

---

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  vérifiant  
les hypothèses de la proposition 2.1.1,  
 $\text{eps}$  : un réel strictement positif.

**Résultat :**  $x$  : un réel tel que  $|x - \alpha| \leq \text{eps}$ .

```

1: Fonction  $x \leftarrow \text{Dichotomie1}( f, a, b, \text{eps} )$ 
2:    $k_{\min} \leftarrow \lceil \log((b-a)/\text{eps})/\log(2) \rceil$ 
3:    $\mathbf{A}, \mathbf{B}, \mathbf{X} \in \mathbb{R}^{k_{\min}+1}$   $\triangleright \mathbf{A}(k+1)$  contiendra  $a_k, \dots$ 
4:    $\mathbf{A}(1) \leftarrow a, \mathbf{B}(1) \leftarrow b, \mathbf{X}(1) \leftarrow (a+b)/2$ 
5:   Pour  $k \leftarrow 1$  à  $k_{\min}$  faire
6:     Si  $f(\mathbf{X}(k)) == 0$  alors
7:        $\mathbf{A}(k+1) \leftarrow \mathbf{X}(k), \mathbf{B}(k+1) \leftarrow \mathbf{X}(k)$ 
8:     Sinon Si  $f(\mathbf{B}(k))f(\mathbf{X}(k)) < 0$  alors
9:        $\mathbf{A}(k+1) \leftarrow \mathbf{X}(k), \mathbf{B}(k+1) \leftarrow \mathbf{B}(k)$ 
10:    Sinon
11:       $\mathbf{A}(k+1) \leftarrow \mathbf{A}(k), \mathbf{B}(k+1) \leftarrow \mathbf{X}(k)$ 
12:    Fin
13:     $\mathbf{X}(k+1) \leftarrow (\mathbf{A}(k+1) + \mathbf{B}(k+1))/2$ 
14:  Fin
15:   $x \leftarrow \mathbf{X}(k_{\min} + 1)$ 
16: Fin

```

---

Des codes Matlab/Octave et C correspondant de cette fonction sont données en Annexe B.4.1, respectivement en Listing 18 et 28. Les Listings 8 et 25 correspondent respectivement à un *script* Matlab et un *main* C utilisant cette fonction.

On peut noter qu'une réécriture de la méthode de dichotomie permet d'éviter d'utiliser les deux suites  $(a_k)$  et  $(b_k)$ . En effet, on a

- $A = a, B = b$  et  $x_0 = \frac{A+B}{2}$ ,

- $\forall k \in \llbracket 0, k_{\min} - 1 \rrbracket$ ,

$$\begin{cases} A = B = x_k & \text{si } f(x_k) = 0, \\ A = x_k, B \text{ inchangé} & \text{si } f(B)f(x_k) < 0, \\ B = x_k, A \text{ inchangé} & \text{sinon (i.e. } f(A)f(x_k) < 0.) \end{cases}$$

et

$$x_{k+1} = \frac{A+B}{2}$$

On utilise, ces formules dans l'Algorithme 2.2.

**Algorithm 2.2** Méthode de dichotomie : version 2

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f : [a, b] \subset \mathbb{R} \longrightarrow \mathbb{R}$  vérifiant  
les hypothèses de la proposition 2.1.1,  
 $\text{eps}$  : un réel strictement positif.

**Résultat :**  $x$  : un réel tel que  $|x - \alpha| \leq \text{eps}$ .

```

1: Fonction  $x \leftarrow \text{Dichotomie2}(f, a, b, \text{eps})$ 
2:  $k_{\min} \leftarrow \lceil \log((b - a)/\text{eps}) / \log(2) \rceil$ 
3:  $\mathbf{X} \in \mathbb{R}^{k_{\min} + 1}$  ▷  $\mathbf{X}(k + 1)$  contiendra  $x_k, \dots$ 
4:  $A \leftarrow a, B \leftarrow b, \mathbf{X}(1) \leftarrow (A + B)/2$ 
5: Pour  $k \leftarrow 1$  à  $k_{\min}$  faire
6:   Si  $f(\mathbf{X}(k)) == 0$  alors
7:      $A \leftarrow \mathbf{X}(k), B \leftarrow \mathbf{X}(k)$ 
8:   Sinon Si  $f(B)f(\mathbf{X}(k)) < 0$  alors
9:      $A \leftarrow \mathbf{X}(k)$  ▷ B inchangé
10:  Sinon
11:     $B \leftarrow \mathbf{X}(k)$  ▷ A inchangé
12:  Fin
13:   $\mathbf{X}(k + 1) \leftarrow (A + B)/2$ 
14: Fin
15:  $x \leftarrow \mathbf{X}(k_{\min} + 1)$ 
16: Fin

```

Si notre objectif n'est que de calculer  $\alpha_\epsilon$ , on peut, sur le même principe, s'affranchir d'utiliser un tableau pour stocker tous les termes de la suite  $x_k$  : seul le dernier nous intéresse. On propose dans l'Algorithme 2.3, une version n'utilisant pas de tableaux.

**Algorithm 2.3** Méthode de dichotomie : version 3

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f : [a, b] \subset \mathbb{R} \longrightarrow \mathbb{R}$  vérifiant  
les hypothèses de la proposition 2.1.1,  
 $\text{eps}$  : un réel strictement positif.

**Résultat :**  $x$  : un réel tel que  $|x - \alpha| \leq \text{eps}$ .

```

1: Fonction  $x \leftarrow \text{Dichotomie3}(f, a, b, \text{eps})$ 
2:  $k_{\min} \leftarrow \lceil \log((b - a)/\text{eps}) / \log(2) \rceil$ 
3:  $A, B \in \mathbb{R}$ 
4:  $A \leftarrow a, B \leftarrow b, x \leftarrow (a + b)/2$ 
5: Pour  $k \leftarrow 1$  à  $k_{\min}$  faire
6:   Si  $f(x) == 0$  alors
7:      $A \leftarrow x, B \leftarrow x$ 
8:   Sinon Si  $f(B)f(x) < 0$  alors
9:      $A \leftarrow x$  ▷ B inchangé
10:  Sinon
11:     $B \leftarrow x$  ▷ A inchangé
12:  Fin
13:   $x \leftarrow (A + B)/2$ 
14: Fin
15: Fin

```

Une autre écriture est possible sans utiliser le nombre  $k_{\min}$  et donc en utilisant une boucle **Tantque** (pour les anglophobes!) ou **While** (pour les anglophiles!). Celle-ci est présentée dans l'Algorithme 2.4

---

**Algorithm 2.4** Méthode de dichotomie : version 4

---

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$  et  $f(a)f(b) < 0$   
 $\text{eps}$  : un réel strictement positif.  
**Résultat :**  $x$  : un réel tel que  $|x - \alpha| \leq \text{eps}$ .

```

1: Fonction  $x \leftarrow \text{Dichotomie4}(f, a, b, \text{eps})$ 
2:    $A, B \in \mathbb{R}$ 
3:    $A \leftarrow a, B \leftarrow b, x \leftarrow (a + b)/2$ 
4:   Tantque  $|x - A| > \text{eps}$  faire
5:     Si  $f(x) == 0$  alors
6:        $A \leftarrow x, B \leftarrow x$ 
7:     Sinon Si  $f(B)f(x) < 0$  alors
8:        $A \leftarrow x$  ▷  $B$  inchangé
9:     Sinon
10:       $B \leftarrow x$  ▷  $A$  inchangé
11:   Fin
12:    $x \leftarrow (A + B)/2$ 
13: Fin
14: Fin

```

---

Que pensez-vous de l'algorithme suivant ?

---

**Algorithm 2.5** Méthode de dichotomie : version 5

---

**Données :**  $a, b$  : deux réels  $a < b$ ,  
 $f$  :  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$  et  $f(a)f(b) < 0$ .  
**Résultat :**  $x$  : un réel tel que  $f(x) = 0$ .

```

1: Fonction  $x \leftarrow \text{Dichotomie5}(f, a, b)$ 
2:    $A, B \in \mathbb{R}$ 
3:    $A \leftarrow a, B \leftarrow b, x \leftarrow (a + b)/2, xp \leftarrow a$ 
4:   Tantque  $x \sim xp$  faire
5:     Si  $f(B)f(x) < 0$  alors
6:        $A \leftarrow x$  ▷  $B$  inchangé
7:     Sinon
8:        $B \leftarrow x$  ▷  $A$  inchangé
9:     Fin
10:     $xp \leftarrow x$ 
11:     $x \leftarrow (A + B)/2$ 
12:  Fin
13: Fin

```

---

Des codes Matlab/Octave et C correspondant à cette fonction sont données en Annexe B.4.1, respectivement en Listing 15 et 16. Les Listings 8 et 17 correspondent respectivement à un *script* Matlab et un *main* C utilisant cette fonction.

## 2.2 Points fixes d'une fonction (dimension 1)

Soit  $\Phi : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une fonction donnée. Rechercher un **point fixe** de  $\Phi$  revient à

Trouver  $\alpha \in [a, b]$  tel que

$$\alpha = \Phi(\alpha).$$

L'algorithme de la **méthode du point fixe** consiste en la construction, si elle existe, de la suite

$$x_{k+1} = \Phi(x_k), \quad \forall k \in \mathbb{N} \tag{2.1}$$

avec  $x_0 \in [a, b]$  donné.

Avant de formuler le théorème du point fixe, un petit rappel peut être utile:

**Définition 2.2.1.** On dit qu'une suite  $(x_k)_{k \in \mathbb{N}}$  obtenue par une méthode numérique, **converge vers  $\alpha$  avec un ordre  $p \geq 1$  au moins** si

$$\exists k_0 \in \mathbb{N}, \exists C > 0 \text{ tels que } |x_{k+1} - \alpha| \leq C|x_k - \alpha|^p, \forall k \geq k_0. \quad (2.2)$$

où  $C < 1$  si  $p = 1$ .

Voici une version du théorème du point fixe dans  $\mathbb{R}$ . Pour la résolution des systèmes non linéaires un théorème plus général sera proposé (Théorème 2.3.1): le théorème du point fixe dans un espace de Banach.

**Théorème 2.2.1** (Théorème du point fixe dans  $\mathbb{R}$ ). Soient  $[a, b]$  un intervalle non vide de  $\mathbb{R}$  et  $\Phi$  une application continue de  $[a, b]$  dans lui-même. Alors, il existe **au moins** un point  $\alpha \in [a, b]$  vérifiant  $\Phi(\alpha) = \alpha$ . Le point  $\alpha$  est appelé **point fixe de la fonction  $\Phi$** .

De plus, si  $\Phi$  est contractante (lipschitzienne de rapport  $L \in [0, 1[$ ), c'est à dire

$$\exists L < 1 \text{ t.q. } |\Phi(x) - \Phi(y)| \leq L|x - y| \forall (x, y) \in [a, b]^2, \quad (2.3)$$

alors  $\Phi$  admet un **unique** point fixe  $\alpha \in [a, b]$ .

Pour tout  $x_0 \in [a, b]$ , la suite

$$x_{k+1} = \Phi(x_k), \forall k \in \mathbb{N} \quad (2.4)$$

est bien définie et elle converge vers  $\alpha$  avec un ordre 1 au moins.

On a les deux estimations suivantes :

$$|x_k - \alpha| \leq L^k |x_0 - \alpha|, \forall k \geq 0, \quad (2.5)$$

$$|x_k - \alpha| \leq \frac{L}{1-L} |x_k - x_{k-1}|, \forall k \geq 0, \quad (2.6)$$

*Preuve. 1ère approche :*

- On montre tout d'abord l'existence du point fixe. Pour cela, on note  $f(x) = \Phi(x) - x$ .  $f$  est donc une application continue de  $[a, b]$  à valeurs réelles. On a  $f(a) = \Phi(a) - a \geq 0$  et  $f(b) = \Phi(b) - b \leq 0$  car  $a \leq \Phi(x) \leq b$ , pour tout  $x \in [a, b]$ . Si  $f(a) = 0$  ou  $f(b) = 0$ , alors l'existence est immédiate. Sinon (i.e.  $f(a) \neq 0$  et  $f(b) \neq 0$ ), on a  $f(a)f(b) < 0$  et par application directe du Théorème B.1.2 de Bolzano (ou TVI) on obtient l'existence.
- On montre ensuite l'unicité sous l'hypothèse de contraction (2.3). On suppose qu'il existe  $\alpha_1$  et  $\alpha_2$  dans  $[a, b]$  tels que  $\Phi(\alpha_1) = \alpha_1$  et  $\Phi(\alpha_2) = \alpha_2$ . Dans ce cas on a

$$|\alpha_1 - \alpha_2| = |\Phi(\alpha_1) - \Phi(\alpha_2)| \leq L|\alpha_1 - \alpha_2|.$$

On en déduit

$$(1 - L)|\alpha_1 - \alpha_2| \leq 0$$

Comme  $L < 1$  on a  $1 - L > 0$  et donc  $|\alpha_1 - \alpha_2| \leq 0$  ce qui entraîne  $\alpha_1 = \alpha_2$ .

- On a  $\Phi([a, b]) \subset [a, b]$  et comme  $x_0 \in [a, b]$ , la suite  $(x_k)_{k \in \mathbb{N}}$  est bien définie. On va démontrer la convergence de la suite  $x_k$  vers l'unique point fixe  $\alpha$  de  $\Phi$ . Pour cela, en utilisant la définition de la suite et du point fixe, on a

$$|x_{k+1} - \alpha| = |\Phi(x_k) - \Phi(\alpha)|$$

Comme  $\Phi$  est contractante,  $x_k \in [a, b]$  et  $\alpha \in [a, b]$  on obtient

$$|x_{k+1} - \alpha| \leq L|x_k - \alpha| \quad (2.7)$$

et une simple récurrence donne alors  $\forall k \in \mathbb{N}$

$$|x_k - \alpha| \leq L^k |x_0 - \alpha|$$

ce qui démontre la formule 2.5. En faisant tendre  $k$  vers  $+\infty$  on obtient la convergence de  $x_k$  vers  $\alpha$ .

De plus, d'après (2.7), comme  $0 < L < 1$ , la convergence est au moins d'ordre 1.

Pour la dernière estimation, on a :

$$\begin{aligned} |x_k - \alpha| &\leq L|x_{k-1} - \alpha| && \text{car } \Phi \text{ contractante} \\ &\leq L(|x_{k-1} - x_k| + |x_k - \alpha|) && \text{inégalité triangulaire} \end{aligned}$$

ce qui donne

$$(1 - L)|x_k - \alpha| \leq L|x_{k-1} - x_k|$$

Comme  $1 - L > 0$ , on en déduit immédiatement l'estimation (2.6).

### 2ème approche :

L'objectif de cette approche est de proposer une démonstration très proche de celle utilisée pour des espaces plus *complexes* (par exemple  $\mathbb{C}$ ,  $\mathbb{R}^n$ , ...). Dans la 1ère approche le théorème de Bolzano et des inégalités ont été utilisées pour démontrer l'existence d'un point fixe: ceci n'est plus possible dans un cadre plus générale.

On va donc supposer, dès le départ, que l'application  $\Phi$  est contractante de  $[a, b]$  dans lui-même. Ceci va permettre d'établir que la suite  $x_k$  est de Cauchy.

- Comme  $x_0 \in [a, b]$  et  $\Phi$  est une application continue de  $[a, b]$  dans lui-même, la suite  $x_k$  est bien définie  $\forall k \in \mathbb{N}$ .
- On démontre ensuite que  $x_k$  est une suite de Cauchy. Soit  $k > 0$ . On a

$$|x_{k+1} - x_k| = |\Phi(x_k) - \Phi(x_{k-1})| \leq L|x_k - x_{k-1}|,$$

et on obtient par récurrence

$$|x_{k+1} - x_k| \leq L^k|x_1 - x_0|$$

et

$$\forall l \geq 0, |x_{k+l} - x_{k+l-1}| \leq L^l|x_k - x_{k-1}|.$$

Soit  $p > 2$ . On en déduit par application répétée de l'inégalité triangulaire que

$$\begin{aligned} |x_{k+p} - x_k| &= |(x_{k+p} - x_{k+p-1}) + (x_{k+p-1} - x_{k+p-2}) + \dots + (x_{k+1} - x_k)| = \left| \sum_{l=0}^{p-1} (x_{k+l+1} - x_{k+l}) \right| \\ &\leq \sum_{l=0}^{p-1} |x_{k+l+1} - x_{k+l}| \\ &\leq \sum_{l=0}^{p-1} L^l|x_{k+1} - x_k| = \frac{1 - L^p}{1 - L}|x_{k+1} - x_k| \quad (\text{voir somme partielle d'une série géométrique}) \\ &\leq \frac{1 - L^p}{1 - L}L^k|x_1 - x_0|. \end{aligned}$$

Comme  $L^k \rightarrow 0$  quand  $k \rightarrow +\infty$ , on conclut que  $(x_k)$  est une suite de Cauchy. En effet, pour que  $(x_k)$  soit une suite de Cauchy, il faut montrer que

$$\forall \epsilon > 0, \exists M \in \mathbb{N}, \text{ tel que } \forall k \in \mathbb{N}, k \geq M, \forall p \in \mathbb{N}, |x_{k+p} - x_k| < \epsilon.$$

Comme  $0 < L < 1$ , on a

$$|x_{k+p} - x_k| \leq \frac{1}{1 - L}L^k|x_1 - x_0|$$

Soit  $\epsilon > 0$ , pour avoir  $|x_{k+p} - x_k| < \epsilon$ , il est suffisant d'avoir

$$\frac{1}{1 - L}L^k|x_1 - x_0| < \epsilon$$

c'est à dire, comme  $1 - L > 0$ ,

$$L^k < \frac{(1 - L)\epsilon}{|x_1 - x_0|}.$$

La fonction  $\ln$  étant croissante strictement on obtient

$$\ln(L^k) = k \ln(L) < \ln\left(\frac{(1-L)\epsilon}{|x_1 - x_0|}\right).$$

Or  $\ln(L) < 0$ , ce qui donne

$$k > \frac{1}{\ln(L)} \ln\left(\frac{(1-L)\epsilon}{|x_1 - x_0|}\right).$$

En prenant  $M \in \mathbb{N}$  tel que

$$M \geq \frac{1}{\ln(L)} \ln\left(\frac{(1-L)\epsilon}{|x_1 - x_0|}\right)$$

alors

$$\forall k \in \mathbb{N}, k \geq M, \forall p \in \mathbb{N}, |x_{k+p} - x_k| < \epsilon.$$

- La suite  $(x_k)$  est une suite de Cauchy dans  $\mathbb{R}$  espace complet donc elle converge vers  $\beta$  dans  $\mathbb{R}$ . De plus pour tout  $k$ ,  $x_k$  appartient à  $[a, b]$  fermé borné, donc sa limite  $\beta$  aussi.
- $\Phi$  étant contractante sur  $[a, b]$ , elle est donc continue. On a alors par continuité de  $\Phi$

$$\lim_{k \rightarrow +\infty} \Phi(x_k) = \Phi(\beta).$$

Comme  $x_{k+1} = \Phi(x_k)$  on aussi

$$\lim_{k \rightarrow +\infty} \Phi(x_k) = \lim_{k \rightarrow +\infty} x_{k+1} = \beta$$

et donc  $\beta$  est un point fixe de  $\Phi$ . L'existence d'un point fixe est donc établi.

La suite de la démonstration est inchangée. □

**Théorème 2.2.2** (Convergence globale de la méthode du point fixe). Soit  $\Phi \in \mathcal{C}^1([a, b])$  vérifiant  $\Phi([a, b]) \subset [a, b]$  et

$$\exists L < 1 \text{ tel que } \forall x \in [a, b], |\Phi'(x)| \leq L, \quad (2.8)$$

Soit  $x_0 \in [a, b]$  et  $(x_k)_{k \in \mathbb{N}}$  la suite définie par  $x_{k+1} = \Phi(x_k)$ . On a alors

- la fonction  $\Phi$  admet un unique point fixe  $\alpha \in [a, b]$ ,
- $\forall k \in \mathbb{N}, x_k \in [a, b]$ ,
- la suite  $(x_k)$  converge vers  $\alpha$  avec un ordre 1 au moins.
- Si  $x_0 \neq \alpha$ , alors

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{x_k - \alpha} = \Phi'(\alpha). \quad (2.9)$$

*Preuve.* Pour démontrer les trois premiers points il suffit de montrer que (2.8) entraîne que  $\Phi$  est contractante sur  $[a, b]$  pour pouvoir appliquer le théorème 2.2.1.

En effet, soit  $(x, y) \in [a, b]^2$ ,  $x \neq y$ . D'après le théorème B.1.3 des accroissements finis il existe  $\xi \in ]\min(x, y), \max(x, y)[$  tel que

$$\frac{\Phi(x) - \Phi(y)}{x - y} = \Phi'(\xi).$$

Ce résultat s'obtient aussi par un développement de Taylor. On obtient alors

$$|\Phi(x) - \Phi(y)| = |x - y| |\Phi'(\xi)| \leq L|x - y|.$$

L'application  $\Phi$  est donc contractante sur  $[a, b]$  et le théorème 2.2.1 s'applique.

Pour le dernier point, on utilise la définition de la suite et du point fixe  $\alpha$ :

$$x_{k+1} - \alpha = \Phi(x_k) - \Phi(\alpha).$$

On utilise le théorème B.1.3 des accroissements finis pour obtenir:  $\exists \xi_k \in ]\min(x_k, \alpha), \max(x_k, \alpha)[$  tel que

$$\frac{\Phi(x_k) - \Phi(\alpha)}{x_k - \alpha} = \Phi'(\xi_k).$$

Quand  $k \rightarrow +\infty$ , on a  $x_k \rightarrow \alpha$  et donc  $\xi_k \rightarrow \alpha$ . Par continuité de la fonction  $\Phi'$  on obtient (2.9).  $\square$

**Théorème 2.2.3** (Convergence locale de la méthode du point fixe). *Soit  $\alpha$  un point fixe d'une fonction  $\Phi$  de classe  $\mathcal{C}^1$  au voisinage de  $\alpha$ .*

*Si  $|\Phi'(\alpha)| < 1$ , alors il existe  $\delta > 0$  pour lequel  $x_k$  converge vers  $\alpha$  pour tout  $x_0$  tel que  $|x_0 - \alpha| \leq \delta$ .*

*De plus, si  $x_0 \neq \alpha$ , on a*

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{x_k - \alpha} = \Phi'(\alpha). \quad (2.10)$$

*si  $\Phi'(\alpha) \neq 0$ , la convergence est d'ordre 1 (exactement).*

*Preuve.* On va construire un intervalle fermé borné, noté  $\mathcal{V}$ , pour lequel les hypothèses du Théorème 2.2.2 sont vérifiées.

Comme  $\Phi'$  est continue dans un voisinage de  $\alpha$  avec  $|\Phi'(\alpha)| < 1$ , alors il existe  $\beta > 0$  tel que

$$\forall x \in ]\alpha - \beta, \alpha + \beta[, \quad |\Phi'(x)| < 1.$$

On propose ici une démonstration de ce résultat.

□ En notant  $I$  le voisinage de  $\alpha$  où  $\Phi$  est  $\mathcal{C}^1$ , et comme  $\Phi'$  est continue en  $\alpha$ , on a

$$\forall \varepsilon > 0, \exists \beta > 0 \text{ tel que } \forall x \in I, \left( |x - \alpha| < \beta \Rightarrow |\Phi'(x) - \Phi'(\alpha)| < \varepsilon \right).$$

On note  $M = \Phi'(\alpha)$  et on prend  $\varepsilon = 1 - |M|$  qui est strictement positif car  $0 \geq |M| = |\Phi'(\alpha)| < 1$ . Dans ce cas, il existe  $\beta > 0$  tel que  $] \alpha - \beta, \alpha + \beta [ \subset I$  et

$$\forall x \in I, \left( |x - \alpha| < \beta \Rightarrow |\Phi'(x) - M| < 1 - |M| \right).$$

Soit  $x \in I$  tel que  $|x - \alpha| < \beta$ , c'est à dire  $x \in ] \alpha - \beta, \alpha + \beta [$ , alors on a

$$\begin{aligned} |\Phi'(x) - M| < 1 - |M| &\Leftrightarrow -1 + |M| < \Phi'(x) - M < 1 - |M| \\ &\Leftrightarrow -1 + M + |M| < \Phi'(x) < 1 + M - |M| \end{aligned}$$

Comme  $-|M| \leq M \leq |M|$ , on a  $M + |M| \geq 0$  et  $M - |M| \leq 0$ , ce qui entraîne

$$|\Phi'(x) - M| < 1 - |M| \Rightarrow -1 < \Phi'(x) < 1.$$

□

En posant  $\delta = \beta/2$  (par ex.), on obtient, en définissant  $\mathcal{V} = [\alpha - \delta, \alpha + \delta]$ ,

$$\forall x \in \mathcal{V}, \quad |\Phi'(x)| < 1.$$

Comme l'intervalle  $\mathcal{V}$  est un fermé et que l'application  $|\Phi'|$  est continue sa borne supérieure est atteinte dans  $\mathcal{V}$  :

$$\exists \bar{x} \in \mathcal{V} \text{ tel que } |\Phi'(\bar{x})| = \sup_{x \in \mathcal{V}} |\Phi'(x)|.$$

On a donc  $L = |\Phi'(\bar{x})| < 1$ .

Montrons que  $\Phi(\mathcal{V}) \subset \mathcal{V}$ . En effet, d'après la formule de Taylor-Lagrange:

$\forall x \in \mathcal{V}, \exists \xi \in ]\min(x, \alpha), \max(x, \alpha)[ \subset \mathcal{V}$  tel que

$$\Phi(x) = \Phi(\alpha) + (x - \alpha)\Phi'(\xi).$$

On en déduit

$$|\Phi(x) - \Phi(\alpha)| = |x - \alpha| |\Phi'(\xi)| \leq \delta |\Phi'(\xi)| \leq \delta$$

Comme  $\Phi(\alpha) = \alpha$ , on obtient  $|\Phi(x) - \alpha| \leq \delta$  i.e.  $\Phi(x) \in \mathcal{V}$ .

On peut donc appliquer le Théorème 2.2.2 avec  $[a, b] = \mathcal{V} = [\alpha - \delta, \alpha + \delta]$  ce qui permet de conclure.  $\square$

**EXERCICE 2.2.1**

Soit  $\alpha$  un point fixe d'une fonction  $\Phi$  de classe  $\mathcal{C}^1$  au voisinage de  $\alpha$  et vérifiant  $\Phi'(\alpha) = 0$ .

**Q. 1**

Montrer qu'il existe  $\delta > 0$  tel que  $\forall x_0 \in ]\alpha - \delta, \alpha + \delta[$  la suite définie par  $x_{k+1} = \Phi(x_k)$  converge vers  $\alpha$ .

On suppose de plus que  $\Phi'$  est dérivable sur  $] \alpha - \delta, \alpha + \delta[$  et qu'il existe  $M \in \mathbb{R}^+$  tel que

$$\forall x \in [\alpha - \delta, \alpha + \delta], |\Phi''(x)| \leq M$$

**Q. 2**

a. Montrer que

$$\forall x_0 \in [\alpha - \delta, \alpha + \delta], |x_k - \alpha| \leq \frac{2}{M} \left( \frac{1}{2} M |x_0 - \alpha| \right)^{2^k}$$

b. Quel est l'ordre de convergence dans ce cas.

**Q. 3**

A quelle condition a-t'on

$$|x_k - \alpha| \leq \frac{2}{M} 10^{-2^k}.$$

**Correction 2.2.1****R. 1****R. 2**

a. D'après la formule de Taylor-Lagrange rappelée au Théorème B.1.1, on a  $\exists \eta \in ]\min(\alpha, x), \max(\alpha, x)[$  tel que

$$\begin{aligned} \Phi(x) &= \Phi(\alpha) + (x - \alpha)\Phi'(\alpha) + \frac{(x - \alpha)^2}{2!}\Phi''(\eta) \\ &= \alpha + \frac{1}{2}\Phi''(\eta)(x - \alpha)^2. \end{aligned}$$

On en déduit que  $|\Phi(x) - \alpha| \leq \frac{M}{2}|x - \alpha|^2$  ce qui s'écrit encore  $\frac{M}{2}|\Phi(x) - \alpha| \leq \left(\frac{M}{2}|x - \alpha|\right)^2$ . Or si  $x_0 \in [\alpha - \delta, \alpha + \delta]$ , alors  $x_{k-1} \in [\alpha - \delta, \alpha + \delta]$ ,  $\forall k \in \mathbb{N}^*$ . On a alors

$$\frac{M}{2}|\Phi(x_{k-1}) - \alpha| = \frac{M}{2}|x_k - \alpha| \leq \left(\frac{M}{2}|x_{k-1} - \alpha|\right)^2$$

et donc par récurrence

$$\frac{M}{2}|x_k - \alpha| \leq \left(\frac{M}{2}|x_0 - \alpha|\right)^{2^k}.$$

b. On a

$$|x_k - \alpha| \leq \frac{2}{M} \left(\frac{M}{2}|x_{k-1} - \alpha|\right)^2$$

c'est à dire

$$\frac{|x_k - \alpha|}{|x_{k-1} - \alpha|^2} \leq \frac{M}{2}$$

et donc la méthode est d'ordre 2.

**R. 3**

En supposant de plus que  $|x_0 - \alpha| \leq \frac{1}{5M}$ , on obtient immédiatement le résultat.

~~~~~ Fin correction ~~~~~

On va maintenant généraliser le résultat de cet exercice:

**Proposition 2.2.1.** Soit  $p \in \mathbb{N}^*$ , et  $\Phi \in \mathcal{C}^{p+1}(\mathcal{V})$  pour un certain voisinage  $\mathcal{V}$  de  $\alpha$  point fixe de  $\Phi$ . Si  $\Phi^{(i)}(\alpha) = 0$ , pour  $1 \leq i \leq p$ , alors la méthode de point fixe associée à la fonction  $\Phi$  est d'ordre  $(p+1)$  au moins et

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^{p+1}} = \frac{\Phi^{(p+1)}(\alpha)}{(p+1)!}. \quad (2.11)$$

Elle est d'ordre  $(p+1)$  (exactement) si  $\Phi^{(p+1)}(\alpha) \neq 0$ .

*Preuve.* Les hypothèses du théorème 2.2.3 étant vérifiées ( $\Phi'(\alpha) = 0$ ), la suite  $(x_k)$  converge vers  $\alpha$ . D'après un développement de Taylor-Lagrange (voir le Théorème B.1.1) de  $\Phi$  en  $\alpha$ , il existe  $\eta_k$  entre  $x_k$  et  $\alpha$  vérifiant

$$\Phi(x_k) = \sum_{i=0}^p \frac{(x_k - \alpha)^i}{i!} \Phi^{(i)}(\alpha) + \frac{(x_k - \alpha)^{p+1}}{(p+1)!} \Phi^{(p+1)}(\eta_k).$$

Comme  $\alpha = \Phi(\alpha)$  et  $\Phi^{(i)}(\alpha) = 0$ , pour  $1 \leq i \leq p$  on obtient

$$\begin{aligned} x_{k+1} &= \Phi(x_k) + \sum_{i=1}^p \frac{(x_k - \alpha)^i}{i!} \Phi^{(i)}(\alpha) + \frac{(x_k - \alpha)^{p+1}}{(p+1)!} \Phi^{(p+1)}(\eta_k) \\ &= \alpha + \frac{(x_k - \alpha)^{p+1}}{(p+1)!} \Phi^{(p+1)}(\eta_k) \end{aligned}$$

De plus  $(\eta_k)$  converge vers  $\alpha$  car  $\eta_k$  est entre  $x_k$  et  $\alpha$  et donc comme  $\Phi^{(p+1)}$  est continue au voisinage de  $\alpha$  on obtient

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^{p+1}} = \lim_{k \rightarrow +\infty} \frac{\Phi^{(p+1)}(\eta_k)}{(p+1)!} = \frac{\Phi^{(p+1)}(\alpha)}{(p+1)!}.$$

□

### 2.2.1 Points fixes attractifs et répulsifs

Soit  $\Phi : [a, b] \rightarrow [a, b]$  une application de classe  $\mathcal{C}^1$  admettant un point fixe  $\alpha \in [a, b]$ .

#### Points fixes attractifs

On suppose  $|\Phi'(\alpha)| < 1$ . D'après le théorème 2.2.3, il existe  $\delta > 0$  tel que

$$\forall x_0 \in [\alpha - \delta, \alpha + \delta], \quad \lim_{k \rightarrow +\infty} x_k = \alpha.$$

Dans ce cas on dit que  $\alpha$  est un **point fixe attractif** pour  $\Phi$ .

#### Points fixes répulsifs

Cas  $|\Phi'(\alpha)| > 1$ . Par définition de la dérivée en  $\alpha$  on a

$$\lim_{\substack{x \rightarrow \alpha \\ x \neq \alpha}} \left| \frac{\Phi(x) - \Phi(\alpha)}{x - \alpha} \right| = |\Phi'(\alpha)| > 1.$$

Il existe alors  $\delta > 0$  tel que

$$\forall x \in [\alpha - \delta, \alpha + \delta] \setminus \{\alpha\}, \quad |\Phi(x) - \alpha| > |x - \alpha|$$

et donc la suite  $(x_k)$  ne peut converger vers  $\alpha$  et ceci même si  $x_0$  est très proche de  $\alpha$ . Dans ce cas on dit que  $\alpha$  est un **point fixe répulsif** pour  $\Phi$ .

Toutefois, on *peut rattrapper le coup*. En effet, comme  $\Phi'$  est non nulle et de signe constant au voisinage de  $\alpha$  : il existe  $h > 0$  tel que la fonction  $\Phi$  soit strictement monotone sur  $I = [\alpha - h, \alpha + h]$ . D'après le corollaire B.1.1  $\Phi$  est une bijection de  $I$  dans l'intervalle fermé  $J = \Phi^{-1}(I)$ . Comme  $\alpha = \Phi(\alpha)$ , on a  $\alpha \in J$  et  $\Phi^{-1}(\alpha) = \alpha$ . Le problème de point fixe trouver  $x \in I$  tel que  $\Phi(x) = x$  revient alors à trouver  $x \in J$  tel que  $\Phi^{-1}(x) = x$ . Or  $\Phi'(\alpha) \neq 0$  et  $\Phi(\alpha) = \alpha$ , la proposition B.1.3 donne alors  $(\Phi^{-1})'(\alpha) = 1/\Phi'(\alpha)$  et donc  $|(\Phi^{-1})'(\alpha)| < 1$ . Le point  $\alpha$  est un **point fixe attractif** pour  $\Phi^{-1}$ .

## Points fixes indéterminés

Dans le cas  $|\Phi'(\alpha)| = 1$ , on ne peut conclure dans le cas général.

### 2.2.2 Interprétations graphiques de la méthode du point fixe

**Exemple 1 : point fixe de la fonction  $x \mapsto x^2$ .**

On s'intéresse ici au point fixe  $\alpha = 1$  de la fonction  $\Phi : x \mapsto x^2$ .

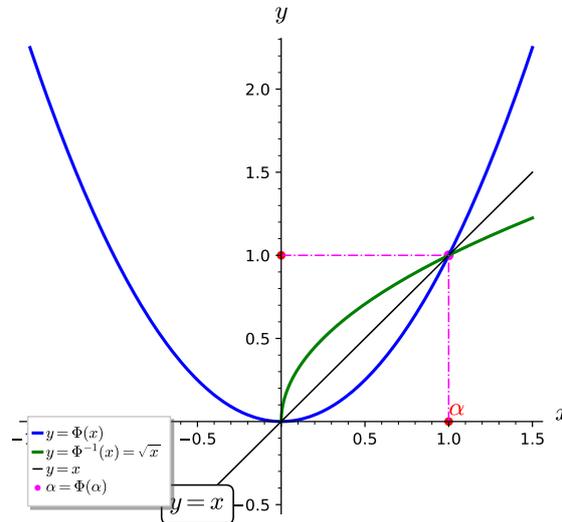


Figure 2.3: fonction  $x^2$  et son fonction inverse  $\sqrt{x}$  sur  $[0, +\infty[$

Comme  $\Phi'(\alpha) = 2$  le point fixe  $\alpha$  est répulsif. On donne dans le tableau suivant les premiers termes de deux suites : l'une avec  $x_0 = 1.05$  et l'autre avec  $x_0 = 0.95$ . Dans ce dernier cas, la suite va converger ... vers un autre point fixe.

| $k$      | $x_k$    | $x_k$                    |
|----------|----------|--------------------------|
| 0        | 1.05000  | 0.950000                 |
| 1        | 1.10250  | 0.902500                 |
| 2        | 1.21551  | 0.814506                 |
| 3        | 1.47746  | 0.663420                 |
| $\vdots$ | $\vdots$ | $\vdots$                 |
| 8        | 265742.  | $1.98264 \times 10^{-6}$ |

Les premières itérations de ces deux suites sont représentées en Figure 2.4.

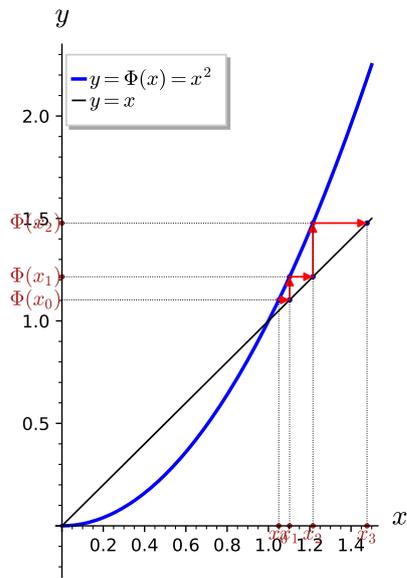
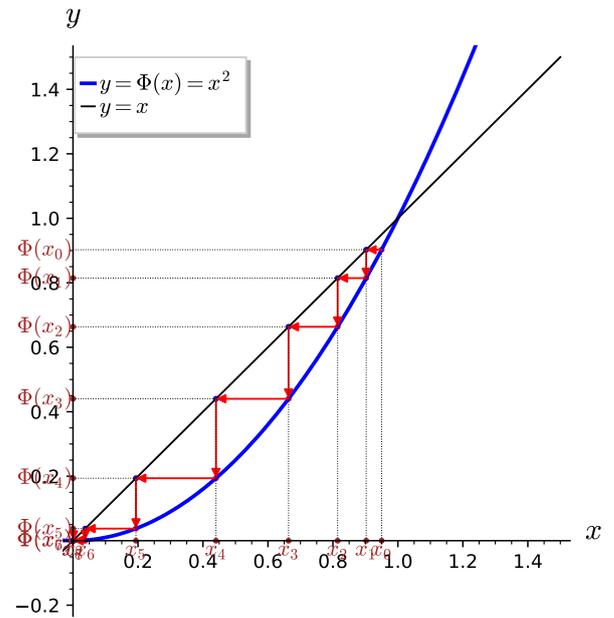
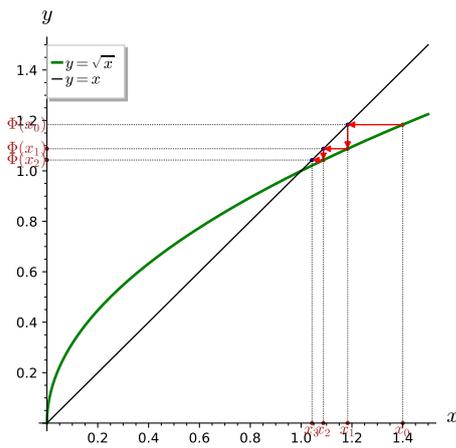
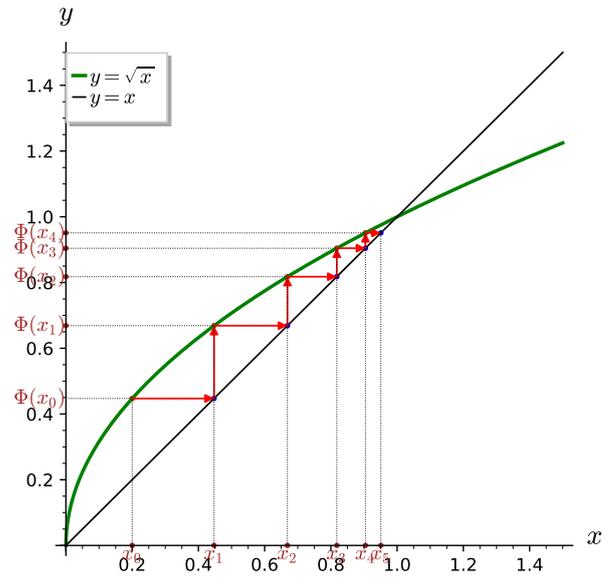
Sur l'intervalle  $[0, +\infty[$ , la fonction  $\Phi$  admet comme fonction inverse  $\Phi^{-1}(x) = \sqrt{x}$  et on a aussi  $\Phi^{-1}(1) = 1$  (i.e.  $\alpha = 1$  est un point fixe de  $\Phi^{-1}$ ). De plus  $(\Phi^{-1})'(1) = 1/2 < 1$ , ce qui permet d'affirmer que  $\alpha = 1$  est un point fixe **attractif** de  $\Phi^{-1}$ ). On donne dans le tableau suivant les premiers termes de deux suites : l'une avec  $x_0 = 1.40$  et l'autre avec  $x_0 = 0.200$ .

| $k$      | $x_k$    | $x_k$    |
|----------|----------|----------|
| 0        | 1.40000  | 0.200000 |
| 1        | 1.18322  | 0.447214 |
| 2        | 1.08776  | 0.668740 |
| 3        | 1.04296  | 0.817765 |
| $\vdots$ | $\vdots$ | $\vdots$ |
| 8        | 1.00132  | 0.993733 |

Les premières itérations de ces deux suites sont représentées en Figure 2.5.

**Exemple 2 : point fixe de la fonction  $x \mapsto x^2 - x + 1$ .**

On s'intéresse ici au point fixe  $\alpha = 1$  de la fonction  $\Phi : x \mapsto x^2 - x + 1$  représenté en Figure 2.6. On note que  $\Phi'(\alpha) = 1$ .

(a)  $x_0 = 1.05$ (b)  $x_0 = 0.950$ Figure 2.4:  $\alpha = 1$ , point fixe répulsif de  $x \mapsto x^2$ (a)  $x_0 = 1.40$ (b)  $x_0 = 0.200$ Figure 2.5:  $\alpha = 1$ , point fixe attractif de  $x \mapsto \sqrt{x}$

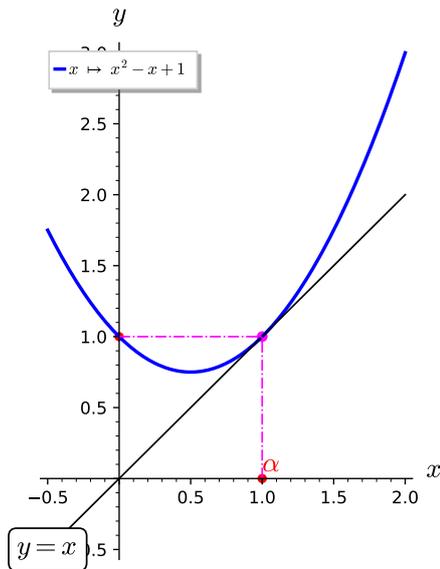
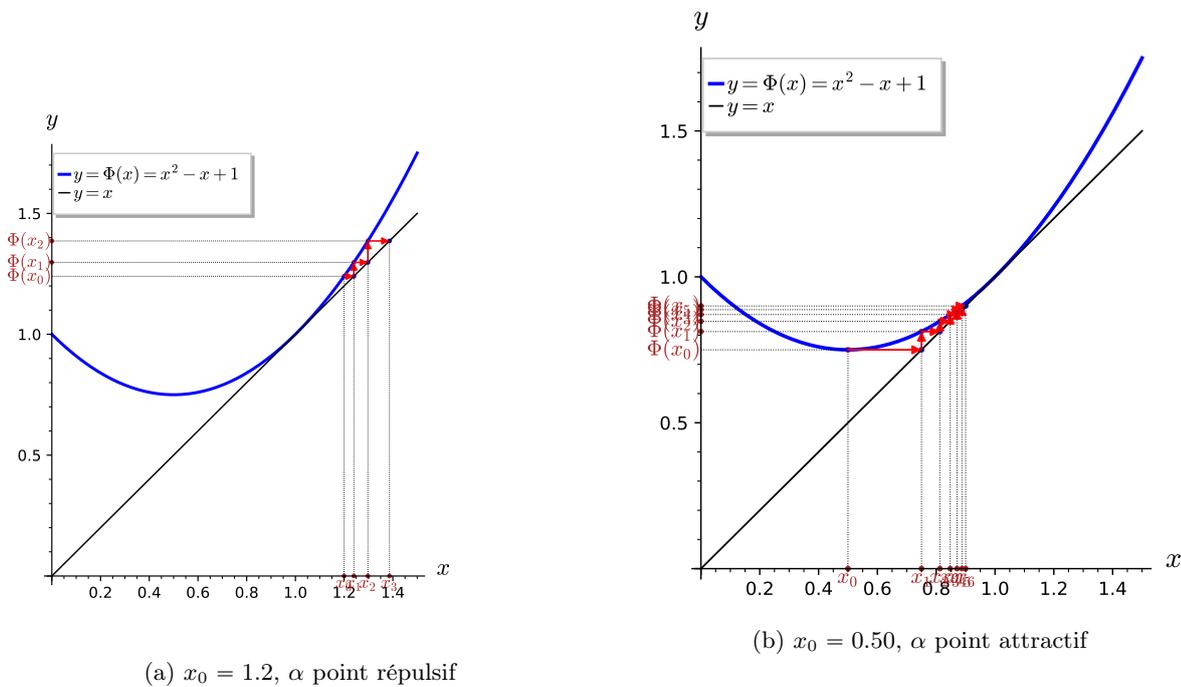


Figure 2.6: fonction  $x^2 - x + 1$  et son point fixe  $\alpha = 1$ .

On donne dans le tableau suivant les premiers termes de deux suites : l'une avec  $x_0 = 1.20$  diverge et l'autre avec  $x_0 = 0.500$  converge vers  $\alpha$ .

| diverge                            | converge              |
|------------------------------------|-----------------------|
| $x_0 = 1.20000$                    | $x_0 = 0.500000$      |
| $x_1 = 1.24000$                    | $x_1 = 0.750000$      |
| $x_2 = 1.29760$                    | $x_2 = 0.812500$      |
| $x_3 = 1.38617$                    | $x_3 = 0.847656$      |
| $\vdots$                           | $\vdots$              |
| $x_{15} = 2.32096 \times 10^{161}$ | $x_{5000} = 0.999800$ |

Les premières itérations de ces deux suites sont représentées en Figure 2.7.



(a)  $x_0 = 1.2$ ,  $\alpha$  point répulsif

(b)  $x_0 = 0.50$ ,  $\alpha$  point attractif

Figure 2.7:  $\alpha = 1$ , point fixe attractif ou répulsif de  $x \mapsto x^2 - x + 1$

**Exemple 3 : point fixe de fonctions affines particulières**

On va étudier les points fixes des trois fonctions affines vérifiant

$$\Phi_1(1) = \Phi_2(1) = \Phi_3(1) = 1 \quad \text{et} \quad \Phi_1'(1) = -1, \quad \Phi_2'(1) = -3/4, \quad \Phi_3'(1) = -4/3.$$

Ces trois fonctions sont données par

$$\Phi_1(x) = -x + 2, \quad \Phi_2(x) = -\frac{3}{4}x + \frac{7}{4} \quad \text{et} \quad \Phi_3(x) = -\frac{4}{3}x + \frac{7}{3}.$$

On donne dans le tableau suivant les premiers termes de trois suites obtenues avec  $x_0 = 1.5$  pour les suites associées à  $\Phi_1$  et  $\Phi_2$ , et avec  $x_0 = 1.1$  pour celle associée à  $\Phi_3$ .

| $k$      | $x_k (\Phi_1)$ | $x_k (\Phi_2)$ | $x_k (\Phi_3)$ |
|----------|----------------|----------------|----------------|
| 0        | 1.50000        | 1.50000        | 1.10000        |
| 1        | 0.50000        | 0.625000       | 0.866667       |
| 2        | 1.50000        | 1.28125        | 1.17778        |
| 3        | 0.50000        | 0.789062       | 0.762963       |
| $\vdots$ | $\vdots$       | $\vdots$       | $\vdots$       |
| 19       | 0.500000       | 0.997886       | -22.6503       |
| 20       | 1.50000        | 1.00159        | 32.5337        |

Les premières itérations des trois suites obtenues sont représentées en Figure 2.8.

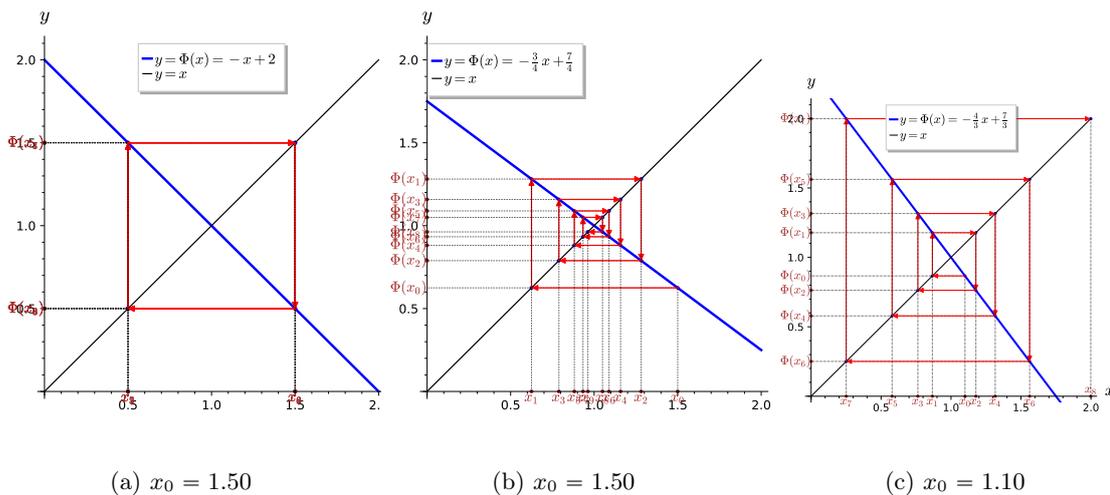


Figure 2.8:  $\alpha = 1$ , point fixe de fonctions affines particulières

### 2.2.3 Algorithme générique du point fixe

Le principe de base est très simple et donné par les Algorithmes 2.6 et 2.7, écrit respectivement avec une boucle **Tantque** et une boucle **Répéter**.

**Algorithm 2.6** Méthode de point fixe : version **Tantque formel**

- 1:  $k \leftarrow 0$
- 2: **Tantque** non convergence **faire**
- 3:  $x_{k+1} \leftarrow \Phi(x_k)$
- 4:  $k \leftarrow k + 1$
- 5: **Fin**
- 6:  $\alpha_{\text{tol}} \leftarrow x_k$  ▷ le dernier calculé.

**Algorithm 2.7** Méthode de point fixe : version **Répéter formel**

- 1:  $k \leftarrow 0$
- 2: **Répéter**
- 3:  $k \leftarrow k + 1$
- 4:  $x_k \leftarrow \Phi(x_{k-1})$
- 5: **jusqu'à** convergence
- 6:  $\alpha_{\text{tol}} \leftarrow x_k$  ▷ le dernier calculé.

Nous allons plus particulièrement étudier les critères d'arrêt des boucles **Tantque** et **Répéter** (négation l'un de l'autre) qui sont bien trop flous : on s'arrête quand on converge!

Tout d'abord, on n'est pas sur de converger : il faudra donc n'autoriser qu'un nombre maximum d'itérations  $k_{\max}$ . De plus, si l'on converge vers une certaine valeur  $\alpha$  celle-ci n'étant pas connue à l'avance (sinon l'algo n'a aucun intérêt), on ne peut utiliser, comme condition du **Tantque**,  $|x_k - \alpha| > \epsilon$  ( $|x_k - \alpha| \leq \epsilon$  pour la condition du **Répéter**) où  $\epsilon$  serait la précision souhaitée. L'idée serait de comparer  $x_{k+1}$  et  $x_k$  et donc de tester s'ils sont suffisamment proches : la condition serait alors  $|\Phi(x_k) - x_k| = |x_{k+1} - x_k| > \text{tol}$ , (boucle **Tantque**) ou  $|\Phi(x_k) - x_k| \leq \text{tol}$ , (boucle **Répéter**) avec  $\text{tol}$  la tolérance souhaitée. Il faut noter que dans ce cas la valeur  $\text{tol}$  doit être choisie correctement et *dépend* de l'ordre de grandeur de  $\alpha$ . Si  $\alpha$  est de l'ordre de  $10^8$ ,  $\text{tol} = 1$  comme valeur est raisonnable. Toutefois on peut lui préférer une condition *relative*  $\frac{|\Phi(x_k) - x_k|}{|x_k|+1} > \text{tol}$  (boucle **Tantque**) ou  $\frac{|\Phi(x_k) - x_k|}{|x_k|+1} \leq \text{tol}$  (boucle **Répéter**) qui permet à  $\text{tol}$  de correspondre à une tolérance *relative* souhaitée que ce soit pour de grandes valeurs de  $\alpha$  ou pour des petites. Les algorithmes du point fixe (recherche de  $\alpha$  solution de  $\Phi(x) = x$ ) avec notations mathématiques sont donnés par Algorithme 2.8 et 2.9, respectivement avec des boucles **Tantque** et **Répéter**.

**Algorithm 2.8** Méthode de point fixe : version **Tantque** *formel* avec critères d'arrêt

```

1:  $k \leftarrow 0$ 
2:  $\text{err} \leftarrow |\Phi(x_0) - x_0|$  ▷ ou  $\frac{|\Phi(x_0) - x_0|}{|x_0|+1}$ 
3: Tantque  $\text{err} > \epsilon$  et  $k \leq k_{\max}$  faire
4:    $k \leftarrow k + 1$ 
5:    $x_k \leftarrow \Phi(x_{k-1})$ 
6:    $\text{err} \leftarrow |\Phi(x_k) - x_k|$  ▷ ou  $\frac{|\Phi(x_k) - x_k|}{|x_k|+1}$ 
7: Fin
8: Si  $\text{err} \leq \text{tol}$  alors ▷ Convergence
9:    $\alpha_{\text{tol}} \leftarrow x_k$  ▷  $|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}| \leq \text{tol}$ 
10: Fin

```

**Algorithm 2.9** Méthode de point fixe : version **Répéter** *formel* avec critères d'arrêt

```

1:  $k \leftarrow 0$ 
2: Répéter
3:    $\text{err} \leftarrow |\Phi(x_k) - x_k|$  ▷ ou  $\frac{|\Phi(x_k) - x_k|}{|x_k|+1}$ 
4:    $x_{k+1} \leftarrow \Phi(x_k)$ 
5:    $k \leftarrow k + 1$ 
6: jusqu'à  $\text{err} \leq \text{tol}$  ou  $k > k_{\max}$ 
7: Si  $\text{err} \leq \text{tol}$  alors ▷ Convergence
8:    $\alpha_{\text{tol}} \leftarrow x_k$  ▷  $|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}| \leq \text{tol}$ 
9: Fin

```

Des versions, sous forme de fonction, plus proches de la programmation sont proposées en Algorithme 2.10 et 2.11. Bien évidemment, suivant l'usage que l'on souhaite de ces fonctions, il est facile de les adapter pour retourner plus d'informations (nombre d'itération effectives, statut de convergence, ...)

**Algorithm 2.10** Méthode de point fixe : version **Tantque** avec critères d'arrêt

**Données :**

$\Phi$  :  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $\text{tol}$  : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 $k_{\max}$  : nombre maximum d'itérations,  $k_{\max} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}| \leq \text{tol}$   
 (ou  $\frac{|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}|}{|\alpha_{\text{tol}}|+1} \leq \text{tol}$ )

```

1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{PtFixe}(\Phi, x_0, \text{tol}, k_{\max})$ 
2:    $k \leftarrow 0$ ,  $\alpha_{\text{tol}} \leftarrow \emptyset$ 
3:    $x \leftarrow x_0$ ,  $\text{fx} \leftarrow \Phi(x_0)$ ,
4:    $\text{err} \leftarrow |\text{fx} - x|$  ▷ ou  $\frac{|\text{fx} - x|}{|x|+1}$ 
5:   Tantque  $\text{err} > \text{tol}$  et  $k \leq k_{\max}$  faire
6:      $k \leftarrow k + 1$ 
7:      $x \leftarrow \text{fx}$ 
8:      $\text{fx} \leftarrow \Phi(x)$ 
9:      $\text{err} \leftarrow |\text{fx} - x|$  ▷ ou  $\frac{|\text{fx} - x|}{|x|+1}$ 
10:  Fin
11:  Si  $\text{err} \leq \text{tol}$  alors ▷ Convergence
12:     $\alpha_{\text{tol}} \leftarrow x$ 
13:  Fin
14: Fin

```

**Algorithm 2.11** Méthode de point fixe : version **Répéter** avec critères d'arrêt

**Données :**

$\Phi$  :  $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $\text{tol}$  : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 $k_{\max}$  : nombre maximum d'itérations,  $k_{\max} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}| \leq \text{tol}$   
 (ou  $\frac{|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}|}{|\alpha_{\text{tol}}|+1} \leq \text{tol}$ )

```

1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{PtFixe}(\Phi, x_0, \text{tol}, k_{\max})$ 
2:    $k \leftarrow 0$ ,  $\alpha_{\text{tol}} \leftarrow \emptyset$ 
3:    $x \leftarrow x_0$ 
4:   Répéter
5:      $x_p \leftarrow x$ 
6:      $x \leftarrow \Phi(x_p)$ 
7:      $\text{err} \leftarrow |x - x_p|$  ▷ ou  $\frac{|x - x_p|}{|x_p|+1}$ 
8:      $k \leftarrow k + 1$ 
9:   jusqu'à  $\text{err} \leq \text{tol}$  ou  $k > k_{\max}$ 
10:  Si  $\text{err} \leq \text{tol}$  alors ▷ Convergence
11:     $\alpha_{\text{tol}} \leftarrow x$ 
12:  Fin
13: Fin

```

## 2.2.4 Méthodes de points fixes pour la recherche de racines

On peut noter que résoudre  $f(x) = 0$  revient, par exemple, à résoudre  $\Phi(x) := x + f(x) = x$ . De manière plus générale, si  $\mathcal{F}$  est une fonction continue vérifiant  $\mathcal{F}(0) = 0$  alors  $\Phi(x) = x + \mathcal{F}(f(x))$  est un autre choix possible.

Soit  $f$  une fonction de classe  $\mathcal{C}^1$  dans un voisinage d'une de ses racines simple  $\alpha$ .

**Selectionner une version :**

**Versión 1** En appliquant la formule de Taylor pour tout  $x$  dans ce voisinage, il existe  $\xi \in ]\min(x, \alpha), \max(x, \alpha)[$  tel que

$$f(\alpha) = 0 = f(x) + (\alpha - x)f'(\xi).$$

Ceci conduit à la méthode itérative suivante :  $x_0$  étant donné dans le voisinage de  $\alpha$ , on doit,  $\forall k \in \mathbb{N}$ , déterminer  $x_{k+1}$  vérifiant  $f(x_k) + (x_{k+1} - x_k)q_k = 0$  sachant que  $q_k$  est une approximation de  $f'(\xi)$ .

**Versión 2** On cherche à déterminer une méthode itérative permettant de calculer  $x_{k+1}$  en fonction des valeurs précédentes en espérant que  $|x_{k+1} - \alpha| \leq |x_k - \alpha|$ . Pour cela, on écrit une formule de Taylor en  $x_k$  supposé proche de  $\alpha$  et on note  $h = \alpha - x_k$

$$f(\alpha) = 0 = f(x_k) + hf'(x_k) + o(h)$$

L'idéal serait de trouver la valeur exacte de  $h$  et de prendre  $x_{k+1} = x_k + h$  mais ceci n'est pas possible dans un cadre général. C'est pourquoi on cherche une approximation  $\tilde{h}$  de  $h$ . De plus, on ne connaît pas forcément explicitement la dérivée de  $f$ . On note donc  $q_k$  une approximation de  $f'(x_k)$ . On choisit alors  $\tilde{h}$  comme solution de

$$f(x_k) + \tilde{h}q_k = 0$$

Si  $q_k \neq 0$ , on obtient la suite itérative

$$x_{k+1} = x_k - \frac{f(x_k)}{q_k}, \quad \forall k \in \mathbb{N} \quad (2.12)$$

La valeur  $x_{k+1}$  est de fait l'intersection de la droite passant par le point  $((x_k), f(x_k))$  et de pente  $q_k$  avec l'axe des  $x$ .

Par la suite, on va écrire un algorithme du point fixe et étudier différentes méthodes :

- la **méthode de la corde** :

$$q_k = q = \frac{f(b) - f(a)}{b - a}$$

- la **méthode de la sécante** :

$$q_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

où  $x_{-1}$  et  $x_0$  sont données,

- la **méthode de Newton** : en supposant  $f'$  connu, on prend

$$q_k = f'(x_k).$$

### La méthode de la corde

Soit  $f$  une fonction de classe  $\mathcal{C}^1$  sur  $[a, b]$  tel que  $f(a) \neq f(b)$ . Soit  $x_0 \in [a, b]$  donné. La suite obtenue par la méthode de la corde est donnée par

$$x_{k+1} = x_k - \frac{b - a}{f(b) - f(a)} f(x_k), \quad \forall k \in \mathbb{N}. \quad (2.13)$$

On peut voir cette relation comme un cas particulier de la méthode du point fixe. En effet, en prenant  $\Phi(x) = x - \frac{b-a}{f(b)-f(a)} f(x)$ , la suite définie en (2.13) s'écrit  $x_{k+1} = \Phi(x_k)$ .

On a alors le résultat suivant

**Proposition 2.2.2** (convergence de la méthode de la corde). Soit  $f \in \mathcal{C}^1([a, b])$  tel que  $f(b) \neq f(a)$  et  $\lambda = \frac{f(b)-f(a)}{b-a}$ . On suppose de plus que  $\forall x \in [a, b]$

$$\min(\lambda(x-a), \lambda(x-b)) \leq f(x) \leq \max(\lambda(x-a), \lambda(x-b)) \quad (2.14)$$

$$\min(0, 2\lambda) < f'(x) < \max(0, 2\lambda) \quad (2.15)$$

On note  $(x_k)_{k \in \mathbb{N}}$  la suite donnée par  $x_0 \in [a, b]$  et pour tout  $k \geq 0$

$$x_{k+1} = x_k - \frac{f(x_k)}{\lambda}. \quad (2.16)$$

alors la suite  $(x_k)$  est bien définie et converge vers l'unique racine  $\alpha \in [a, b]$  de  $f$ .

Preuve. voir correction Exercice 2.2.2 □

### EXERCICE 2.2.2

Soit  $f$  une fonction de classe  $\mathcal{C}^1$  sur  $[a, b]$  vérifiant  $f(a)f(b) < 0$ . et  $\lambda = \frac{f(b)-f(a)}{b-a}$ . Soit  $x_0 \in [a, b]$  donné. La suite obtenue par la méthode de la corde est donnée par

$$x_{k+1} = x_k - \frac{f(x_k)}{\lambda}, \quad \forall k \in \mathbb{N}.$$

On note  $\Phi(x) = x - \frac{f(x)}{\lambda}$ .

**Q. 1** Montrer que si pour tout  $x \in [a, b]$  on a

$$\min(\lambda(x-a), \lambda(x-b)) \leq f(x) \leq \max(\lambda(x-a), \lambda(x-b)) \quad (2.1)$$

alors  $\Phi([a, b]) \subset [a, b]$ .

**Q. 2** Montrer que si pour tout  $x \in [a, b]$  on a

$$\min(0, 2\lambda) < f'(x) < \max(0, 2\lambda) \quad (2.2)$$

alors  $|\Phi'(x)| < 1$ .

**Q. 3** En déduire que sous les deux conditions précédentes la méthode de la corde converge vers l'unique solution  $\alpha \in [a, b]$  de  $f(x) = 0$ .

### Correction 2.2.2

**R. 1**

Si  $\lambda > 0$ , l'inéquation (2.1) devient

$$\begin{aligned} \lambda(x-b) \leq f(x) \leq \lambda(x-a) &\Leftrightarrow a \leq x - \frac{f(x)}{\lambda} \leq b \\ &\Leftrightarrow a \leq \Phi(x) \leq b. \end{aligned}$$

Si  $\lambda < 0$ , l'inéquation (2.1) devient

$$\begin{aligned} \lambda(x-a) \leq f(x) \leq \lambda(x-b) &\Leftrightarrow a \leq x - \frac{f(x)}{\lambda} \leq b \\ &\Leftrightarrow a \leq \Phi(x) \leq b. \end{aligned}$$

R. 2

Si  $\lambda > 0$ , l'inéquation (2.2) devient

$$\begin{aligned} 0 < f'(x) < 2\lambda &\Leftrightarrow 0 < \frac{f'(x)}{\lambda} < 2 \\ &\Leftrightarrow -1 < 1 - \frac{f'(x)}{\lambda} < 1 \\ &\Leftrightarrow -1 < \Phi'(x) < 1. \end{aligned}$$

Si  $\lambda < 0$ , l'inéquation (2.2) devient

$$\begin{aligned} 2\lambda < f'(x) < 0 &\Leftrightarrow 0 < \frac{f'(x)}{\lambda} < 2 \\ &\Leftrightarrow -1 < 1 - \frac{f'(x)}{\lambda} < 1 \\ &\Leftrightarrow -1 < \Phi'(x) < 1. \end{aligned}$$

R. 3

Sous les hypothèses (2.1) et (2.2) on a  $\Phi([a, b]) \subset [a, b]$  et  $\forall x \in [a, b]$ ,  $|\Phi'(x)| < 1$ . Comme  $f$  est de classe  $\mathcal{C}^1$  sur  $[a, b]$ , la fonction  $\Phi$  l'est aussi. La suite  $(x_k)$  est définie par  $x_{k+1} = \Phi(x_k)$ . Ainsi les hypothèses du théorème 2.2.2 sont vérifiées ce qui assure l'unicité du point fixe ainsi que la convergence de la suite  $(x_k)$  vers ce point fixe.

~~~~~ Fin correction ~~~~~

**Proposition 2.2.3** (ordre de convergence de la méthode de la corde). Soit  $f \in \mathcal{C}^1([a, b])$  tel que  $f(b) \neq f(a)$ . Soit  $(x_k)_{k \in \mathbb{N}}$  la suite définie par la méthode de la corde

$$x_{k+1} = x_k - \frac{b-a}{f(b)-f(a)} f(x_k), \quad \forall k \in \mathbb{N} \text{ avec } x_0 \in [a, b]$$

Si cette suite converge vers  $\alpha \in ]a, b[$  alors la convergence est au moins d'ordre 1.

De plus, si  $f$  est de classe  $\mathcal{C}^2$  sur un certain voisinage  $\mathcal{V}$  de  $\alpha$  et si  $f'(\alpha) = \frac{f(b)-f(a)}{b-a}$  alors la convergence est au moins d'ordre 2.

*Preuve.* • **Order 1** : On note  $\lambda = \frac{f(b)-f(a)}{b-a}$ . On a par définition  $x_{k+1} = x_k - \frac{f(x_k)}{\lambda}$  (ce qui suppose que  $f(x_k)$  soit bien définie, i.e.  $x_k \in [a, b]$ ). Comme  $\lambda \neq 0$  et  $f$  continue, l'hypothèse  $(x_k)$  converge vers  $\alpha$  entraîne que  $f(\alpha) = 0$ .

Pour définir l'ordre de convergence, on suppose de plus que  $\forall k \in \mathbb{N}$ ,  $x_k \neq \alpha$ . On peut alors appliquer la formule de Taylor-Lagrange : il existe  $\xi_k$  compris entre  $x_k$  et  $\alpha$  tel que

$$f(x_k) = \underbrace{f(\alpha)}_{=0} + (x_k - \alpha)f'(\xi_k) = (x_k - \alpha)f'(\xi_k).$$

On a alors en utilisant cette expression dans la définition de la suite  $x_k$

$$x_{k+1} = x_k - (x_k - \alpha) \frac{f'(\xi_k)}{\lambda}.$$

En soustrayant  $\alpha$  à cette équation on obtient

$$x_{k+1} - \alpha = x_k - \alpha - (x_k - \alpha) \frac{f'(\xi_k)}{\lambda} = (x_k - \alpha) \left(1 - \frac{f'(\xi_k)}{\lambda}\right).$$

Comme  $x_k \neq \alpha$ , on a alors

$$\frac{x_{k+1} - \alpha}{x_k - \alpha} = 1 - \frac{f'(\xi_k)}{\lambda}.$$

Or  $x_k$  converge vers  $\alpha$  et  $\xi_k$  compris entre  $x_k$  et  $\alpha$ , ce qui entraîne que  $\xi_k$  converge vers  $\alpha$ . La fonction  $f'$  étant continue, on en déduit que  $f'(\xi_k)$  converge vers  $f'(\alpha)$ . Ceci donne donc

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{x_k - \alpha} = 1 - \frac{f'(\alpha)}{\lambda}.$$

La convergence est donc (au moins) d'ordre 1.

- **Order 2 :** La suite étant convergente, il existe  $k_0 \in \mathbb{N}$  tel que  $\forall k \geq k_0, x_k \in \mathcal{V}$ . Soit  $k \geq k_0$ , comme  $f \in \mathcal{C}^2(\mathcal{V})$ , on peut appliquer la formule de Taylor-Lagrange : il existe  $\eta_k \in \mathcal{V}$  compris entre  $x_k$  et  $\alpha$  que

$$f(x_k) = \underbrace{f(\alpha)}_{=0} + (x_k - \alpha)f'(\alpha) + \frac{(x_k - \alpha)^2}{2!}f^{(2)}(\eta_k).$$

On a alors en utilisant cette expression dans la définition de la suite  $x_k$

$$x_{k+1} = x_k - \frac{1}{\lambda} \left( (x_k - \alpha)f'(\alpha) + \frac{(x_k - \alpha)^2}{2!}f^{(2)}(\eta_k) \right).$$

En soustrayant  $\alpha$  à cette équation on obtient

$$x_{k+1} - \alpha = (x_k - \alpha) \underbrace{\left( 1 - \frac{f'(\alpha)}{\lambda} \right)}_{=0 \text{ par hyp.}} + \frac{1}{\lambda} \frac{(x_k - \alpha)^2}{2!} f^{(2)}(\eta_k).$$

Comme  $\eta_k \in \mathcal{V}$  converge vers  $\alpha$  (car compris entre  $x_k$  et  $\alpha$ ) et  $f^{(2)}$  continue sur  $\mathcal{V}$ , on en déduit

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^2} = \frac{f^{(2)}(\alpha)}{2\lambda}.$$

La convergence est donc (au moins) d'ordre 2. □

On présente en Algorithme 2.12 l'implémentation usuelle de la méthode de la corde. Une autre version utilisant la fonction `PtFixe` est donnée en Algorithme 2.13.

---

**Algorithm 2.12** Méthode de la corde
 

---

**Données :**

$f$  :  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $a, b$  : deux réels tels que  $f(a) \neq f(b)$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $\text{tol}$  : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 $\text{kmax}$  : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $|f(\alpha_{\text{tol}})| \leq \text{tol}$

```

1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{Corde}(f, a, b, x_0, \text{tol}, \text{kmax})$ 
2:    $k \leftarrow 0, \alpha_{\text{tol}} \leftarrow \emptyset$ 
3:    $q \leftarrow \frac{b-a}{f(b)-f(a)}$ 
4:    $x \leftarrow x_0,$ 
5:    $\text{err} \leftarrow \text{tol} + 1$ 
6:   Tantque  $\text{err} > \text{tol}$  et  $k \leq \text{kmax}$  faire
7:      $k \leftarrow k + 1$ 
8:      $x_p \leftarrow x$ 
9:      $x \leftarrow x_p - q * f(x_p)$ 
10:     $\text{err} \leftarrow |x - x_p|$ 
11:   Fin
12:   Si  $\text{err} \leq \text{tol}$  alors ▷ Convergence
13:      $\alpha_{\text{tol}} \leftarrow x$ 
14:   Fin
15: Fin

```

---



---

**Algorithm 2.13** Méthode de la corde utilisant la fonction `PtFixe`


---

**Données :**

$f$  :  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $a, b$  : deux réels tels que  $f(a) \neq f(b)$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $\text{tol}$  : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 $\text{kmax}$  : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

$\alpha_{\text{tol}}$  : un réel tel que  $|f(\alpha_{\text{tol}})| \leq \text{tol}$

```

1: Fonction  $\alpha_{\text{tol}} \leftarrow \text{Corde}(f, a, b, x_0, \text{tol}, \text{kmax})$ 
2:    $q \leftarrow \frac{b-a}{f(b)-f(a)}$ 
3:    $\Phi \leftarrow (x \mapsto x - q * f(x))$  ▷ définition de
   fonction
4:    $\alpha_{\text{tol}} \leftarrow \text{PtFixe}(\Phi, x_0, \text{tol}, \text{kmax})$ 
5: Fin

```

---

Pour illustrer ces résultats de convergence, on va rechercher la racine positive de  $f(x) = x^2 - 1$  par la méthode de la corde avec comme données

- exemple 1 :  $a = 0.000$ ,  $b = 2.000$ ,  $x_0 = 1.800$ ,
- exemple 2 :  $a = 0.5000$ ,  $b = 1.900$ ,  $x_0 = 1.800$ .

On représente en Figure 2.9 les itérations de la méthode de la corde pour l'exemple 1 à partir du graphe de la fonction  $f$  (figure de gauche) et à partir du graphe de la fonction  $\Phi$  (figure de droite). Même chose pour l'exemple 2 avec la Figure 2.10.

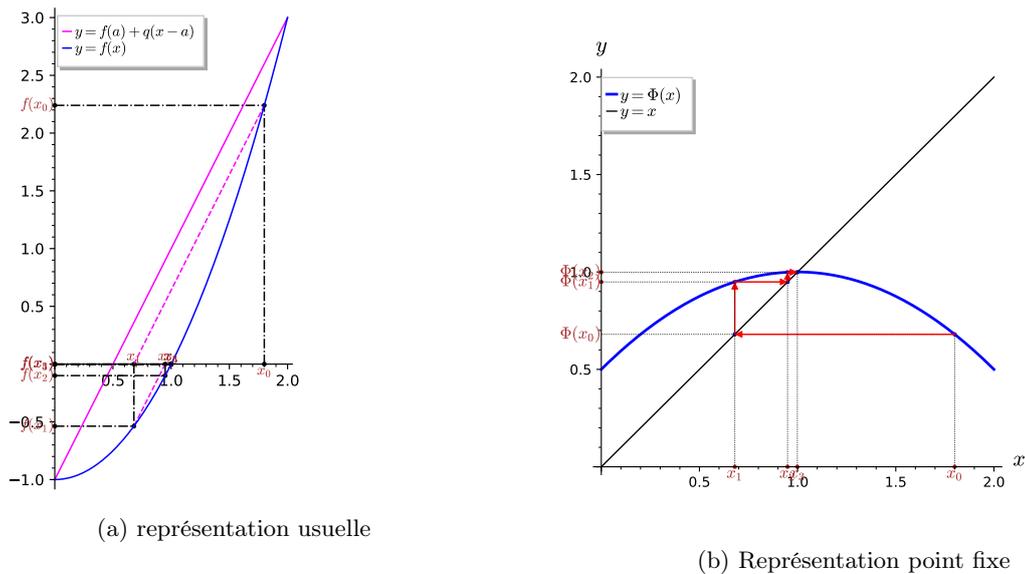


Figure 2.9: Exemple 1, méthode de la corde,  $\alpha = 1$ , racine de  $f : x \mapsto x^2 - 1$  avec  $a = 0.00$ ,  $b = 2.00$ ,  $x_0 = 1.80$ ,

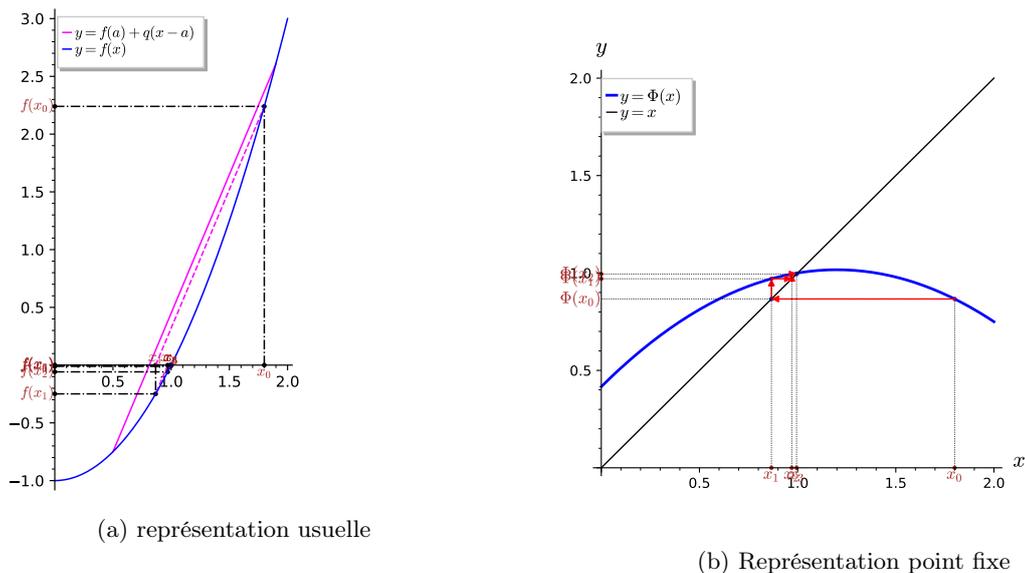


Figure 2.10: Exemple 2, méthode de la corde,  $\alpha = 1$ , racine de  $f : x \mapsto x^2 - 1$  avec  $a = 0.50$ ,  $b = 1.90$ ,  $x_0 = 1.80$ ,

Dans le tableau suivant on donne l'erreur commise par les suites dérivées des exemples 1 et 2 et ceci pour quelques itérations.

|          | exemple 1        | exemple 2        |
|----------|------------------|------------------|
| $k$      | $ x_k - \alpha $ | $ x_k - \alpha $ |
| 0        | 8.0000e-01       | 8.0000e-01       |
| 1        | 3.2000e-01       | 1.3333e-01       |
| 2        | 5.1200e-02       | 2.9630e-02       |
| 3        | 1.3107e-03       | 5.3041e-03       |
| 4        | 8.5899e-07       | 8.9573e-04       |
| 5        | 3.6893e-13       | 1.4962e-04       |
| 6        | 6.8056e-26       | 2.4947e-05       |
| $\vdots$ | $\vdots$         | $\vdots$         |
| 10       | 0.0000e+00       | 1.9250e-08       |

On remarque qu'avec les données de l'exemple 1 la convergence est beaucoup plus rapide. Ceci est

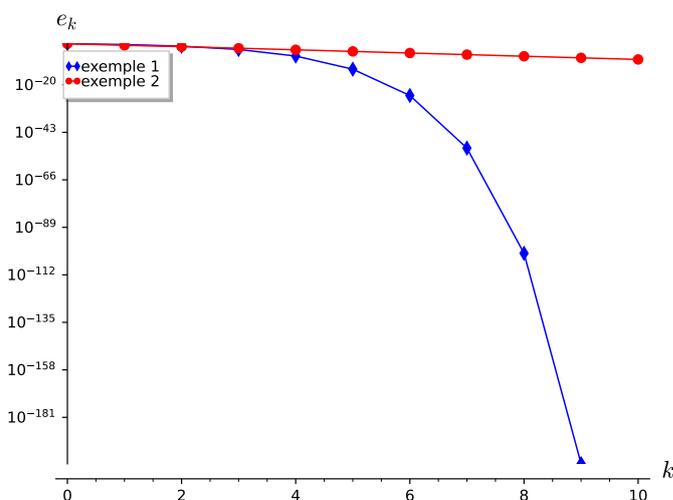


Figure 2.11: Erreurs en fonctions des itérations

illustré en Figure 2.11. En effet dans ce cas on a

$$\frac{f(b) - f(a)}{b - a} = 2 \quad \text{et} \quad f'(\alpha) = 2.$$

D'après la proposition 2.2.3, la convergence est d'ordre 2 pour l'exemple 1. Par contre, on a pour l'exemple 2

$$\frac{f(b) - f(a)}{b - a} = 2.400 \neq f'(\alpha) = 2$$

et donc la convergence est d'ordre 1. On retrouve ces résultats sur la Figure 2.12 où l'on a représenté en échelle logarithmique  $e_{k+1} = |x_{k+1} - \alpha|$  en fonction de  $e_k = |x_k - \alpha|$ . En effet si une méthode est convergente d'ordre  $p$  exactement on aura pour  $k$  suffisamment grand  $e_{k+1} \approx C e_k^p$ .

[Ajouter un autre exemple](#)

### La méthode de Newton

Soient  $f$  une fonction de classe  $\mathcal{C}^1$  et  $x_0$  un réel donné. La suite obtenue par la méthode de Newton est donnée par

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad \forall k \in \mathbb{N}. \quad (2.17)$$

Bien évidemment, il faudra s'assurer que  $f'(x_k) \neq 0$ . On peut voir cette relation comme un cas particulier de la méthode du point fixe. En effet, en prenant  $\Phi(x) = x - \frac{f(x)}{f'(x)}$ , la suite définie en (2.19) s'écrit  $x_{k+1} = \Phi(x_k)$ .

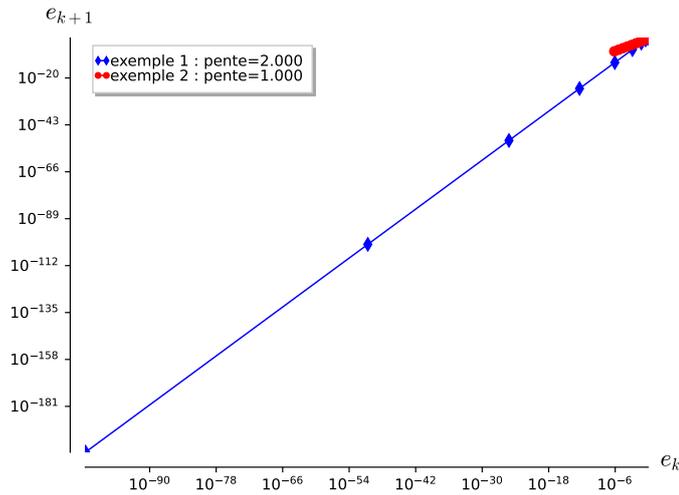


Figure 2.12: Représentation en échelle logarithmique de  $e_{k+1}$  en fonction de  $e_k$ . Les pentes sont calculées numériquement

**Proposition 2.2.4** (convergence de la méthode de Newton). *Soit  $f$  une fonction de classe  $\mathcal{C}^2$  sur un certain voisinage d'une racine simple  $\alpha$  de  $f$ . Soit  $x_0$  donné dans ce voisinage, la suite  $(x_k)_{k \in \mathbb{N}}$  définie par la méthode de Newton*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad \forall k \in \mathbb{N}. \quad (2.18)$$

*est localement convergente d'ordre 2.*

*Preuve.* Comme  $\alpha$  est racine simple de  $f$  (i.e.  $f(\alpha) = 0$  et  $f'(\alpha) \neq 0$ ) et  $f'$  continue, il existe un voisinage  $\mathcal{V}$  de  $\alpha$  tel que pour tout  $x \in \mathcal{V}$ ,  $f'(x) \neq 0$ . On peut alors définir la fonction  $\Phi$  sur  $\mathcal{V}$  par

$$\forall x \in \mathcal{V}, \quad \Phi(x) = x - \frac{f(x)}{f'(x)}.$$

On a alors  $x_{k+1} = \Phi(x_k)$ . La fonction  $\Phi$  est de classe  $\mathcal{C}^1$  sur  $\mathcal{V}$  et

$$\Phi'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}.$$

On a donc  $\Phi'(\alpha) = 0$  car  $f(\alpha) = 0$ . D'après le théorème de convergence locale du point fixe (théorème 2.2.3), on en déduit que la suite  $(x_k)$  converge vers  $\alpha$  (et que la convergence est au moins d'ordre 1). Pour démontrer qu'elle est d'ordre 2, on ne peut utiliser le théorème 2.2.1 car  $\Phi$  n'est pas de classe  $\mathcal{C}^2$ . Toutefois comme  $f$  est de classe  $\mathcal{C}^2$ , on applique la formule de Taylor-Lagrange aux points  $x_k, \alpha$  (en supposant  $x_k \neq \alpha$ ) : il existe  $\xi_k$  compris entre  $x_k$  et  $\alpha$  tel que

$$\underbrace{f(\alpha)}_{=0} = f(x_k) + (\alpha - x_k)f'(x_k) + \frac{(\alpha - x_k)^2}{2!}f^{(2)}(\xi_k).$$

Comme  $f'(x_k) \neq 0$ , l'équation précédente s'écrit aussi

$$\begin{aligned} \alpha &= x_k - \frac{f(x_k)}{f'(x_k)} - (\alpha - x_k)^2 \frac{f^{(2)}(\xi_k)}{2f'(x_k)} \\ &= x_{k+1} - (\alpha - x_k)^2 \frac{f^{(2)}(\xi_k)}{2f'(x_k)}. \end{aligned}$$

On obtient alors

$$\frac{\alpha - x_{k+1}}{(\alpha - x_k)^2} = -\frac{f^{(2)}(\xi_k)}{2f'(x_k)}.$$

La fonction  $f$  est de classe  $\mathcal{C}^2$  et la suite  $\xi_k$  converge vers  $\alpha$  (car  $\xi_k$  compris entre  $x_k$  et  $\alpha$ ). Ceci entraîne par passage à la limite que

$$\lim_{k \rightarrow +\infty} \frac{x_{k+1} - \alpha}{(x_k - \alpha)^2} = \frac{f^{(2)}(\alpha)}{2f'(\alpha)}.$$

La convergence est donc d'ordre 2.  $\square$

Soit  $f$  une fonction de classe  $\mathcal{C}^2$  au voisinage de  $\alpha$ , racine de  $f$ . On suppose que  $f'(x) \neq 0$  pour tout  $x \in \mathcal{V}$  (i.e.  $\alpha$  racine simple de  $f$ ). Soit  $x_0 \in \mathcal{V}$  donné. La suite obtenue par la méthode de Newton est donnée par

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad \forall k \in \mathbb{N}. \quad (2.19)$$

On peut voir cette relation comme un cas particulier de la méthode du point fixe. En effet, en prenant  $\Phi(x) = x - \frac{f(x)}{f'(x)}$ , la suite définie en (2.19) s'écrit  $x_{k+1} = \Phi(x_k)$  et on note que  $\Phi(\alpha) = \alpha$  (i.e.  $\alpha$  point fixe de  $\Phi$ ).

De plus  $f$  étant de classe  $\mathcal{C}^3$  et sa dérivée non nulle sur  $\mathcal{V}$ , on obtient que  $\Phi$  est de classe  $\mathcal{C}^2$  sur  $\mathcal{V}$ , et  $\forall x \in \mathcal{V}$ ,

$$\Phi'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

et

$$\Phi''(x) = \frac{(f'(x)f''(x) + f(x)f^{(3)}(x))(f'(x))^2 - 2f(x)f'(x)(f''(x))^2}{(f'(x))^4}$$

On a alors

$$\Phi'(\alpha) = 0, \quad \Phi''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)}.$$

D'après la proposition 2.2.1, si  $f''(\alpha) \neq 0$  alors la **méthode de Newton est d'ordre 2**.

**Faire méthode de Newton modifiée dans le cas de racine de multiplicité  $m > 1$ .???**

### EXERCICE 2.2.3

En  $-1700$  av. J.-C., les babyloniens ne connaissaient que les nombres rationnels (fractions) et ils utilisaient le système sexagésimal (base 60). Pour approcher la valeur  $\sqrt{2}$ , ils utilisaient comme approximation (voir tablette YBC 7289)

$$\alpha = 1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = \frac{30547}{21600}$$

L'erreur commise est  $|\alpha - \sqrt{2}| \approx 5.994e - 7$ .



Q. 1

Comment feriez-vous pour trouver **à la main** une méthode permettant de trouver des nombres rationnels approchant  $\sqrt{2}$ .

Q. 2

Généraliser la méthode pour trouver une approximation rationnelle de  $\sqrt{a}$  où  $a$  est un réel positif.

Q. 3

Généraliser la méthode pour trouver une approximation rationnelle de  $\sqrt[n]{a}$  où  $a$  est un réel positif et  $n \in \mathbb{N}^*$ .

### Correction 2.2.3

R. 1

Il suffit de voir que  $\sqrt{2}$  est la racine positive de  $f(x) = x^2 - 2$  et d'appliquer la méthode de Newton par exemple. La suite des itérés de Newton s'écrit alors

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - 2}{2x_k} = \frac{x_k^2 + 2}{2x_k}$$

Avec  $x_0 = 1$ , on obtient

| $k$ | $x_k$             | $ \sqrt{2} - x_k $ |
|-----|-------------------|--------------------|
| 1   | $\frac{3}{2}$     | 8.57864e-02        |
| 2   | $\frac{17}{12}$   | 2.45310e-03        |
| 3   | $\frac{577}{408}$ | 2.12390e-06        |

Avec  $x_0 = \frac{5}{4}$ , on obtient

| $k$ | $x_k$                       | $ \sqrt{2} - x_k $ |
|-----|-----------------------------|--------------------|
| 1   | $\frac{57}{40}$             | 1.07864e-02        |
| 2   | $\frac{6449}{4560}$         | 4.08236e-05        |
| 3   | $\frac{83176801}{58814880}$ | 5.89203e-10        |

R. 2

Il suffit de voir que  $\sqrt{a}$  est la racine positive de  $f(x) = x^2 - a$  et d'appliquer la méthode de Newton par exemple. La suite des itérés de Newton s'écrit alors

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k^2 + a}{2x_k}$$

Avec  $a = 3$  et  $x_0 = 1$ , on obtient

| $k$ | $x_k$           | $ \sqrt{3} - x_k $ |
|-----|-----------------|--------------------|
| 1   | 2               | 2.67949e-01        |
| 2   | $\frac{7}{4}$   | 1.79492e-02        |
| 3   | $\frac{97}{56}$ | 9.20496e-05        |

R. 3

Il suffit de voir que  $\sqrt[n]{a}$  est la racine positive de  $f(x) = x^n - a$  et d'appliquer la méthode de Newton par exemple. La suite des itérés de Newton s'écrit alors

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^n - a}{nx_k^{n-1}} = \frac{(n-1)x_k^n + a}{nx_k^{n-1}}$$

Avec  $a = 3$ ,  $n = 4$  et  $x_0 = 1$ , on obtient

| $k$ | $x_k$                                                                         | $ \sqrt[4]{3} - x_k $ |
|-----|-------------------------------------------------------------------------------|-----------------------|
| 1   | $\frac{3}{2}$                                                                 | 1.83926e-01           |
| 2   | $\frac{97}{72}$                                                               | 3.11482e-02           |
| 3   | $\frac{115403137}{87616608}$                                                  | 1.06368e-03           |
| 4   | $\frac{236297297271008837816738085152257}{179546943199700984864483416264832}$ | 1.28780e-06           |

~~~~~ Fin correction ~~~~~

On présente en Algorithme 2.14 l'implémentation standard de la méthode de Newton. Une autre version utilisant la fonction `PtFixe` est donnée en Algorithme 2.15.

**Algorithm 2.14** Méthode de Newton**Données :**

$f$  :  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $df$  : la dérivée de  $f$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $tol$  : la tolérance,  $tol \in \mathbb{R}^+$ ,  
 $kmax$  : nombre maximum d'itérations,  $kmax \in \mathbb{N}^*$

**Résultat :**

$\alpha_{tol}$  : un réel tel que  $|\Phi(\alpha_{tol}) - \alpha_{tol}| \leq tol$   
1: **Fonction**  $\alpha_{tol} \leftarrow \text{Newton}(f, df, x_0, tol, kmax)$   
2:  $k \leftarrow 0, \alpha_{tol} \leftarrow \emptyset$   
3:  $x \leftarrow x_0$ ,  
4:  $err \leftarrow tol + 1$   
5: **Tantque**  $err > tol$  et  $k \leq kmax$  **faire**  
6:    $k \leftarrow k + 1$   
7:    $xp \leftarrow x$   
8:    $x \leftarrow xp - f(xp)/df(xp) \quad \triangleright df(xp) \neq 0$   
9:    $err \leftarrow |x - xp|$   
10: **Fin**  
11: **Si**  $err \leq tol$  **alors**  $\triangleright$  Convergence  
12:    $\alpha_{tol} \leftarrow x$   
13: **Fin**  
14: **Fin**

**Algorithm 2.15** Méthode de Newton scalaire**Données :**

$f$  :  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  
 $df$  : la dérivée de  $f$ ,  
 $x_0$  : donnée initiale,  $x_0 \in \mathbb{R}$ ,  
 $tol$  : la tolérance,  $tol \in \mathbb{R}^+$ ,  
 $kmax$  : nombre maximum d'itérations,  $kmax \in \mathbb{N}^*$

**Résultat :**

$\alpha_{tol}$  : un réel tel que  
1: **Fonction**  $\alpha_{tol} \leftarrow \text{Newton}(f, df, x_0, tol, kmax)$   
2:  $\Phi \leftarrow (x \mapsto x - f(x)/df(x))$   
3:  $\alpha_{tol} \leftarrow \text{PtFixe}(\Phi, x_0, tol, kmax)$   
4: **Fin**

Comme premier exemple, on prend  $f(x) = x^2 - 1$  avec  $x_0 = 0.40$  et pour le second  $f(x) = x^2 \cos(x)$  avec  $x_0 = 2.00$ .

On représente en Figures 2.13 and 2.14, respectivement pour les exemples 1 et 2, les itérations de la méthode de Newton à partir du graphe de la fonction  $f$  (figure de gauche) et à partir du graphe de la fonction  $\Phi$  (figure de droite).

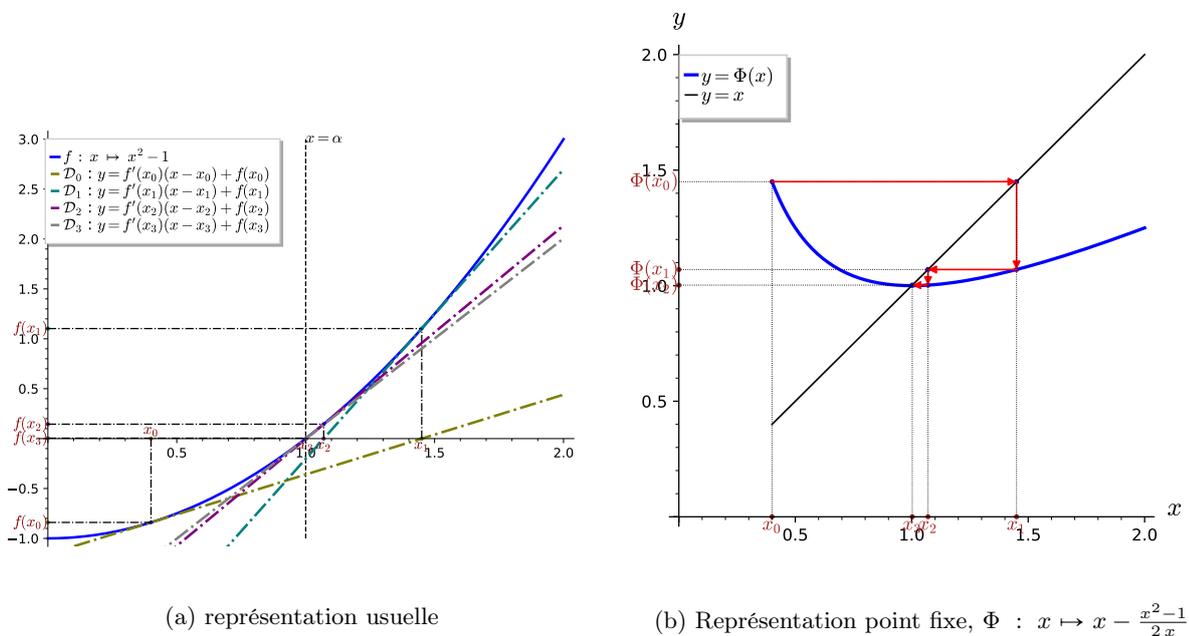


Figure 2.13: Exemple 1, méthode de Newton,  $\alpha = 1$ , racine de  $f : x \mapsto x^2 - 1$  avec  $x_0 = 0.40$ ,

On illustre la convergence et l'ordre de convergence respectivement en Figures 2.15a et 2.15b. Pour cette dernière, on a représenté en échelle logarithmique  $e_{k+1} = |x_{k+1} - \alpha|$  en fonction de  $e_k = |x_k - \alpha|$ . En effet si une méthode est convergente d'ordre  $p$  exactement on aura pour  $k$  suffisamment grand  $e_{k+1} \approx C e_k^p$ .

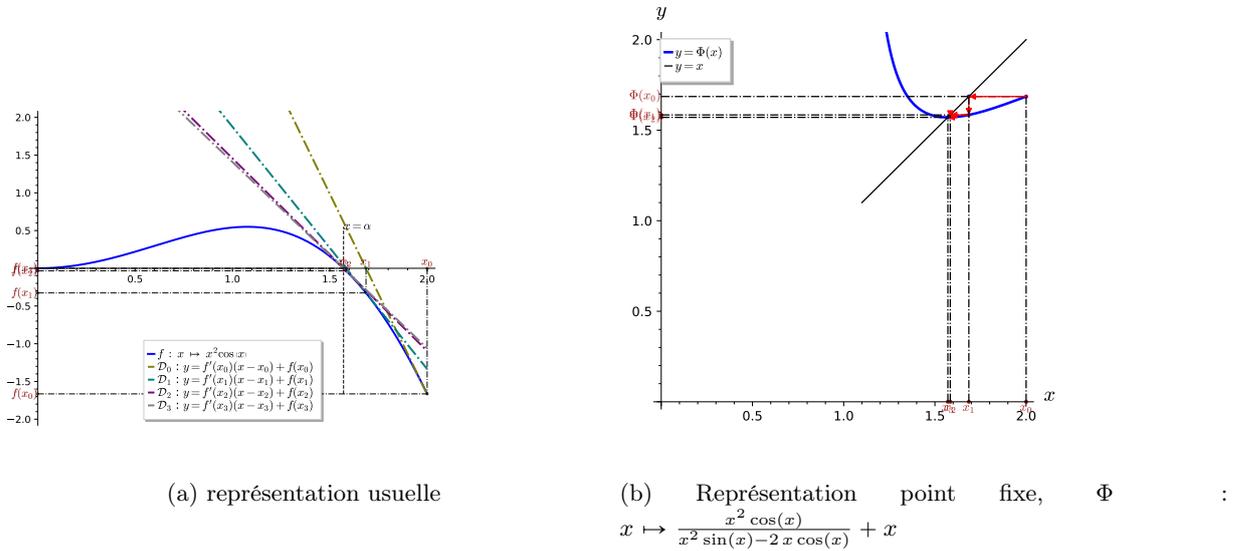


Figure 2.14: Exemple 2, méthode de Newton,  $\alpha = \frac{1}{2} \pi$ , racine de  $f : x \mapsto x^2 \cos(x)$  avec  $x_0 = 0.40$ ,

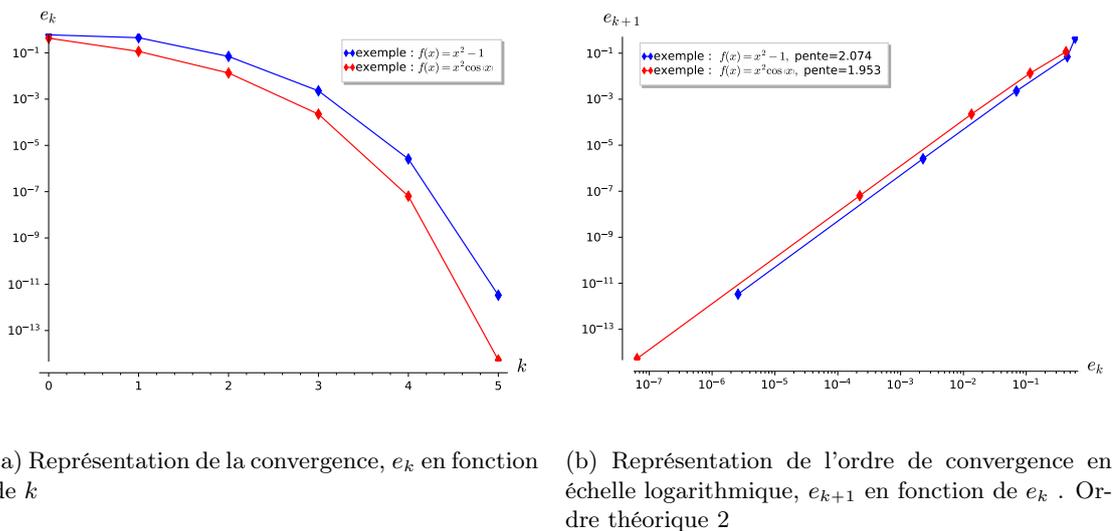


Figure 2.15: Méthode de Newton, convergence et ordre

## 2.2.5 La méthode de la sécante

Cette méthode est une alternative à la méthode de Newton lorsque l'on ne connaît pas la dérivée de la fonction  $f$ . A l'itération  $k$ , de la méthode de Newton, on approche  $f'(x_k)$  par  $\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ . En effet, d'après la formule de Taylor-Lagrange, il existe  $\xi_k$  compris entre  $x_{k-1}$  et  $x_k$  vérifiant

$$f(x_{k-1}) = f(x_k) + (x_{k-1} - x_k)f'(x_k) + \frac{(x_{k-1} - x_k)^2}{2!}f^{(2)}(\xi_k)$$

ce qui donne

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} + \frac{x_{k-1} - x_k}{2!}f^{(2)}(\xi_k).$$

Si la suite est convergente, on a  $\frac{x_{k-1} - x_k}{2!}f^{(2)}(\xi_k) \rightarrow 0$ , ce qui justifie l'approximation précédente. On a alors la méthode de la sécante donnée en (2.20). Cette méthode est une **méthode à deux pas** : le calcul de  $x_{k+1}$  nécessite de connaître  $x_k$  et  $x_{k-1}$ . Il faut donc deux valeurs d'initialisations  $x_{-1}$  et  $x_0$  pour définir la suite  $(x_k)$ .

**Proposition 2.2.5** (Convergence méthode de la sécante (Admis)). *Soit  $f$  une fonction de classe  $\mathcal{C}^2$  sur un certain voisinage d'une racine simple  $\alpha$  de  $f$ . Soient  $x_{-1}$  et  $x_0$  donnés dans ce voisinage tels que  $f(x_{-1}) \neq f(x_0)$ , la suite  $(x_k)_{k \in \mathbb{N}}$  définie par la méthode de la sécante*

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}f(x_k), \quad \forall k \in \mathbb{N}. \quad (2.20)$$

*est localement convergente d'ordre  $\frac{1+\sqrt{5}}{2} \approx 1.618$ .*

Comme premier exemple, on recherche une racine de  $x^2 - 1$  avec  $x_{-1} = 0.000$  et  $x_0 = 2.000$ . Une représentation graphique des premières itérés de la suite est donnée en Figure 2.16. Sur cette figure, les droites  $\mathcal{D}_k$  sont celles passant par les points  $(x_{k-1}, f(x_{k-1}))$  et  $(x_k, f(x_k))$ . Pour deuxième exemple, on

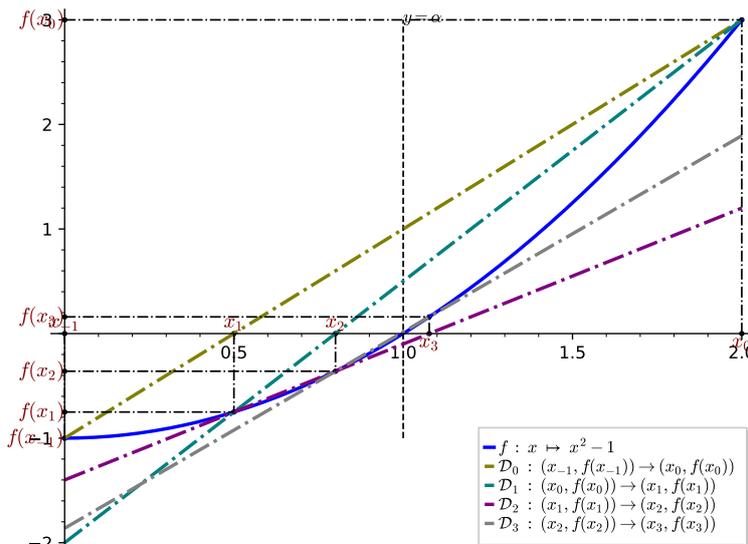


Figure 2.16: Méthode de la sécante pour  $f(x) = x^2 - 1$ ,  $x_{-1} = 0.000$  et  $x_0 = 2.000$

recherche une racine de  $x^2 \cos(x)$  avec  $x_{-1} = 1.000$  et  $x_0 = 3.000$ . Une représentation graphique des premières itérés de la suite est donnée en Figure 2.17.

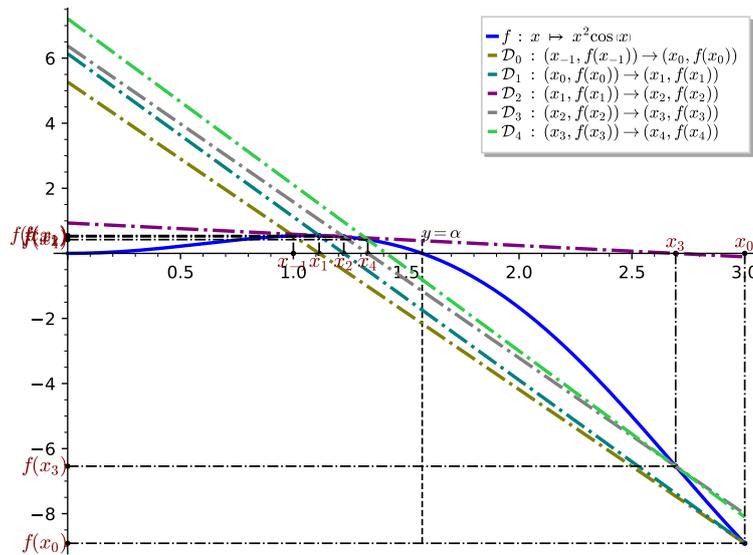
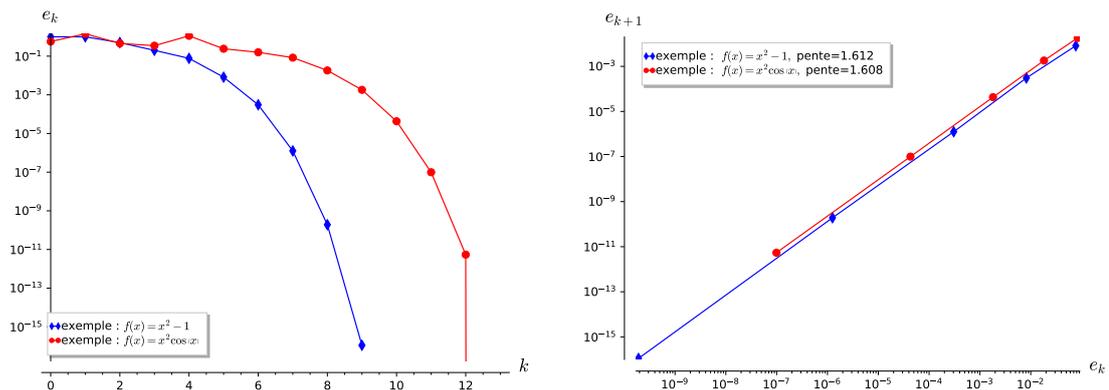


Figure 2.17: Méthode de la sécante pour  $f(x) = x^2 \cos(x)$ ,  $x_{-1} = 1.000$  et  $x_0 = 3.000$

On illustre la convergence et l'ordre de convergence respectivement en Figures 2.18a et 2.18b. Pour cette dernière, on a représenté en échelle logarithmique  $e_{k+1} = |x_{k+1} - \alpha|$  en fonction de  $e_k = |x_k - \alpha|$ . En effet si une méthode est convergente d'ordre  $p$  exactement on aura pour  $k$  suffisamment grand  $e_{k+1} \approx C e_k^p$ .



(a) Représentation de la convergence,  $e_k$  en fonction de  $k$

(b) Représentation de l'ordre de convergence en échelle logarithmique,  $e_{k+1}$  en fonction de  $e_k$ . Ordre théorique  $\frac{1+\sqrt{5}}{2} \approx 1.618$

Figure 2.18: Méthode de la sécante, convergence et ordre

## 2.2.6 Méthode Regula-Falsi ou fausse position

Cette méthode diffère de la méthode de dichotomie par le choix de  $x_k$  à chaque itération. Pour la méthode de dichotomie on a pris  $x_k = \frac{a_k + b_k}{2}$  point milieu du segment  $[a_k, b_k]$ . Pour la méthode Regula-Falsi, on prend pour  $x_k$  l'intersection de la droite passant par les points  $(a_k, f(a_k))$  et  $(b_k, f(b_k))$  avec l'axe des abscisses. Si  $f(a_k)f(b_k) < 0$  cela nous assure que  $x_k \in ]a_k, b_k[$ .

L'équation de la droite est donnée par

$$y = cx + d, \text{ avec } c = \frac{f(b_k) - f(a_k)}{b_k - a_k} \text{ et } d = \frac{b_k f(a_k) - a_k f(b_k)}{b_k - a_k}.$$

On a alors

$$x_k = -d/c = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}.$$

En résumé, on définit les trois suites  $(a_k)_{k \in \mathbb{N}}$ ,  $(b_k)_{k \in \mathbb{N}}$  et  $(x_k)_{k \in \mathbb{N}}$  par

- $a_0 = a$ ,  $b_0 = b$  et  $x_0 = \frac{a_0 f(b_0) - b_0 f(a_0)}{f(b_0) - f(a_0)}$ ,
- $\forall k \in \mathbb{N}$ ,
 
$$\begin{cases} a_{k+1} = b_{k+1} = x_{k+1} = x_k, & \text{si } f(x_k) = 0, \\ a_{k+1} = x_k, b_{k+1} = b_k, & \text{si } f(b_k)f(x_k) < 0, \\ a_{k+1} = a_k, b_{k+1} = x_k, & \text{si } f(a_k)f(x_k) < 0, \\ x_{k+1} = \frac{a_{k+1}f(b_{k+1}) - b_{k+1}f(a_{k+1})}{f(b_{k+1}) - f(a_{k+1})}, & \text{si } f(x_k) \neq 0. \end{cases}$$

### EXERCICE 2.2.4

On suppose que la fonction  $f$  est continue sur  $[a, b]$ , vérifie  $f(a)f(b) < 0$  et qu'il existe un unique  $\xi \in ]a, b[$  tel que  $f(\xi) = 0$ .

**Q. 1** Montrer que

$$a \leq \frac{af(b) - bf(a)}{f(b) - f(a)} \leq b.$$

**Q. 2** Montrer que  $a_0 \leq a_1 \leq \dots \leq a_k \leq x_k \leq b_k \leq \dots \leq b_1 \leq b_0$  pour tout  $k \in \mathbb{N}$ . et que si  $f(x_k) \neq 0$  alors  $f(a_k)f(b_k) < 0$ .

**Q. 3** En déduire la convergence de la suite  $(x_k)$  vers  $\xi$ .

### Correction 2.2.4

**R. 1**

On pose  $x = \frac{af(b) - bf(a)}{f(b) - f(a)}$  qui est bien défini car  $f(a) \neq f(b)$ . En effet si  $f(a) = f(b)$  alors  $f(a)f(b) = f(a)^2 \geq 0$  qui est en contradiction avec l'hypothèse  $f(a)f(b) < 0$ . On a

$$x - a = \frac{af(b) - bf(a) - a(f(b) - f(a))}{f(b) - f(a)} = (b - a) \frac{f(a)}{f(a) - f(b)}$$

Comme  $f(a)f(b) < 0$ ,  $f(a) - f(b)$  est du même signe que  $f(a)$  et alors  $\frac{f(a)}{f(a) - f(b)} \geq 0$ . De plus  $b - a > 0$  et donc on a  $x - a \geq 0$ .

De la même manière, on a

$$x - b = \frac{af(b) - bf(a) - b(f(b) - f(a))}{f(b) - f(a)} = (a - b) \frac{f(b)}{f(b) - f(a)}$$

Comme  $f(a)f(b) < 0$ ,  $f(b) - f(a)$  est du même signe que  $f(b)$  et alors  $\frac{f(b)}{f(b) - f(a)} \geq 0$ . De plus  $a - b < 0$  et donc on a  $x - b \leq 0$ .

**R. 2**

On va démontrer par récurrence la validité de la proposition  $(\mathcal{P}_k)$  suivante  $\forall k \in \mathbb{N}$  :

$$(\mathcal{P}_k) \begin{cases} (i) & f(a_k)f(b_k) < 0 \text{ si } f(x_k) \neq 0 \\ (ii) & a_0 \leq a_1 \leq \dots \leq a_k \leq x_k \leq b_k \leq \dots \leq b_1 \leq b_0, \end{cases}$$

**Initialisation** On vérifie la proposition  $(\mathcal{P}_k)$  pour  $k = 0$ . On a  $f(a)f(b) < 0$  donc  $(\mathcal{P}_0) - (i)$  est vérifiée.

D'après **Q. 1**, on obtient  $a_0 \leq x_0 \leq b_0$ . La proposition  $(\mathcal{P}_0)$  est donc vérifiée.

**Hérédité** Soit  $k \geq 1$ . On suppose la proposition  $(\mathcal{P}_k)$  est vraie. Montrons que  $(\mathcal{P}_{k+1})$  est vérifiée.

Si  $f(x_k) = 0$  alors  $a_{k+1} = b_{k+1} = x_{k+1} = x_k$  et la proposition  $(\mathcal{P}_{k+1})$  est vérifiée.

On suppose maintenant que  $f(x_k) \neq 0$ .

De  $(\mathcal{P}_k) - (i)$  on a  $f(a_k) \neq 0$  et  $f(b_k) \neq 0$ . De plus  $f(a_k) \neq f(b_k)$  car sinon  $f(a_k)f(b_k) = f(a_k)^2 \geq 0$  ce qui est en contradiction avec  $(\mathcal{P}_k) - (i)$ . Par hypothèse  $(\mathcal{P}_k)$ , on a  $a_k \leq x_k \leq b_k$  et  $f(a_k)f(b_k) < 0$ . Par continuité de  $f$ , on a alors soit  $f(b_k)f(x_k) < 0$  (et donc  $f(a_k)f(x_k) > 0$ ) soit  $f(b_k)f(x_k) > 0$  (et donc  $f(a_k)f(x_k) < 0$ ).

- Si  $f(b_k)f(x_k) < 0$  on a  $a_{k+1} = x_k$  et  $b_{k+1} = b_k$ . Comme  $f(a_k) \neq f(b_k)$ ,  $x_{k+1}$  est bien défini. D'après **Q. 1** en prenant  $[a_{k+1}, b_{k+1}]$  comme intervalle, et sachant que  $f(a_{k+1})f(b_{k+1}) = f(x_k)f(b_k) < 0$  on obtient

$$a_{k+1} \leq x_{k+1} \leq b_{k+1}.$$

De plus par hypothèse  $a_k \leq x_k = a_{k+1}$  et donc  $a_k \leq a_{k+1}$  et  $b_{k+1} \leq b_k$ . La proposition ( $\mathcal{P}_{k+1}$ ) est donc vérifiée.

- Si  $f(a_k)f(x_k) < 0$  on a  $a_{k+1} = a_k$  et  $b_{k+1} = x_k$ . Comme  $f(a_k) \neq f(b_k)$ ,  $x_{k+1}$  est bien défini. D'après **Q. 1** en prenant  $[a_{k+1}, b_{k+1}]$  comme intervalle, et sachant que  $f(a_{k+1})f(b_{k+1}) = f(a_k)f(x_k) < 0$  on obtient

$$a_{k+1} \leq x_{k+1} \leq b_{k+1}.$$

La proposition ( $\mathcal{P}_{k+1}$ ) est donc vérifiée.

**R. 3**

**Supposons qu'il existe  $s \in \mathbb{N}$  tel que  $f(x_s) = 0$ .** Alors, pour tout  $i \leq 1$ , on a  $a_{s+i} = b_{s+i} = x_{s+i} = x_s$ . Les trois suites convergent donc vers  $x_s$ . D'après la question précédente,  $x_s \in ]a, b[$ . Comme  $f(a) \neq 0$  et  $f(b) \neq 0$ , on en déduit  $x_s \in ]a, b[$ . Par hypothèse il existe un unique  $\xi \in ]a, b[$  tel que  $f(\xi) = 0$ , on a alors  $x_s = \xi$ .

**Supposons que  $\forall k \in \mathbb{N}$ ,  $f(x_k) \neq 0$ .** D'après **Q. 2**, la suite  $(a_k)$  est croissante majorée par  $b$  et la suite  $(b_k)$  est décroissante minorée par  $a$ . Elles sont donc convergentes et l'on note respectivement  $l$  et  $L$  les limites de  $(a_k)$  et  $(b_k)$ . Comme  $a \leq a_k \leq b_k \leq b$ , on a  $a \leq l \leq L \leq b$ .

- Supposons  $f(l) = f(L)$ . On a  $f(a_k)f(b_k) < 0$ . Comme  $f$  est continue, à la limite on obtient  $f(l)f(L) = f(l)^2 = f(L)^2 \leq 0$  et donc  $f(l) = f(L) = 0$ . On a nécessairement  $l$  et  $L$  dans  $]a, b[$  car  $f(a)f(b) < 0$  et donc  $f(a)$  et  $f(b)$  non nuls. Par unicité du zéro de  $f$  dans  $]a, b[$  on obtient  $l = L = \xi$ . Comme  $a_k \leq x_k \leq b_k$ , on en déduit que la suite  $(x_k)$  converge aussi vers  $\xi$ .
- Supposons  $f(l) \neq f(L)$ . Par continuité de la fonction  $f$  la suite  $(x_k)$  converge alors vers  $M = \frac{lf(L) - Lf(l)}{f(L) - f(l)}$ . Comme  $a_k \leq x_k \leq b_k$  on a aussi

$$l \leq M = \frac{lf(L) - Lf(l)}{f(L) - f(l)} \leq L. \quad (\text{R4.1})$$

De plus ayant  $f(x_k) \neq 0 \forall k \in \mathbb{N}$ , on a  $f(a_k)f(b_k) < 0 \forall k \in \mathbb{N}$ . En passant à la limite et par continuité de  $f$  on obtient  $f(l)f(L) \leq 0$ .

Montrons que  $f(l) = 0$  ou  $f(L) = 0$ .

- Si  $f(l) < f(L)$ , alors de (R4.1) on obtient

$$l(f(L) - f(l)) \leq lf(L) - Lf(l) \leq L(f(L) - f(l))$$

ce qui donne  $lf(l) \geq Lf(l)$  et  $lf(L) \leq Lf(L)$  i.e.  $(l-L)f(l) \geq 0$  et  $(L-l)f(L) \leq 0$ . Comme  $L-l \geq 0$ , on en déduit  $f(l) \leq 0$  et  $f(L) \leq 0$ . Or  $f(l)f(L) \leq 0$ , ce qui donne  $f(l) = 0$  ou  $f(L) = 0$ .

- Si  $f(l) > f(L)$ , alors de (R4.1) on obtient

$$l(f(L) - f(l)) \geq lf(L) - Lf(l) \geq L(f(L) - f(l))$$

ce qui donne  $lf(l) \leq Lf(l)$  et  $lf(L) \geq Lf(L)$  i.e.  $(L-l)f(l) \geq 0$  et  $(L-l)f(L) \leq 0$ . Comme  $L-l \geq 0$ , on en déduit  $f(l) \geq 0$  et  $f(L) \geq 0$ . Or  $f(l)f(L) \leq 0$ , ce qui donne  $f(l) = 0$  ou  $f(L) = 0$ .

On a donc démontré que si  $f(l) \neq f(L)$ , alors  $f(l) = 0$  ou  $f(L) = 0$  et donc

- si  $f(l) = 0$  alors  $M = \frac{lf(L) - Lf(l)}{f(L) - f(l)} = l$  et donc  $f(M) = 0$ .
- si  $f(L) = 0$  alors  $M = \frac{lf(L) - Lf(l)}{f(L) - f(l)} = L$  et donc  $f(M) = 0$ .

Puisque  $l$  et  $L$  appartiennent à  $]a, b[$ , on a  $M \in ]a, b[$ . Par unicité du zéro de  $f$  sur  $]a, b[$  on en déduit que  $M = \xi$ .



On vient de démontrer le théorème suivant

**Théorème 2.2.4.** Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une fonction continue vérifiant  $f(a)f(b) < 0$  et admettant  $\alpha \in ]a, b[$  comme **unique** solution de  $f(x) = 0$ . Alors la suite  $(x_k)_{k \in \mathbb{N}}$  définie par la **méthode Regula-Falsi** converge vers  $\alpha$

On a aussi la proposition suivante

**Proposition 2.2.6** (ordre de la méthode de Regula-Falsi). Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une fonction continue vérifiant  $f(a)f(b) < 0$  et admettant  $\alpha \in ]a, b[$  comme **unique** solution de  $f(x) = 0$ . Si  $f$  est deux fois dérivable sur  $]a, b[$  et si  $f''$  est monotone sur  $]a, b[$  alors il existe  $C \in \mathbb{R}$  tel que

$$\lim_{k \rightarrow +\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|} \leq C \tag{2.21}$$

La méthode de Regula-Falsi est alors à convergence linéaire (d'ordre 1).

## 2.3 Résolution de systèmes non linéaires

Nous allons (très rapidement) introduire les premières notions permettant la résolution de systèmes d'équations non linéaires. Par exemple, dans  $\mathbb{R}^2$  nous allons regarder le problème suivant avec  $c$  une constante réelle

$$\begin{cases} f_1(x_1, x_2) = -x_1^3 + x_2 - \frac{1}{2} & = 0 \\ f_2(x_1, x_2) = \frac{1}{25} (10x_2 + 1)^2 + c - x_1 & = 0. \end{cases} \tag{2.22}$$

En Figures 2.19 et 2.21, on représente pour différentes valeurs de  $c$  les courbes  $f_1(x_1, x_2) = 0$  et  $f_2(x_1, x_2) = 0$  : les intersections de ces deux courbes sont les solutions du problème (2.22). Comme on le voit graphiquement, il peut y avoir, suivant les valeurs de  $c$ , 0, 1, 2 ou 4 solutions.

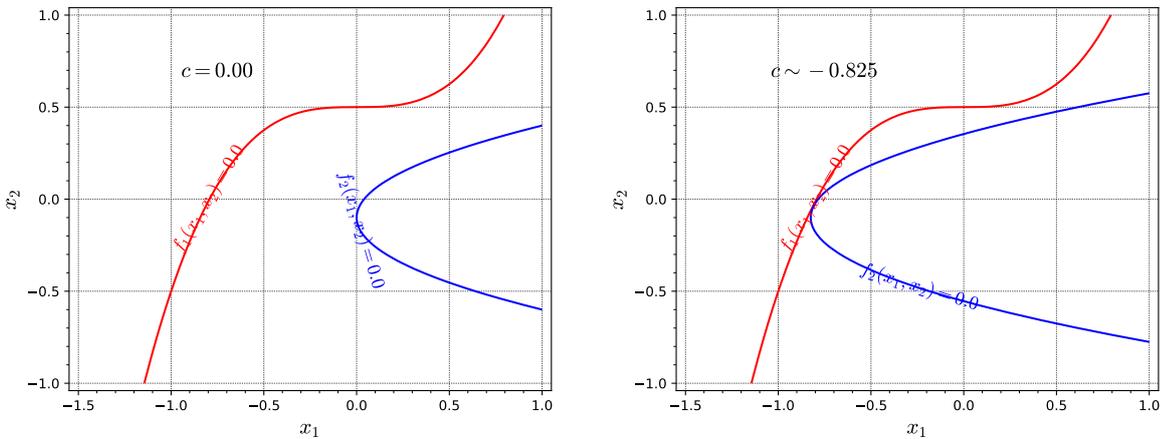


Figure 2.19: Résolution graphique de 2.22 avec  $c = 0.00$  (gauche) et  $c \sim -0.825$  (droite)

De manière plus générale, soient  $U \subset \mathbb{R}^N$  un ouvert et  $f$  une application continue de  $U$  dans  $\mathbb{R}^N$ . Le problème que l'on souhaite résoudre est le suivant

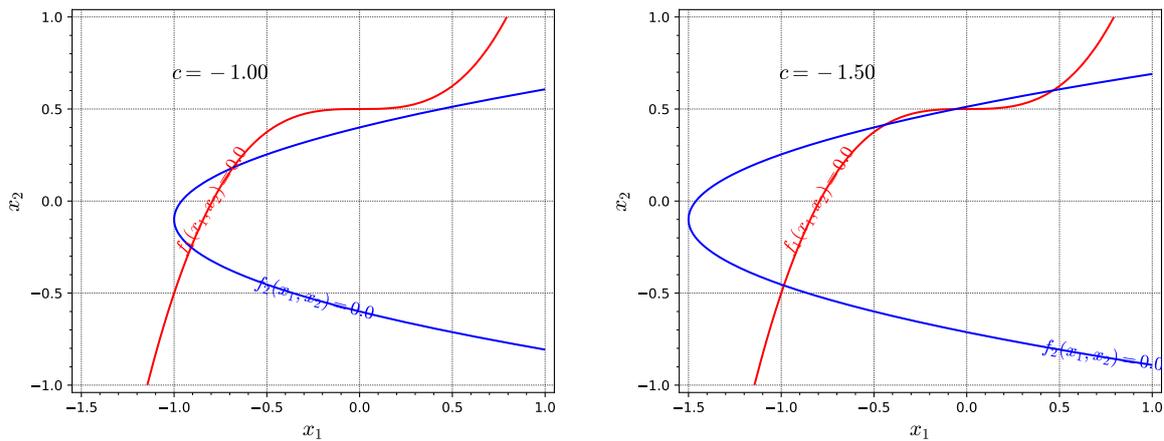


Figure 2.20: Résolution graphique de 2.22 avec  $c = -1.00$  (gauche) et  $c = -1.50$  (droite)

Trouver  $\alpha \in U \subset \mathbb{R}^N$  tel que

$$f(\alpha) = 0 \iff \begin{cases} f_1(\alpha_1, \dots, \alpha_N) = 0 \\ f_2(\alpha_1, \dots, \alpha_N) = 0 \\ \vdots \\ f_N(\alpha_1, \dots, \alpha_N) = 0 \end{cases}$$

Comme dans le cas scalaire, pour résoudre numériquement ce genre de problème on utilise des suites itératives et plus particulièrement celles basées sur les méthodes de points fixes. En effet, nous allons voir que le théorème du point fixe se généralise très facilement (voir Théorème 2.3.1).

En définissant, par exemple, la fonction  $\Phi \in C^0(U; \mathbb{R}^N)$  par  $\Phi(\mathbf{x}) = \mathbf{x} + \mathbf{f}(\mathbf{x})$ , on peut remarquer  $\mathbf{f}(\mathbf{x}) = 0$  est équivalent à  $\Phi(\mathbf{x}) = \mathbf{x}$ . On peut donc se ramener à la recherche d'un point fixe (s'il existe) de la fonction  $\Phi$ .

Trouver  $\alpha \in U \subset \mathbb{R}^N$  tel que

$$\Phi(\alpha) = \alpha \iff \begin{cases} \Phi_1(\alpha_1, \dots, \alpha_N) = \alpha_1 \\ \Phi_2(\alpha_1, \dots, \alpha_N) = \alpha_2 \\ \vdots \\ \Phi_N(\alpha_1, \dots, \alpha_N) = \alpha_N \end{cases}$$

Les suites itératives sont donc de la forme

$$\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]})$$

où  $\Phi$  est une fonction à déterminer et  $\mathbf{x}^{[0]} \in \mathbb{R}^N$ . Bien évidemment le choix d'une *bonne* fonction  $\Phi$  est primordiale pour espérer avoir convergence.

Ce type de problème peut s'avérer délicat à traiter : comment choisir  $\Phi$ ?  $\mathbf{x}^{[0]}$ ? Si l'on converge vers quel point fixe?

### 2.3.1 Point fixe

**Théorème 2.3.1.** Soit  $\mathcal{B}$  un espace de Banach et  $U \subset \mathcal{B}$  un sous-ensemble fermé. On suppose que  $\Phi : U \rightarrow U$  est une application strictement contractante, i.e.

$$\exists L \in [0, 1[, \quad \|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall (\mathbf{x}, \mathbf{y}) \in U \times U. \quad (2.23)$$

Alors

- $\Phi$  admet un unique point fixe  $\alpha \in U$  (i.e. unique solution de  $\mathbf{x} = \Phi(\mathbf{x})$ ).
- La suite des itérés  $\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]})$  converge vers  $\alpha$  pour toute valeur initiale  $\mathbf{x}^{[0]} \in U$ .
- Pour tout  $k \in \mathbb{N}$ ,

$$\|\alpha - \mathbf{x}^{[k]}\| \leq \frac{L^{k-l}}{1-L} \|\mathbf{x}^{[l+1]} - \mathbf{x}^{[l]}\|, \quad 0 \leq l \leq k \quad (2.24)$$

*Preuve.* On démontre tout d'abord l'existence d'un point fixe. Pour cela on va démontrer que la suite  $\mathbf{x}^{[k]}$  est de Cauchy dans  $U$  fermé d'un espace de Banach (donc elle converge dans  $U$ ). Comme  $\Phi$  est contractante, on a pour tout  $k \in \mathbb{N}^*$

$$\|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\| = \|\Phi(\mathbf{x}^{[k]}) - \Phi(\mathbf{x}^{[k-1]})\| \leq L \|\mathbf{x}^{[k]} - \mathbf{x}^{[k-1]}\|$$

ce qui donne par récurrence pour tout  $0 \leq j \leq k$

$$\|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\| \leq L^j \|\mathbf{x}^{[k+1-j]} - \mathbf{x}^{[k-j]}\| \quad (2.25)$$

ou encore pour tout  $0 \leq l \leq k$ , ( $l = k - j$ )

$$\|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\| \leq L^{k-l} \|\mathbf{x}^{[l+1]} - \mathbf{x}^{[l]}\| \quad (2.26)$$

On obtient aussi par récurrence

$$\forall l \geq 0, \|\mathbf{x}^{[k+1+l]} - \mathbf{x}^{[k+l]}\| \leq L^l \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|. \quad (2.27)$$

Soit  $p \geq 1$ . On en déduit par application répétée de l'inégalité triangulaire que

$$\begin{aligned} \|\mathbf{x}^{[k+p]} - \mathbf{x}^{[k]}\| &= \|(\mathbf{x}^{[k+p]} - \mathbf{x}^{[k+p-1]}) + (\mathbf{x}^{[k+p-1]} - \mathbf{x}^{[k+p-2]}) + \dots + (\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]})\| \\ &= \left\| \sum_{l=0}^{p-1} (\mathbf{x}^{[k+l+1]} - \mathbf{x}^{[k+l]}) \right\| \\ &\leq \sum_{l=0}^{p-1} \|\mathbf{x}^{[k+l+1]} - \mathbf{x}^{[k+l]}\| \\ &\stackrel{(2.27)}{\leq} \sum_{l=0}^{p-1} L^l \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\| = \frac{1-L^p}{1-L} \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\| \\ &\leq \frac{1-L^p}{1-L} L^k \|\mathbf{x}^{[1]} - \mathbf{x}^{[0]}\|. \quad (\text{en utilisant (2.26), avec } l=0) \end{aligned}$$

Comme  $L^k \rightarrow 0$  quand  $k \rightarrow +\infty$ , on conclut que  $(\mathbf{x}^{[k]})$  est une suite de Cauchy. De plus, par construction  $\mathbf{x}^{[k]} \in U \subset \mathcal{B}$ , pour tout  $k \in \mathbb{N}$ , et  $\mathcal{B}$  étant un espace de Banach et  $U$  un fermé, la suite  $(\mathbf{x}^{[k]})$  converge alors vers  $\alpha \in U$ . Comme  $\Phi$  est contractante, elle est donc continue et en passant à la limite dans  $\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]})$ , on aboutit à  $\alpha = \Phi(\alpha)$ , i.e.  $\alpha$  est un point fixe de  $\Phi$  dans  $U$ .

L'unicité se déduit immédiatement par la contraction de la fonction  $\Phi$ . En effet, soit  $\alpha_1$  et  $\alpha_2$  deux points fixes de  $\Phi$ , alors

$$\|\alpha_1 - \alpha_2\| = \|\Phi(\alpha_1) - \Phi(\alpha_2)\| \leq L \|\alpha_1 - \alpha_2\|$$

Or  $L < 1$ , et donc nécessairement on a  $\alpha_1 = \alpha_2$ .

Il reste à démontrer l'inégalité (2.24). On a vu que pour  $p \geq 1$

$$\|\mathbf{x}^{[k+p]} - \mathbf{x}^{[k]}\| \leq \frac{1-L^p}{1-L} \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|$$

L'application norme étant continue et  $L < 1$ , on obtient à la limite quand  $p \rightarrow +\infty$

$$\|\alpha - \mathbf{x}^{[k]}\| \leq \frac{1}{1-L} \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|$$

On obtient l'inégalité en utilisant (2.26). □

On déduit immédiatement l'algorithme du point fixe vectoriel à partir de de l'algorithme scalaire 2.10:

---

**Algorithm 2.16** Méthode de point fixe vectorielle
 

---

**Données :**

- $\Phi$  :  $\Phi : \mathbb{K}^N \longrightarrow \mathbb{K}^N$ ,  
 $\mathbf{x0}$  : donnée initiale,  $\mathbf{x0} \in \mathbb{K}^N$ ,  
 tol : la tolérance,  $\text{tol} \in \mathbb{R}^+$ ,  
 kmax : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

- $\alpha_{\text{tol}}$  : un réel tel que  $\|\Phi(\alpha_{\text{tol}}) - \alpha_{\text{tol}}\| \leq \text{tol}$

1: **Fonction**  $\alpha_{\text{tol}} \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x0}, \text{tol}, \text{kmax})$

2:  $k \leftarrow 0, \alpha_{\text{tol}} \leftarrow \emptyset$

3:  $\mathbf{x} \leftarrow \mathbf{x0}, \mathbf{fx} \leftarrow \Phi(\mathbf{x0}),$

4:  $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$

$\triangleright$  ou  $\frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$

5: **Tantque**  $\text{err} > \text{tol}$  et  $k \leq \text{kmax}$  **faire**

6:  $k \leftarrow k + 1$

7:  $\mathbf{x} \leftarrow \mathbf{fx}$

8:  $\mathbf{fx} \leftarrow \Phi(\mathbf{x})$

9:  $\text{err} \leftarrow \|\mathbf{fx} - \mathbf{x}\|$

$\triangleright$  ou  $\frac{\|\mathbf{fx} - \mathbf{x}\|}{\|\mathbf{x}\| + 1}$

10: **Fin**

11: **Si**  $\text{err} \leq \text{tol}$  **alors**

12:  $\alpha_{\text{tol}} \leftarrow \mathbf{x}$

13: **Fin**

14: **Fin**

---

### 2.3.2 Méthode de Newton

On commence par rappeler un résultat de calcul différentiel. Soit  $\mathbf{f} : \mathbb{R}^N \longrightarrow \mathbb{R}^N$  une fonction suffisamment régulière. On définit la **matrice Jacobienne de  $\mathbf{f}$** , notée  $\mathbb{J}_{\mathbf{f}}$ , par

$$\mathbb{J}_{\mathbf{f}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix}$$

On a alors  $\forall \mathbf{h} \in \mathbb{R}^N$  à l'ordre 1

$$\mathbf{f}(\mathbf{x} + \mathbf{h}) \approx \mathbf{f}(\mathbf{x}) + \mathbb{J}_{\mathbf{f}}(\mathbf{x}).\mathbf{h}. \quad (2.28)$$

Nous voulons trouver  $\alpha$  tel que  $\mathbf{f}(\alpha) = 0$ . Si  $\mathbf{x}^{[k]}$  est proche de  $\alpha$ , alors en utilisant (2.28) avec  $\mathbf{x} = \mathbf{x}^{[k]}$  et  $\alpha = \mathbf{x}^{[k]} + \mathbf{h}$  (i.e.  $\mathbf{h} = \alpha - \mathbf{x}^{[k]}$ ) on obtient

$$\mathbf{f}(\alpha) \approx \mathbf{f}(\mathbf{x}^{[k]}) + \mathbb{J}_{\mathbf{f}}(\mathbf{x}^{[k]}).\mathbf{h}$$

Au lieu de résoudre  $\mathbf{f}(\mathbf{x}) = 0$ , on résout le système linéarisé

$$\mathbf{f}(\mathbf{x}^{[k]}) + \mathbb{J}_{\mathbf{f}}(\mathbf{x}^{[k]}).\tilde{\mathbf{h}} = 0$$

c'est à dire le système linéaire

$$\mathbb{J}_{\mathbf{f}}(\mathbf{x}^{[k]}).\tilde{\mathbf{h}} = -\mathbf{f}(\mathbf{x}^{[k]}). \quad (2.29)$$

On espère alors que  $\tilde{\mathbf{h}}$  est une bonne approximation de  $\mathbf{h}$  au sens où  $\mathbf{x}^{[k]} + \tilde{\mathbf{h}}$  est une meilleure approximation de  $\alpha$  que  $\mathbf{x}^{[k]}$ . On note alors  $\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} + \tilde{\mathbf{h}}$ . En posant  $\Phi(\mathbf{x}) = \mathbf{x} - ((\mathbb{J}_{\mathbf{f}}(\mathbf{x}))^{-1} \mathbf{f}(\mathbf{x}))$  la méthode de Newton s'écrit alors

$$\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]}) = \mathbf{x}^{[k]} - \left( (\mathbb{J}_{\mathbf{f}}(\mathbf{x}^{[k]}))^{-1} \mathbf{f}(\mathbf{x}^{[k]}) \right) \quad (2.30)$$

Cette formule est une généralisation de celle vue dans le cas scalaire (voir ??). Il faut noter qu'à chaque itération la matrice Jacobienne est modifiée et qu'il faut calculer le vecteur  $-(\mathbb{J}_{\mathbf{f}}(\mathbf{x}^{[k]}))^{-1} \mathbf{f}(\mathbf{x}^{[k]})$ . Numériquement, on ne calcule que très rarement l'inverse d'une matrice car cela est très coûteux en temps mais on résout le système linéaire (2.29) ce qui est bien plus efficace.

On admet dans ce cours le théorème suivant

**Théorème 2.3.2.** Soit  $f \in C^3(\mathbb{R}^N; \mathbb{R}^N)$ . On suppose que la matrice Jacobienne appliquée en  $\alpha$ ,  $J_f(\alpha)$  est inversible dans un voisinage de  $\alpha$ , avec  $f(\alpha) = 0$ . Alors pour tout  $x^{[0]}$  suffisamment proche de  $\alpha$  la suite définie par

$$x^{[k+1]} = x^{[k]} - \left( (J_f(x^{[k]}))^{-1} \right) f(x^{[k]})$$

converge vers  $\alpha$  et la convergence est d'ordre 2.

On donne ensuite l'algorithme 2.17 permettant de déterminer une approximation d'un point fixe d'une fonction  $f$ . Dans cet algorithme on suppose donnée la fonction `Solve` permettant de résoudre un système linéaire.

**Algorithm 2.17** Méthode de Newton

**Données :**

- $f$  :  $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$ ,
- $Jf$  : la matrice Jacobienne de  $f$ ,
- $x0$  : donnée initiale,  $x0 \in \mathbb{R}^N$ ,
- $tol$  : la tolérance,  $tol \in \mathbb{R}^+$ ,
- $kmax$  : nombre maximum d'itérations,  $kmax \in \mathbb{N}^*$

**Résultat :**

$\alpha_{tol}$  : un élément de  $\mathbb{R}^N$  proche de  $\alpha$ .

- 1: **Fonction**  $\alpha_{tol} \leftarrow \text{Newton}(f, Jf, x0, tol, kmax)$
- 2:  $k \leftarrow 0, \alpha_{tol} \leftarrow \emptyset$
- 3:  $x \leftarrow x0$ ,
- 4:  $err \leftarrow tol + 1$
- 5: **Tantque**  $err > tol$  et  $k \leq kmax$  **faire**
- 6:      $k \leftarrow k + 1$
- 7:      $xp \leftarrow x$
- 8:      $h \leftarrow \text{Solve}(Jf(xp), -f(xp))$       $\triangleright x \leftarrow \text{Solve}(A, b)$  : résoud le système linéaire  $Ax = b$
- 9:      $x \leftarrow xp + h$
- 10:      $err \leftarrow \text{Norm}(x - xp)$
- 11: **Fin**
- 12: **Si**  $err \leq tol$  **alors**      $\triangleright$  Convergence
- 13:      $\alpha_{tol} \leftarrow x$
- 14: **Fin**
- 15: **Fin**

**Remarque 2.3.1.** Si l'on ne connaît pas explicitement la Jacobienne de  $f$ , il est possible de calculer une approximation de celle-ci en utilisant des formules de dérivation numérique.

On peut aussi

**2.3.3** Exemples

**Exemple modèle**

Comme premier exemple, nous reprenons le système 2.22 avec  $c = -1.5$

$$\begin{cases} f_1(x_1, x_2) = -x_1^3 + x_2 - \frac{1}{2} & = 0 \\ f_2(x_1, x_2) = \frac{1}{25} (10x_2 + 1)^2 - x_1 - \frac{3}{2} & = 0. \end{cases} \quad (2.31)$$

On représente en Figure 2.21 les itérées successives pour 4 suites avec une initialisation différentes. On remarque qu'il est très difficile, si l'on n'est pas suffisamment proche d'un point fixe, de prédire vers lequel on converge.

En Figure 2.22a, on représente les bassins d'attraction pour les itérées de Newton associés au système 2.31 : à chaque point initial  $x_0 = (x, y)$  on associe le point fixe vers lequel la suite de Newton converge et chaque point fixe correspond une couleur. En Figure 2.22b, on représente le nombre d'itérations assurant

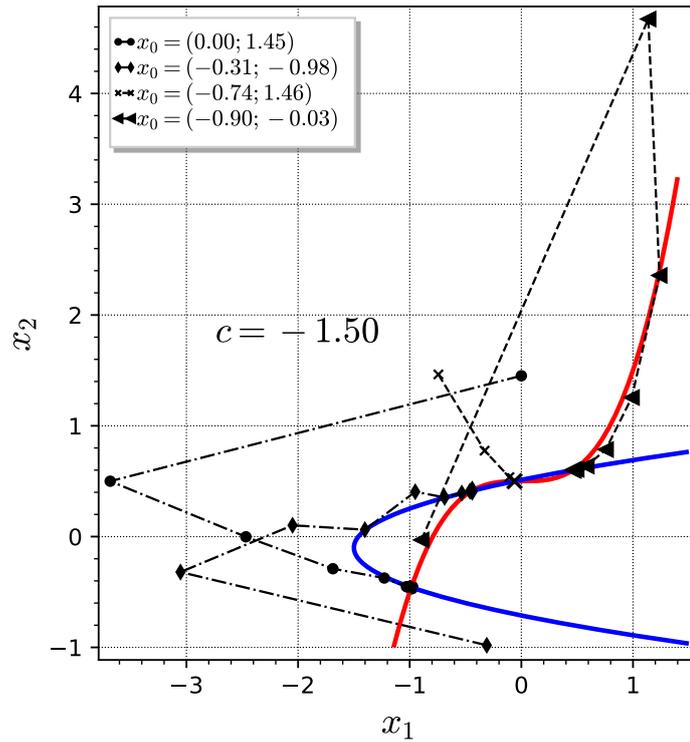
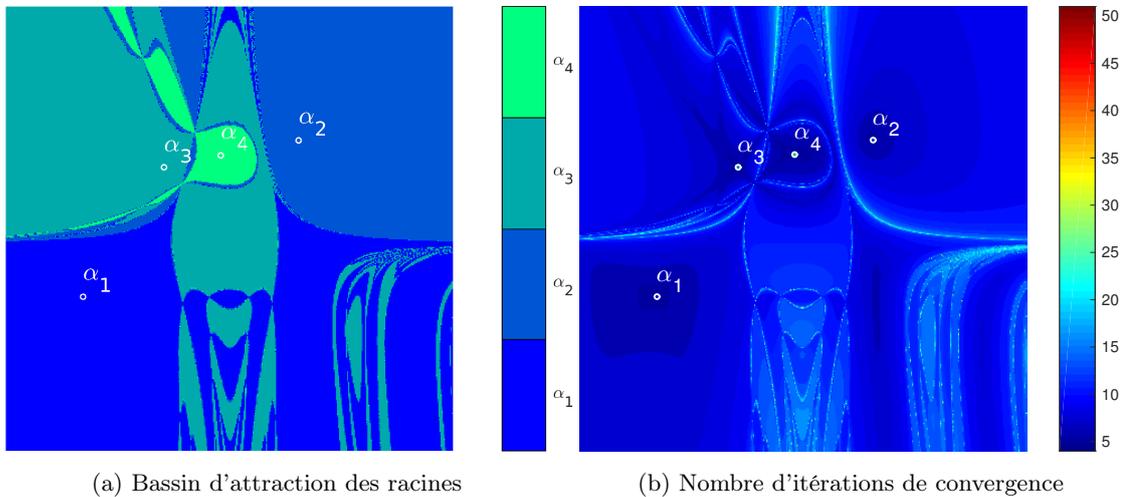


Figure 2.21: Représentation de 4 suites pour le système 2.31

la convergence des itérées de Newton : à chaque point initial  $x_0 = (x, y)$  on associe le nombre d'itérations nécessaire à la convergence et une échelle de couleur permet de visualiser ces nombres.



(a) Bassin d'attraction des racines

(b) Nombre d'itérations de convergence

Figure 2.22: Méthode de Newton, système (2.31)

**Exemple complexe :**  $z^3 - 1 = 0$

On souhaite trouver les racines complexes de  $z^3 - 1$ . Pour cela on peut poser  $z = x + iy$ , et le système équivalent devient

$$\begin{cases} f_1(x, y) = x^3 - 3xy^2 - 1 = 0 \\ f_2(x, y) = 3x^2y - y^3 = 0. \end{cases} \quad (2.32)$$

Bien évidemment en restant dans le corps des complexes, l'algorithme de Newton est le même (encore faut-il que le langage de programmation utilisé le supporte).

On représente en Figure 2.23 les bassins d'attraction et le nombre d'itérations de convergence associé à des données initiales dans  $[-1.5, 1.5] \times [-1.5, 1.5]$ . On obtient alors une *fractale de Newton*. Pour illustrer ce *caractère fractale* des représentations, on donne en Figures 2.24 et 2.25 des zooms successifs sur les graphes.

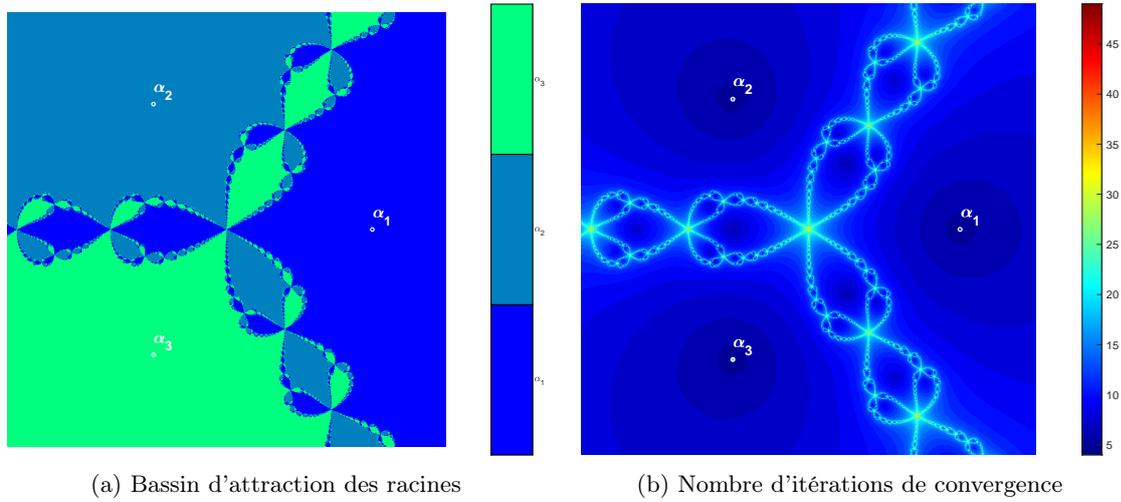


Figure 2.23: Méthode de Newton, système (2.32)

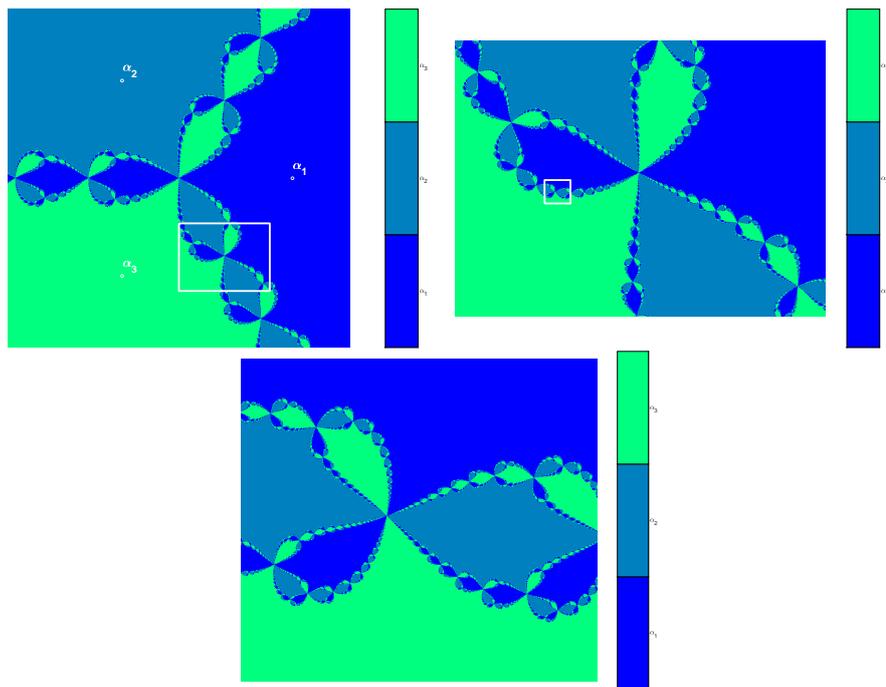


Figure 2.24: Méthode de Newton, système (2.32), zooms sur les bassins d'attraction

**Exemple complexe :**  $z^5 - 1 = 0$

On représente en Figure 2.26 les bassins d'attraction et le nombre d'itérations de convergence associé à des données initiales dans  $[-1.5, 1.5] \times [-1.5, 1.5]$ .

**Exemple complexe :**  $z^3 - 2z + 2 = 0$

On représente en Figure 2.27 les bassins d'attraction et le nombre d'itérations de convergence associé à des données initiales dans  $[-2, 2] \times [-2, 2]$ .

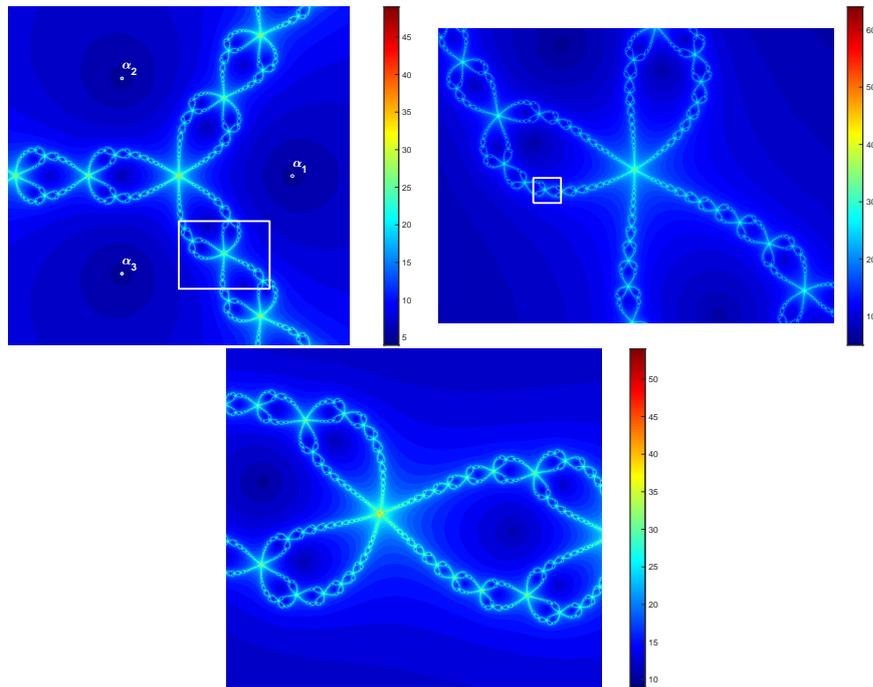
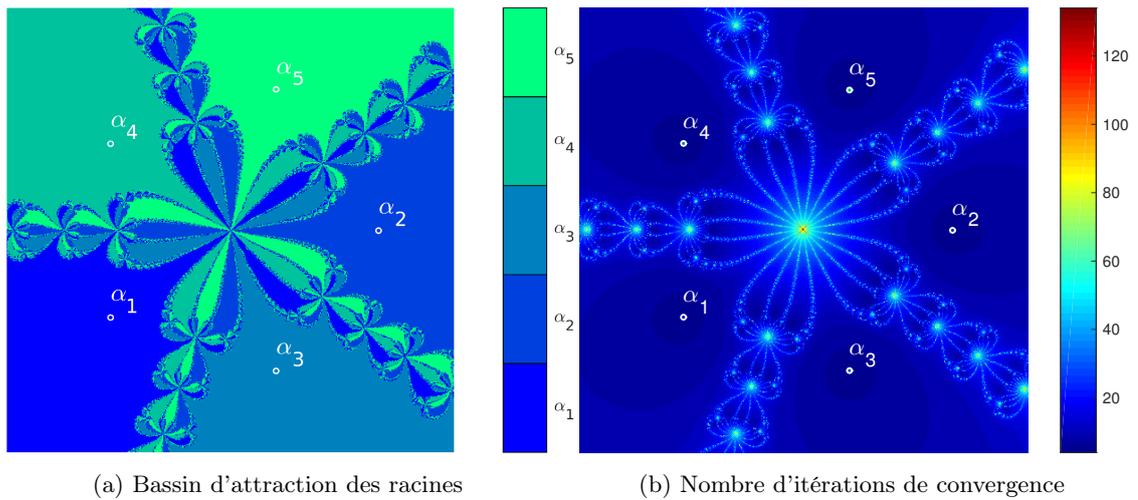


Figure 2.25: Méthode de Newton, système (2.32), zooms sur les nombres d'itérations



(a) Bassin d'attraction des racines

(b) Nombre d'itérations de convergence

Figure 2.26: Méthode de Newton pour  $z^5 - 1 = 0$

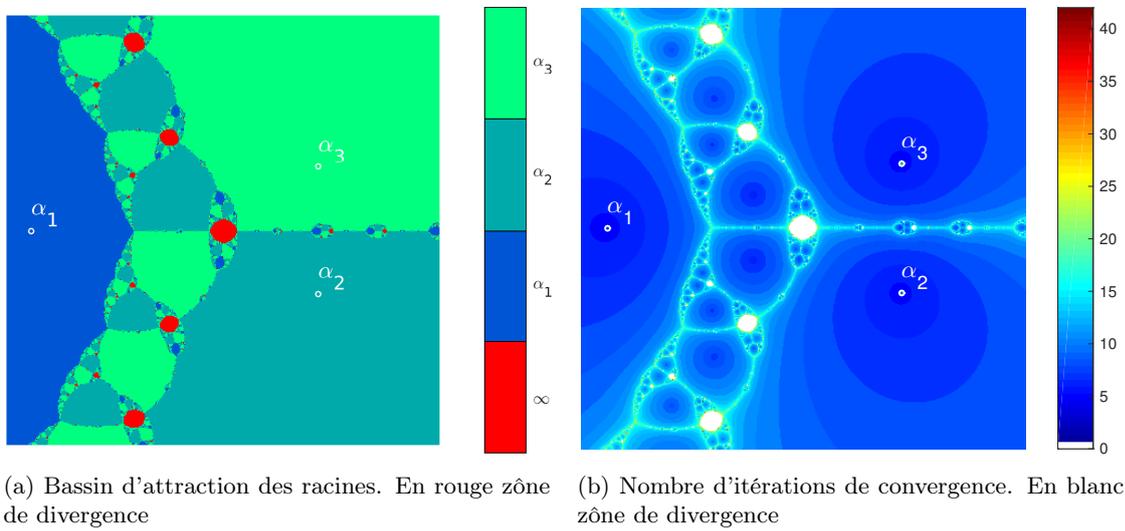


Figure 2.27: Méthode de Newton pour  $z^3 - 2z + 2 = 0$

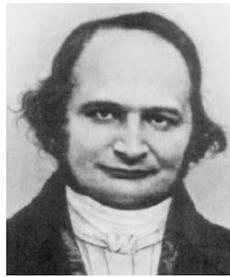


# Chapitre 3

## Résolution de systèmes linéaires



(a) *Carl Friedrich Gauss* 1777-1855, mathématicien allemand



(b) *Carl Gustav Jacobi* 1804-1851, mathématicien allemand



(c) *Philipp Ludwig von Seidel* 1879-1882, mathématicien allemand



(d) *André-Louis Cholesky* 1875-1918, mathématicien français



(e) *Wilhelm Jordan* 1842-1899, mathématicien allemand



(f) *Issai Schur* 1875-1941, mathématicien russe

Dans cette partie nous allons considérer la résolution numérique d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  dont la matrice  $\mathbb{A}$  est inversible.

On pourrait penser que pour résoudre le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ ,  $\mathbb{A}$  inversible, le plus simple serait de calculer la matrice  $\mathbb{A}^{-1}$ , inverse de  $\mathbb{A}$ , puis d'effectuer un produit matrice-vecteur pour obtenir  $\mathbf{x} = \mathbb{A}^{-1}\mathbf{b}$ . Or pour calculer l'inverse d'une matrice d'ordre  $n$  on doit résoudre  $n$  systèmes linéaires d'ordre  $n$ ! En effet, déterminer l'inverse d'une matrice  $\mathbb{A}$  revient à rechercher la matrice  $\mathbb{X}$  solution de

$$\mathbb{A}\mathbb{X} = \mathbb{I} \iff \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{n-1,n} \\ A_{n,1} & \dots & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,n} \\ X_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & X_{n-1,n} \\ X_{n,1} & \dots & X_{n,n-1} & X_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

Si on note  $\mathbf{X}_i$ , le  $i$ -ème vecteur colonne de la matrice  $\mathbb{X}$  et  $\mathbf{e}_i$  le  $i$ -ème de la base canonique de  $\mathbb{R}^n$  alors le système précédant s'écrit

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{n-1,n} \\ A_{n,1} & \dots & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_n \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_n \end{pmatrix}$$

Ce dernier système est alors équivalent à résoudre les  $n$  systèmes linéaires

$$\mathbb{A}\mathbf{X}_j = \mathbf{e}_j, \quad \forall j \in \llbracket 1, n \rrbracket.$$



Pour résoudre un système linéaire, on ne calcule pas la matrice inverse associée.

Nous allons en section 3.1 étudier quelques **méthodes directes** pour la résolution d'un système linéaire basées sur la recherche d'une matrice  $\mathbb{M}$  inversible telle que la matrice  $\mathbb{M}\mathbb{A}$  soit triangulaire supérieure. Ceci conduit à la résolution du système linéaire équivalent

$$\mathbb{M}\mathbb{A}\mathbf{x} = \mathbb{M}\mathbf{b}.$$

par la *méthode de la remontée* décrite en section 3.1.1).

En section 3.4, nous nous intéresserons aux **méthodes itératives** pour la résolution d'un système linéaire qui peuvent s'écrire sous la forme

$$\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}, \quad k \geq 0, \quad \mathbf{x}^{[0]} \text{ donné}$$

où la matrice  $\mathbb{B}$  et le vecteur  $\mathbf{c}$  sont construits à partir de la matrice  $\mathbb{A}$  et du vecteur  $\mathbf{b}$ . On espère alors avoir  $\lim_{k \rightarrow +\infty} \mathbf{x}^{[k]} = \mathbf{x}$ .

## 3.1 Méthodes directes

Pour résoudre le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ , nous allons le transformer en un système linéaire triangulaire supérieure équivalent

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

où  $\mathbb{M}$  est une matrice inversible telle que  $\mathbb{M}\mathbb{A}$  soit triangulaire supérieure. Nous allons voir que ce nouveau système est très facile à résoudre par la *méthode de la remontée*.

Nous étudierons la *méthode de Gauss-Jordan* que nous réécrirons sous forme algébrique. Puis nous ferons le lien avec les méthodes utilisant la *factorisation LU* et la *factorisation de Cholesky*. Nous finirons par la factorisation QR d'une matrice,

Nous allons tout d'abord regarder quelques cas particuliers : la matrice du système est diagonale, triangulaire inférieure ou triangulaire supérieure.

### 3.1.1 Matrices particulières

#### Matrices diagonales

Soit  $A$  une matrice de  $\mathcal{M}_n(\mathbb{K})$  diagonale inversible et  $\mathbf{b} \in \mathbb{K}^n$ . Dans ce cas les coefficients diagonaux de  $A$  sont tous non nuls et l'on a

$$x_i = b_i/A_{i,i}, \quad \forall i \in \llbracket 1, n \rrbracket. \quad (3.1)$$

On a immédiatement l'algorithme

**Algorithm 3.1** Fonction `RSLMatDiag` permettant de résoudre le système linéaire à matrice diagonale inversible

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

**Données :**  $A$  : matrice diagonale de  $\mathcal{M}_n(\mathbb{R})$  inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{RSLMatDiag}(A, \mathbf{b})$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $x(i) \leftarrow b(i)/A(i, i)$ 
4:   Fin
5: Fin

```

#### Matrices triangulaires inférieures

Soit  $A$  une matrice de  $\mathcal{M}_n(\mathbb{K})$  triangulaire inférieure inversible et  $\mathbf{b} \in \mathbb{K}^n$ . On veut résoudre le système linéaire

$$A\mathbf{x} = \mathbf{b} \iff \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ A_{n,1} & \dots & \dots & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

#### EXERCICE 3.1.1

Soit  $A \in \mathcal{M}_n(\mathbb{K})$  une matrice **triangulaire**. Montrer que

$$A \text{ inversible} \iff A_{i,i} \neq 0, \forall i \in \llbracket 1, n \rrbracket.$$

On remarque que l'on peut calculer successivement  $x_1, x_1, \dots, x_n$ , car il est possible de calculer  $x_i$  si on connaît  $x_1, \dots, x_{i-1}$  : c'est la **méthode de descente**. En effet, on a

$$(A\mathbf{x})_i = b_i, \quad \forall i \in \llbracket 1, n \rrbracket.$$

et donc, par définition d'un produit matrice-vecteur,

$$\sum_{j=1}^n A_{i,j}x_j = b_i, \quad \forall i \in \llbracket 1, n \rrbracket.$$

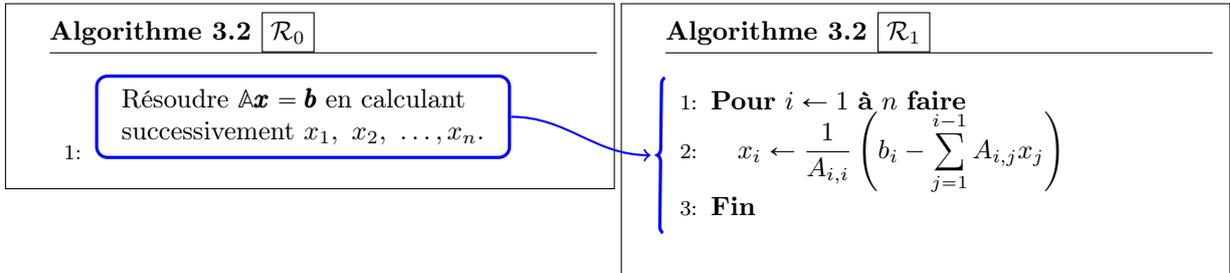
Comme  $A$  est une matrice triangulaire inférieure, on a (voir Définition B.2.24)  $A_{i,j} = 0$  si  $j > i$ . Ceci donne alors pour tout  $i \in \llbracket 1, n \rrbracket$

$$\begin{aligned} b_i &= \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \underbrace{\sum_{j=i+1}^n A_{i,j}}_{=0} x_j \\ &= \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i \end{aligned}$$

De plus la matrice  $\mathbb{A}$  étant triangulaire inversible ses éléments diagonaux sont tous non nuls. On obtient alors  $x_i$  en fonction des  $x_1, \dots, x_{i-1}$ :

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j} x_j \right), \quad \forall i \in \llbracket 1, n \rrbracket. \quad (3.2)$$

On écrit en détail les raffinements successifs permettant d'aboutir à l'Algorithme 3.2 final ne comportant que des opérations élémentaires (... à finaliser) de telle sorte que le passage entre deux raffinements successifs soit le plus compréhensible possible.

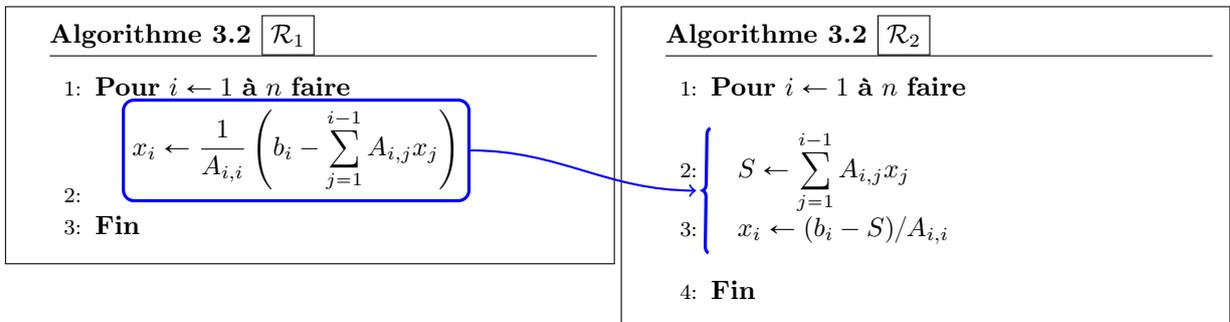


Dans le raffinement  $\mathcal{R}_1$ , la seule difficulté restante est le calcul de la somme  $\sum_{j=1}^{i-1} A_{i,j} x_j$ . En effet, l'opérateur mathématique  $\sum$  n'est pas défini dans notre langage algorithmique : il va donc falloir détailler un peu plus l'algorithme. Pour isoler le calcul de cette somme, on la note  $S$ . La ligne 2 peut donc s'écrire

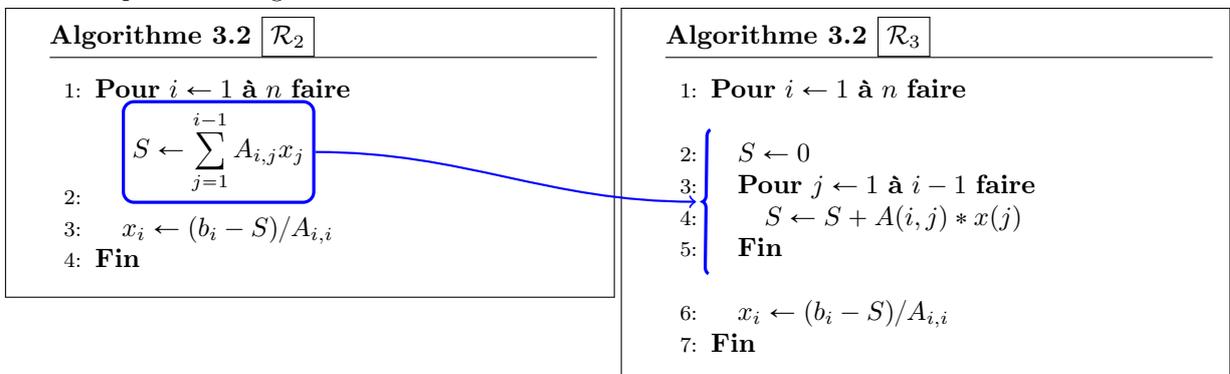
$$x_i \leftarrow \frac{1}{A_{i,i}} (b_i - S).$$

Mais où calculer la valeur  $S$ ? 4 choix possible : avant la ligne 1, entre les lignes 1 et 2, entre les lignes 2 et 3 ou après la ligne 3.

On ne peut pas calculer  $S$  après utilisation de sa valeur dans le calcul de  $x_i$ ! ce qui élimine les 2 derniers choix. Ensuite, on ne peut sortir le calcul de  $S$  de la boucle puisque la somme dépend de l'indice de boucle  $i$  : ce qui élimine le premier choix. On doit donc calculer  $S$  dans la boucle et avant le calcul de  $x_i$ .



Maintenant que l'on a isolé la *difficulté*, il reste à détailler le calcul de  $S$ . Celui-ci se fait **intégralement** en lieu et place de la ligne 2.



Insister sur  $S \leftarrow 0$  à l'intérieur de la boucle en  $i$ ? (erreur trop courante des débutants)

On obtient alors l'algorithme final

**Algorithm 3.2** Fonction **RSLTriInf** permettant de résoudre le système linéaire triangulaire inférieur inversible

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

**Données :**  $\mathbb{A}$  : matrice triangulaire de  $\mathcal{M}_n(\mathbb{K})$  inférieure inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{K}^n$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{RSLTriInf}(\mathbb{A}, \mathbf{b})$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $S \leftarrow 0$ 
4:     Pour  $j \leftarrow 1$  à  $i - 1$  faire
5:        $S \leftarrow S + A(i, j) * x(j)$ 
6:     Fin
7:      $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
8:   Fin
9: Fin

```

### Matrices triangulaires supérieures

Soit  $\mathbb{A}$  une matrice de  $\mathcal{M}_n(\mathbb{R})$  triangulaire supérieure inversible et  $\mathbf{b} \in \mathbb{R}^n$ . On veut résoudre le système linéaire

$$\mathbb{A}\mathbf{x} = \mathbf{b} \iff \begin{pmatrix} A_{1,1} & \dots & \dots & A_{1,n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

On remarque que l'on peut calculer successivement  $x_n, x_{n-1}, \dots, x_1$ , car il est possible de calculer  $x_i$  si on connaît  $x_{i+1}, \dots, x_n$  : c'est la **méthode de remontée**. En effet, on a

$$(\mathbb{A}\mathbf{x})_i = b_i, \forall i \in \llbracket 1, n \rrbracket.$$

et donc, par définition d'un produit matrice-vecteur,

$$\sum_{j=1}^n A_{i,j} x_j = b_i, \forall i \in \llbracket 1, n \rrbracket.$$

Comme  $\mathbb{A}$  est une matrice triangulaire supérieure, on a (voir Définition B.2.24)  $A_{i,j} = 0$  si  $j < i$ . Ceci donne alors pour tout  $i \in \llbracket 1, n \rrbracket$

$$\begin{aligned} b_i &= \sum_{j=1}^{i-1} \underbrace{A_{i,j}}_{=0} x_j + A_{i,i} x_i + \sum_{j=i+1}^n A_{i,j} x_j \\ &= A_{i,i} x_i + \sum_{j=i+1}^n A_{i,j} x_j \end{aligned}$$

De plus la matrice  $\mathbb{A}$  étant triangulaire inversible ses éléments diagonaux sont tous non nuls. On obtient donc  $x_i$  en fonction des  $x_{i+1}, \dots, x_n$ :

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=i+1}^n A_{i,j} x_j \right), \forall i \in \llbracket 1, n \rrbracket. \quad (3.3)$$

Algorithme 3.3  $\mathcal{R}_0$ 

1: Résoudre  $\mathbb{A}\mathbf{x} = \mathbf{b}$  en calculant successivement  $x_n, x_{n-1}, \dots, x_1$ .

Algorithme 3.3  $\mathcal{R}_1$ 

1: **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )  
 2:     calculer  $x_i$  connaissant  $x_{i+1}, \dots, x_n$   
       à l'aide de l'équation (3.3)  
 3: **Fin**

Algorithme 3.3  $\mathcal{R}_1$ 

1: **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )

2:     Calculer  $x_i$  connaissant  $x_{i+1}, \dots, x_n$   
       à l'aide de l'équation (3.3)

3: **Fin**

Algorithme 3.3  $\mathcal{R}_2$ 

1: **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )

2:      $S \leftarrow \sum_{j=i+1}^n A_{i,j}x_j$   
 3:      $x_i \leftarrow (b_i - S)/A_{i,i}$

4: **Fin**

Algorithme 3.3  $\mathcal{R}_2$ 

1: **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )

2:      $S \leftarrow \sum_{j=i+1}^n A_{i,j}x_j$

3:      $x_i \leftarrow (b_i - S)/A_{i,i}$

4: **Fin**

Algorithme 3.3  $\mathcal{R}_3$ 

1: **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )

2:      $S \leftarrow 0$   
 3:     **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**  
 4:          $S \leftarrow S + A(i, j) * x(j)$   
 5:     **Fin**

6:      $x_i \leftarrow (b_i - S)/A_{i,i}$

7: **Fin**

On obtient alors l'algorithme final

**Algorithme 3.3** Fonction `RSLTriSup` permettant de résoudre le système linéaire triangulaire supérieur inversible

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

**Données :**  $\mathbb{A}$  : matrice triangulaire de  $\mathcal{M}_n(\mathbb{R})$  supérieure inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ .

1: **Fonction**  $\mathbf{x} \leftarrow \text{RSLTriSup}(\mathbb{A}, \mathbf{b})$   
 2:     **Pour**  $i \leftarrow n$  à 1 **faire**(pas de  $-1$ )  
 3:          $S \leftarrow 0$   
 4:         **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**  
 5:              $S \leftarrow S + A(i, j) * x(j)$   
 6:         **Fin**  
 7:          $x(i) \leftarrow (b(i) - S)/A(i, i)$   
 8:     **Fin**  
 9: **Fin**

## 3.1.2 Exercices et résultats préliminaires

## EXERCICE 3.1.2 correction page 253

Soit  $A \in \mathcal{M}_{n,n}(\mathbb{C})$  une matrice et  $(\lambda, \mathbf{u})$  un élément propre de  $A$  avec  $\|\mathbf{u}\|_2 = 1$ .

Q. 1

- En s'aidant de la base canonique  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ , construire une base orthonormée  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  telle que  $\mathbf{x}_1 = \mathbf{u}$ .
- Ecrire la fonction algorithmique `BaseOrtho` permettant de construire la base orthonormée  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  à partir d'un vecteur  $\mathbf{u}$  donné. Les vecteurs  $\mathbf{x}_k$  seront stockés dans une matrice de  $\mathcal{M}_{n,n}(\mathbb{C})$ , le vecteur  $\mathbf{x}_k$  étant en colonne  $k$  de la matrice. On pourra pour cela utiliser les fonctions prédéfinies  $s \leftarrow \text{dot}(\mathbf{u}, \mathbf{v})$  qui retourne le produit scalaire de deux vecteurs,  $s \leftarrow \text{abs}(x)$  qui retourne le module d'un nombre complexe, ...

Notons par  $P$  la matrice de  $\mathcal{M}_{n,n}(\mathbb{C})$  définie par

$$P = \begin{pmatrix} | & & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_n \\ | & & | \end{pmatrix}$$

et par  $B$  la matrice définie par  $B = P^*AP$ .

Q. 2

- Calculer la matrice  $P^*P$ . Que peut-on en conclure?
- Exprimer les coefficients de la matrice  $B$  en fonction de la matrice  $A$  et des vecteurs  $\mathbf{x}_i$ ,  $i \in \llbracket 1, n \rrbracket$ .

$$B = P^*AP.$$

- En déduire que la première colonne de  $B$  est  $(\lambda, 0, \dots, 0)^t$ .

Q. 3

Montrer par récurrence sur l'ordre de la matrice que la matrice  $A$  s'écrit

$$A = UTU^*$$

où  $U$  est une matrice unitaire et  $T$  une matrice triangulaire supérieure.

Q. 4

On dispose des fonctions  $\mathbf{x} \leftarrow \text{ResTriSup}(S, \mathbf{b})$  et  $\mathbf{x} \leftarrow \text{ResTriInf}(L, \mathbf{b})$  permettant de résoudre respectivement le système triangulaire supérieur  $S\mathbf{x} = \mathbf{b}$  et le système triangulaire inférieur  $L\mathbf{x} = \mathbf{b}$ . En supposant  $A$  inversible et la décomposition  $A = UTU^*$  connue, expliquez comment résoudre "simplement" le système linéaire  $A\mathbf{x} = \mathbf{b}$ .

On tire de cet exercice le théorème suivant

**Théorème 3.1.1** (Décomposition de Schur ★★★★★). Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . Il existe une matrice unitaire  $U$  et une matrice triangulaire supérieure  $T$  telles que

$$A = UTU^* \tag{3.4}$$

Preuve. voir Exercice 3.1.2 □

**Théorème 3.1.2** (Réduction de matrices ★★★★★). a. Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . Il existe une matrice unitaire  $U$  telle que  $U^{-1}AU$  soit triangulaire.

b. Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice normale. Il existe une matrice unitaire  $U$  telle que  $U^{-1}AU$  soit diagonale.

c. Soit  $A \in \mathcal{M}_n(\mathbb{R})$  une matrice symétrique. Il existe une matrice orthogonale  $P$  telle que

$\mathbb{P}^{-1}\mathbb{A}\mathbb{P}$  soit *diagonale*.

Preuve. voir [1] Théorème 1.2-1 page 9 □

### EXERCICE 3.1.3 Matrice d'élimination

Soit  $\mathbf{v} \in \mathbb{C}^n$  avec  $v_1 \neq 0$ . On note  $\mathbb{E}^{[\mathbf{v}]} \in \mathcal{M}_n(\mathbb{C})$  la matrice triangulaire inférieure à diagonale unité définie par

$$\mathbb{E}^{[\mathbf{v}]} = \left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ -v_2/v_1 & 1 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -v_n/v_1 & 0 & \dots & 0 & 1 \end{array} \right) \quad (3.1)$$

Q. 1

a. Calculer le déterminant de  $\mathbb{E}^{[\mathbf{v}]}$ .

b. Déterminer l'inverse de  $\mathbb{E}^{[\mathbf{v}]}$ .

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  avec  $A_{1,1} \neq 0$ . On note  $\mathbf{A}_{\cdot,j}$  le  $j$ -ème vecteur colonne de  $\mathbb{A}$  et  $\mathbf{A}_{i,\cdot}$  son  $i$ -ème vecteur ligne. On pose  $\mathbf{A}_1 = \mathbf{A}_{\cdot,1}$ .

Q. 2

a. Calculer  $\tilde{\mathbb{A}} = \mathbb{E}^{[\mathbf{A}_1]}\mathbb{A}$  en fonction des vecteurs lignes de  $\mathbb{A}$ .

b. Montrer que la première colonne de  $\tilde{\mathbb{A}}$  est le vecteur  $(A_{1,1}, 0, \dots, 0)^t$  i.e.

$$\mathbb{E}^{[\mathbf{A}_1]}\mathbb{A}\mathbf{e}_1 = A_{1,1}\mathbf{e}_1 \quad (3.2)$$

où  $\mathbf{e}_1$  est le premier vecteur de la base canonique de  $\mathbb{C}^n$ .

Soit  $m \in \mathbb{N}^*$ . On note  $\mathbb{E}^{[m,\mathbf{v}]} \in \mathcal{M}_{m+n}(\mathbb{C})$  la matrice triangulaire inférieure à diagonale unité définie par

$$\mathbb{E}^{[m,\mathbf{v}]} = \left( \begin{array}{c|c} \mathbb{I}_m & \mathbf{0} \\ \hline \mathbf{0} & \mathbb{E}^{[\mathbf{v}]} \end{array} \right) \quad (3.3)$$

Q. 3

a. Calculer le déterminant de  $\mathbb{E}^{[m,\mathbf{v}]}$ .

b. Déterminer l'inverse de  $\mathbb{E}^{[m,\mathbf{v}]}$  en fonction de l'inverse de  $\mathbb{E}^{[\mathbf{v}]}$ .

Soit  $\mathbb{C}$  la matrice bloc définie par

$$\mathbb{C} = \left( \begin{array}{c|c} \mathbb{C}_{1,1} & \mathbb{C}_{1,2} \\ \hline \mathbf{0} & \mathbb{A} \end{array} \right)$$

où  $\mathbb{C}_{1,1} \in \mathcal{M}_m(\mathbb{C})$  et  $\mathbb{C}_{1,2} \in \mathcal{M}_{m,n}(\mathbb{C})$ .

Q. 4

Déterminer la matrice produit  $\mathbb{E}^{[m,\mathbf{A}_1]}\mathbb{C}$  en fonction des matrices  $\mathbb{C}_{1,1}$ ,  $\mathbb{C}_{1,2}$  et  $\tilde{\mathbb{A}}$ .

### Correction 3.1.3

R. 1

a. La matrice  $\mathbb{E}^{[\mathbf{v}]}$  est triangulaire : son déterminant est donc le produit de ses éléments diagonaux (Proposition B.2.8 de [?]) On a alors  $\det(\mathbb{E}^{[\mathbf{v}]}) = 1$ .

b. Pour calculer son inverse qui existe puisque  $\det(\mathbb{E}^{[\mathbf{v}]}) \neq 0$ , on écrit  $\mathbb{E}^{[\mathbf{v}]}$  sous forme bloc :

$$\mathbb{E}^{[\mathbf{v}]} = \left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \mathbf{e} & & & \\ \hline & & \mathbb{I}_{n-1} & \end{array} \right)$$

avec  $\mathbf{e} = (-v_2/v_1, \dots, -v_n/v_1)^t \in \mathbb{C}^{n-1}$  On note  $\mathbb{X} \in \mathcal{M}_n(\mathbb{C})$  son inverse qui s'écrit avec la même

structure bloc

$$\mathbb{X} = \left( \begin{array}{c|c} a & \mathbf{b}^* \\ \hline \mathbf{c} & \mathbb{D} \end{array} \right)$$

avec  $a \in \mathbb{C}$ ,  $\mathbf{b} \in \mathbb{C}^{n-1}$ ,  $\mathbf{c} \in \mathbb{C}^{n-1}$  et  $\mathbb{D} \in \mathcal{M}_{n-1}(\mathbb{C})$ .

La matrice  $\mathbb{X}$  est donc solution de  $\mathbb{E}^{[\mathbf{v}]} \mathbb{X} = \mathbb{I}$ . Grâce à l'écriture bloc des matrices on en déduit rapidement la matrice  $\mathbb{X}$ . En effet, en utilisant les produits blocs des matrices, on obtient

$$\begin{aligned} \mathbb{E}^{[\mathbf{v}]} \mathbb{X} &= \left( \begin{array}{c|c} 1 & \mathbf{0}_{n-1}^t \\ \hline \mathbf{e} & \mathbb{I}_{n-1} \end{array} \right) \left( \begin{array}{c|c} a & \mathbf{b}^* \\ \hline \mathbf{c} & \mathbb{D} \end{array} \right) = \left( \begin{array}{c|c} 1 \times a & 1 \times \mathbf{b}^* + \mathbf{0}_{n-1}^t \times \mathbb{D} \\ \hline \mathbf{e} \times a + \mathbb{I}_{n-1} \times \mathbf{c} & \mathbf{e} \times \mathbf{b}^* + \mathbb{I}_{n-1} \times \mathbb{D} \end{array} \right) \\ &= \left( \begin{array}{c|c} a & \mathbf{b}^* \\ \hline \mathbf{ae} + \mathbf{c} & \mathbf{eb}^* + \mathbb{D} \end{array} \right) \end{aligned}$$

Comme  $\mathbb{X}$  est l'inverse de  $\mathbb{E}^{[\mathbf{v}]}$ , on a  $\mathbb{E}^{[\mathbf{v}]} \mathbb{X} = \mathbb{I}$  et donc en écriture bloc

$$\left( \begin{array}{c|c} a & \mathbf{b}^* \\ \hline \mathbf{ae} + \mathbf{c} & \mathbf{eb}^* + \mathbb{D} \end{array} \right) = \left( \begin{array}{c|c} 1 & \mathbf{0}_{n-1}^t \\ \hline \mathbf{0}_{n-1} & \mathbb{I}_{n-1} \end{array} \right).$$

Ceci revient à résoudre les 4 équations

$$a = 1, \quad \mathbf{b}^* = \mathbf{0}_{n-1}^t, \quad \mathbf{ae} + \mathbf{c} = \mathbf{0}_{n-1} \quad \text{et} \quad \mathbf{eb}^* + \mathbb{D} = \mathbb{I}_{n-1}$$

qui donnent immédiatement  $a = 1$ ,  $\mathbf{b} = \mathbf{0}_{n-1}$ ,  $\mathbf{c} = -\mathbf{e}$  et  $\mathbb{D} = \mathbb{I}_{n-1}$ . On obtient le résultat suivant

$$\left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline -\mathbf{e} & \mathbb{I}_{n-1} & & \end{array} \right) \left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline \mathbf{e} & \mathbb{I}_{n-1} & & \end{array} \right) = \mathbb{I}_n.$$

**R. 2**

- a. Pour simplifier les notations, on note  $\mathbb{E} = \mathbb{E}^{[\mathbf{A}_1]}$ . Par définition du produit de deux matrices on a

$$\tilde{A}_{i,j} = \sum_{k=1}^n E_{i,k} A_{k,j}, \quad \forall (i,j) \in \llbracket 1, n \rrbracket^2.$$

Quand  $i = 1$ , on a par construction  $E_{1,k} = \delta_{1,k}$  et donc

$$\tilde{A}_{1,j} = A_{1,j}, \quad \forall j \in \llbracket 1, n \rrbracket \iff \tilde{\mathbf{A}}_{1,:} = \mathbf{A}_{1,:}. \quad (\text{R3.2})$$

Pour  $i \geq 2$ , on a  $E_{i,1} = -\frac{v_i}{v_1}$  et  $E_{i,k} = \delta_{i,k}$ ,  $\forall k \in \llbracket 2, n \rrbracket$ . On obtient alors pour tout  $j \in \llbracket 1, n \rrbracket$

$$\tilde{A}_{i,j} = E_{i,1} A_{1,j} + \sum_{k=2}^n E_{i,k} A_{k,j} = -\frac{v_i}{v_1} A_{1,j} + \sum_{k=2}^n \delta_{i,k} A_{k,j} = -\frac{v_i}{v_1} A_{1,j} + A_{i,j}$$

ce qui donne pour tout  $i \in \llbracket 2, n \rrbracket$

$$\tilde{A}_{i,j} = A_{i,j} - \frac{v_i}{v_1} A_{1,j}, \quad \forall j \in \llbracket 1, n \rrbracket \iff \tilde{\mathbf{A}}_{i,:} = -\frac{v_i}{v_1} \mathbf{A}_{1,:} + \mathbf{A}_{i,:}. \quad (\text{R3.3})$$

En conclusion, la matrice  $\tilde{A}$  s'écrit

$$\tilde{A} = \begin{pmatrix} \mathbf{A}_{1,:} \\ \mathbf{A}_{2,:} - (v_2/v_1)\mathbf{A}_{1,:} \\ \vdots \\ \mathbf{A}_{n,:} - (v_n/v_1)\mathbf{A}_{1,:} \end{pmatrix}$$

- b. De (R3.2), on tire  $\tilde{A}_{1,1} = A_{1,1}$ . A partir de (R3.3) on obtient pour tout  $i \in \llbracket 2, n \rrbracket$ ,  $\tilde{A}_{i,1} = A_{i,1} - \frac{v_i}{v_1}A_{1,1}$ . Par construction  $v_j = A_{j,1}$  pour tout  $j \in \llbracket 1, n \rrbracket$ , ce qui donne  $\tilde{A}_{i,1} = 0$ . La première colonne de  $\tilde{A}$  est  $(1, 0, \dots, 0)^t$ .

**R. 3**

- a. La matrice  $\mathbb{E}^{[m, \mathbf{v}]}$  est triangulaire inférieure. Son déterminant est donc le produit de ses éléments diagonaux. Comme cette matrice est à diagonale unité (i.e. tous ses éléments diagonaux valent 1), on obtient  $\det \mathbb{E}^{[m, \mathbf{v}]} = 1$ .

Une autre manière de le démontrer. On peut voir que la matrice  $\mathbb{E}^{[m, \mathbf{v}]}$  est bloc-diagonale. D'après la Proposition B.2.9 de [?], son déterminant est le produit des déterminant des blocs diagonaux :  $\det \mathbb{E}^{[m, \mathbf{v}]} = \det \mathbb{I}_m \times \det \mathbb{E}^{[\mathbf{v}]} = 1$ .

- b. On note  $\mathbb{X}$  l'inverse de la matrice  $\mathbb{E}^{[m, \mathbf{v}]}$ . Cette matrice s'écrit avec la même structure bloc

$$\mathbb{X} = \begin{pmatrix} \mathbb{X}_{1,1} & \mathbb{X}_{1,2} \\ \mathbb{X}_{2,1} & \mathbb{X}_{2,2} \end{pmatrix} \text{ avec } \mathbb{X}_{1,1} \in \mathcal{M}_m(\mathbb{C}) \text{ et } \mathbb{X}_{2,2} \in \mathcal{M}_n(\mathbb{C})$$

On a donc  $\mathbb{X}\mathbb{E}^{[m, \mathbf{v}]} = \mathbb{I}_{m+n}$  c'est à dire en écriture bloc

$$\begin{pmatrix} \mathbb{X}_{1,1} & \mathbb{X}_{1,2} \\ \mathbb{X}_{2,1} & \mathbb{X}_{2,2} \end{pmatrix} \begin{pmatrix} \mathbb{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbb{E}^{[\mathbf{v}]} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_n \end{pmatrix} =$$

On doit donc résoudre les 4 équations suivantes :

$$\mathbb{X}_{1,1}\mathbb{I}_m = \mathbb{I}_m, \quad \mathbb{X}_{1,2}\mathbb{I}_n = \mathbf{0}, \quad \mathbb{X}_{2,1}\mathbb{I}_m = \mathbf{0} \quad \text{et} \quad \mathbb{X}_{2,2}\mathbb{E}^{[\mathbf{v}]} = \mathbb{I}_n.$$

Comme la matrice  $\mathbb{E}^{[\mathbf{v}]}$  est inversible, on obtient

$$\mathbb{X} = \begin{pmatrix} \mathbb{I}_m & \mathbf{0} \\ \mathbf{0} & (\mathbb{E}^{[\mathbf{v}]})^{-1} \end{pmatrix}$$

**NOTE**

Plus rapidement, comme la matrice  $\mathbb{E}^{[m, \mathbf{v}]}$  est bloc-diagonale, d'après la Proposition B.2.10 de [?], on déduit directement le résultat.

**R. 4**

Le produit  $\mathbb{E}^{[m, \mathbf{v}]} \mathbb{C}$  peut s'effectuer par bloc car les blocs sont de dimensions compatibles et on a

$$\begin{aligned} \mathbb{E}^{[m, \mathbf{v}]} \mathbb{C} &= \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \mathbb{E} \end{pmatrix} \begin{pmatrix} \mathbb{C}_{1,1} & \mathbb{C}_{1,2} \\ \mathbf{0}_{n,m} & \mathbb{A} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_m \mathbb{C}_{1,1} + \mathbf{0}_{m,n} \mathbf{0}_{n,m} & \mathbb{I}_m \mathbb{C}_{1,2} + \mathbf{0}_{m,n} \mathbb{A} \\ \mathbf{0}_{n,m} \mathbb{C}_{1,1} + \mathbb{E} \mathbf{0}_{n,m} & \mathbf{0}_{n,m} \mathbb{C}_{1,2} + \mathbb{E} \mathbb{A} \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{C}_{1,1} & \mathbb{C}_{1,2} \\ \mathbf{0}_{n,m} & \mathbb{E} \mathbb{A} \end{pmatrix} = \begin{pmatrix} \mathbb{C}_{1,1} & \mathbb{C}_{1,2} \\ \mathbf{0}_{n,m} & \tilde{\mathbb{A}} \end{pmatrix} \end{aligned}$$

~~~~~ Fin correction ~~~~~

On tire de cet exercice le lemme suivant

**Lemme 3.1.1.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  avec  $A_{1,1} \neq 0$ . Il existe une matrice  $\mathbb{E} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure à diagonale unité telle que

$$\mathbb{E}\mathbf{a}_1 = A_{1,1}\mathbf{e}_1 \quad (3.5)$$

où  $\mathbf{e}_1$  est le premier vecteur de la base canonique de  $\mathbb{C}^n$ .

### EXERCICE 3.1.4 Matrice de permutation

Soit  $(i, j) \in \llbracket 1, n \rrbracket^2$ ,  $i \neq j$ , on note  $\mathbb{P}_n^{[i,j]} \in \mathcal{M}_n(\mathbb{R})$  la matrice identité dont on a permuté les lignes  $i$  et  $j$ .

**Q. 1** Représenter cette matrice et la définir proprement.

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ . On note  $\mathbf{A}_{r,:}$  le  $r$ -ème vecteur ligne de  $\mathbb{A}$  et  $\mathbf{A}_{:,s}$  le  $s$ -ème vecteur colonne de  $\mathbb{A}$ .

**Q. 2**

- Déterminer les lignes de la matrice  $\mathbb{D} = \mathbb{P}_n^{[i,j]}\mathbb{A}$  en fonction des vecteurs lignes de  $\mathbb{A}$ .
- Déterminer les colonnes de la matrice  $\mathbb{E} = \mathbb{A}\mathbb{P}_n^{[i,j]}$  en fonction des vecteurs colonnes de  $\mathbb{A}$ .

**Q. 3**

- Calculer le déterminant de  $\mathbb{P}_n^{[i,j]}$ .
- Déterminer l'inverse de  $\mathbb{P}_n^{[i,j]}$ .

### Correction 3.1.4

**R. 1**

On note, dans toute la correction,  $\mathbb{P} = \mathbb{P}_n^{[i,j]}$ .

On peut définir cette matrice par ligne,

$$\begin{cases} \forall r \in \llbracket 1, n \rrbracket \setminus \{i, j\}, & P_{r,s} = \delta_{r,s}, & \forall s \in \llbracket 1, n \rrbracket, \\ & P_{i,s} = \delta_{j,s}, & \forall s \in \llbracket 1, n \rrbracket, \\ & P_{j,s} = \delta_{i,s}, & \forall s \in \llbracket 1, n \rrbracket. \end{cases}$$

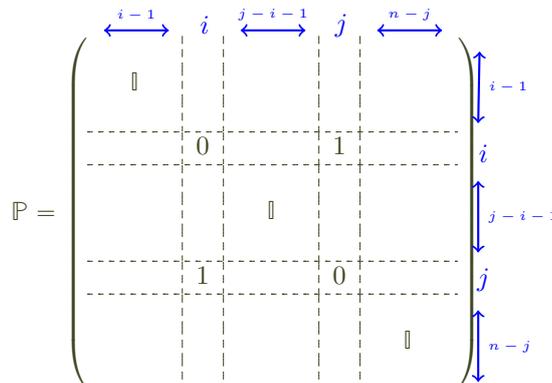
ou par colonne

$$\begin{cases} \forall s \in \llbracket 1, n \rrbracket \setminus \{i, j\}, & P_{r,s} = \delta_{r,s}, & \forall r \in \llbracket 1, n \rrbracket, \\ & P_{r,i} = \delta_{r,j}, & \forall r \in \llbracket 1, n \rrbracket, \\ & P_{r,j} = \delta_{r,i}, & \forall r \in \llbracket 1, n \rrbracket. \end{cases}$$

 Ne pas utiliser les indices  $i$  et  $j$  qui sont déjà fixés dans la définition de la matrice  $\mathbb{P} = \mathbb{P}_n^{[i,j]}$ .

On peut noter que la matrice  $\mathbb{P}$  est symétrique. Pour la représentation, on suppose  $i < j$ . On effectue une représentation bloc  $5 \times 5$  avec des blocs diagonaux carrés sachant que tous les blocs non décrits sont nuls:

$$\mathbb{P} = \begin{pmatrix} \boxed{\phantom{0}} & & & & \\ & 0 & & 1 & \\ & & \boxed{\phantom{0}} & & \\ & 1 & & 0 & \\ & & & & \boxed{\phantom{0}} \end{pmatrix}$$



R. 2

a. On note  $\mathbb{D} = \mathbb{P}\mathbb{A}$ . Par définition du produit matriciel on a

$$D_{r,s} = \sum_{k=1}^n P_{r,k} A_{k,s}.$$

On obtient,  $\forall s \in \llbracket 1, n \rrbracket$ ,

$$\begin{cases} D_{r,s} = \sum_{k=1}^n \delta_{r,k} A_{k,s} = A_{r,s}, & \forall r \in \llbracket 1, n \rrbracket \setminus \{i, j\}, \\ D_{i,s} = \sum_{k=1}^n \delta_{j,k} A_{k,s} = A_{j,s}, \\ D_{j,s} = \sum_{k=1}^n \delta_{i,k} A_{k,s} = A_{i,s}. \end{cases}$$

ce qui donne

$$\begin{cases} \mathbf{D}_{r,:} = \mathbf{A}_{r,:}, & \forall r \in \llbracket 1, n \rrbracket \setminus \{i, j\}, \\ \mathbf{D}_{i,:} = \mathbf{A}_{j,:}, \\ \mathbf{D}_{j,:} = \mathbf{A}_{i,:}. \end{cases}$$

**NOTE**

La notation  $\mathbf{D}_{i,:}$  correspond au vecteur ligne  $(D_{i,1}, \dots, D_{i,n})$  et  $\mathbf{D}_{:,j}$  correspond au vecteur colonne  $\begin{pmatrix} D_{1,j} \\ \vdots \\ D_{n,j} \end{pmatrix}$

b. On note  $\mathbb{E} = \mathbb{A}\mathbb{P}$ . Par définition du produit matriciel et par symétrie de  $\mathbb{P}$  on a

$$E_{r,s} = \sum_{k=1}^n A_{r,k} P_{k,s} = \sum_{k=1}^n A_{r,k} P_{s,k}.$$

 Ne pas utiliser les indices  $i$  et  $j$  qui sont déjà fixés dans la définition de la matrice  $\mathbb{P} = \mathbb{P}_n^{[i,j]}$ .

On obtient en raisonnant par colonne,  $\forall r \in \llbracket 1, n \rrbracket$ ,

$$\begin{cases} E_{r,s} = \sum_{k=1}^n A_{r,k} \delta_{s,k} = A_{r,s}, & \forall s \in \llbracket 1, n \rrbracket \setminus \{i, j\}, \\ E_{r,i} = \sum_{k=1}^n A_{r,k} \delta_{j,k} = A_{r,j}, \\ E_{r,j} = \sum_{k=1}^n A_{r,k} \delta_{i,k} = A_{r,i}. \end{cases}$$

ce qui donne

$$\begin{cases} \mathbf{E}_{:,s} = \mathbf{A}_{:,s}, & \forall s \in \llbracket 1, n \rrbracket \setminus \{i, j\}, \\ \mathbf{E}_{:,i} = \mathbf{A}_{:,j}, \\ \mathbf{E}_{:,j} = \mathbf{A}_{:,i}. \end{cases}$$

R. 3

a.  $\det(\mathbb{P}) = -1$ , si  $i \neq j$  et  $\det(\mathbb{P}) = 1$  sinon.

b. Immédiat par calcul direct on a  $\mathbb{P}\mathbb{P} = \mathbb{I}$  et donc la matrice  $\mathbb{P}$  est inversible et  $\mathbb{P}^{-1} = \mathbb{P}$ .

~~~~~ Fin correction ~~~~~

On tire de cet exercice le lemme suivant

**Lemme 3.1.2.** Soit  $(i, j) \in \llbracket 1, n \rrbracket^2$ . On note  $\mathbb{P}_n^{[i,j]} \in \mathcal{M}_n(\mathbb{R})$  la matrice identité dont on a permuté les lignes  $i$  et  $j$ . Alors la matrice  $\mathbb{P}_n^{[i,j]}$  est **symétrique et orthogonale**. Pour toute matrice  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ ,

- a. la matrice  $\mathbb{P}_n^{[i,j]} \mathbb{A}$  est la matrice  $\mathbb{A}$  dont on a permuté les **lignes**  $i$  et  $j$ ,
- b. la matrice  $\mathbb{A} \mathbb{P}_n^{[i,j]}$  est la matrice  $\mathbb{A}$  dont on a permuté les **colonnes**  $i$  et  $j$ ,

### 3.1.3 Méthode de Gauss-Jordan, écriture matricielle

Soient  $\mathbb{A} \in \mathcal{M}_{\mathbb{K}}()$  une matrice inversible et  $\mathbf{b} \in \mathbb{K}^n$ .

On va tout d'abord rappeler (très) brièvement l'**algorithme d'élimination** ou **algorithme de Gauss-Jordan** permettant de transformer le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  en un système linéaire équivalent dont la matrice est triangulaire supérieure. Ce dernier système se résout par la méthode de remontée.

Ensuite, on va réécrire cet algorithme sous forme algébrique pour obtenir le théorème ...



Cette méthode doit son nom aux mathématiciens Carl Friedrich Gauss (1777-1855, mathématicien, astronome et physicien allemand) et Wilhelm Jordan (1842-1899, mathématicien et géodésien allemand) mais elle est connue des Chinois depuis au moins le I<sup>er</sup> siècle de notre ère. Elle est référencée dans l'important livre chinois *Jiuzhang suanshu* ou *Les Neuf Chapitres sur l'art mathématique*, dont elle constitue le huitième chapitre, sous le titre « Fang cheng » (la disposition rectangulaire). La méthode est présentée au moyen de dix-huit exercices. Dans son commentaire daté de 263, Liu Hui en attribue la paternité à Chang Ts'ang, chancelier de l'empereur de Chine au II<sup>e</sup> siècle avant notre ère.

#### Algorithme de Gauss-Jordan usuel

Pour la résolution du système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  l'algorithme de Gauss-Jordan produit la forme échelonnée (réduite) d'une matrice à l'aide d'opérations élémentaires sur les lignes du système. Trois types d'opérations élémentaires sont utilisées:

- Permutation de deux lignes ;
- Multiplication d'une ligne par un scalaire non nul ;
- Ajout du multiple d'une ligne à une autre ligne.

A l'aide de ces opérations élémentaires cet algorithme permet donc de transformer le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  en le système équivalent  $\mathbb{U}\mathbf{x} = \mathbf{f}$  où  $\mathbb{U}$  est triangulaire supérieure. En fait, l'algorithme va transformer la matrice  $\mathbb{A}$  et le second membre  $\mathbf{b}$  pour aboutir à un système dont la matrice est triangulaire supérieure.

---

**Algorithm 3.4** Algorithme de Gauss-Jordan formel pour la résolution de  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

- 1: **Pour**  $j \leftarrow 1$  **à**  $n - 1$  **faire**
  - 2:   Rechercher l'indice  $k$  de la ligne du pivot (sur la colonne  $j$ ,  $k \in \llbracket j, n \rrbracket$ )
  - 3:   Permuter les lignes  $j$  ( $\mathcal{L}_j$ ) et  $k$  ( $\mathcal{L}_k$ ) du système si besoin.
  - 4:   **Pour**  $i \leftarrow j + 1$  **à**  $n$  **faire**
  - 5:     Eliminer en effectuant  $\mathcal{L}_i \leftarrow \mathcal{L}_i - \frac{A_{i,j}}{A_{j,j}} \mathcal{L}_j$
  - 6:   **Fin**
  - 7: **Fin**
  - 8: Résoudre le système triangulaire supérieur par la méthode de la remontée.
-

On va maintenant voir comment écrire cet algorithme de manière plus détaillée. Pour conserver sa lisibilité, on choisit pour chaque ligne un peu délicate de créer et d'utiliser une fonction dédiée à cette tâche.

---

**Algorithm 3.5** Algorithme de Gauss-Jordan avec fonctions pour la résolution de  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{K}^n$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{RSLGauss}(\mathbb{A}, \mathbf{b})$ 
2:   Pour  $j \leftarrow 1$  à  $n - 1$  faire
3:      $k \leftarrow \text{ChercheIndPivot}(\mathbb{A}, j)$  ▷ ChercheIndPivot à écrire
4:      $[\mathbb{A}, \mathbf{b}] \leftarrow \text{PermLignesSys}(\mathbb{A}, \mathbf{b}, j, k)$  ▷ PermLignesSys à écrire
5:     Pour  $i \leftarrow j + 1$  à  $n$  faire
6:        $[\mathbb{A}, \mathbf{b}] \leftarrow \text{CombLignesSys}(\mathbb{A}, \mathbf{b}, j, i, -A(i, j)/A(j, j))$  ▷ CombLignesSys à écrire
7:     Fin
8:   Fin
9:    $\mathbf{x} \leftarrow \text{RSLTriSup}(\mathbb{A}, \mathbf{b})$  ▷ RSLTriSup déjà écrite
10: Fin

```

---

Bien évidemment, il reste à décrire et écrire les différentes fonctions utilisées dans cette fonction :

**Fonction**  $k \leftarrow \text{ChercheIndPivot}(\mathbb{A}, j)$  : recherche  $k \in \llbracket j, n \rrbracket$  tel que  $\forall l \in \llbracket j, n \rrbracket, |A_{l,j}| \leq |A_{k,j}|$ .

**Fonction**  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{PermLignesSys}(\mathbb{A}, \mathbf{b}, i, k)$  : permute les lignes  $i$  et  $k$  de la matrice  $\mathbb{A}$  ainsi que celles du vecteur  $\mathbf{b}$ .

**Fonction**  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{CombLignesSys}(\mathbb{A}, \mathbf{b}, j, i, \alpha)$  : remplace la ligne  $i$  de la matrice  $\mathbb{A}$  par la combinaison linéaire  $\mathcal{L}_i \leftarrow \mathcal{L}_i + \alpha \mathcal{L}_j$ . De même on remplace la ligne  $i$  de  $\mathbf{b}$  par  $b_i + \alpha b_j$ .

Ces trois fonctions sont simples à écrire et sont données en Algorithmes 3.6, 3.7 et 3.8.

---

**Algorithm 3.6** Recherche d'un pivot pour l'algorithme de Gauss-Jordan.

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .

$j$  : entier,  $1 \leq j \leq n$ .

**Résultat :**  $k$  : entier, indice ligne pivot

```

1: Fonction  $k \leftarrow \text{ChercheIndPivot}(\mathbb{A}, j)$ 
2:    $k \leftarrow j$ , pivot  $\leftarrow |\mathbb{A}(j, j)|$ 
3:   Pour  $i \leftarrow j + 1$  à  $n$  faire
4:     Si  $|\mathbb{A}(i, j)| >$  pivot alors
5:        $k \leftarrow i$ , pivot  $\leftarrow |\mathbb{A}(i, j)|$ 
6:   Fin
7: Fin
8: Fin

```

---



---

**Algorithm 3.7** Permutte deux lignes d'une matrice et d'un vecteur.

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

$j, k$  : entiers,  $1 \leq j, k \leq n$ .

**Résultat :**  $\mathbb{A}$  et  $\mathbf{b}$  modifiés.

```

1: Fonction  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{PermLignesSys}(\mathbb{A}, \mathbf{b}, j, k)$ 
2:   Pour  $l \leftarrow 1$  à  $n$  faire
3:      $t \leftarrow \mathbb{A}(j, l)$ ,  $\mathbb{A}(j, l) \leftarrow \mathbb{A}(k, l)$ ,  $\mathbb{A}(k, l) \leftarrow t$ 
4:   Fin
5:    $t \leftarrow \mathbf{b}(j)$ ,  $\mathbf{b}(j) \leftarrow \mathbf{b}(k)$ ,  $\mathbf{b}(k) \leftarrow t$ 
6: Fin

```

---



---

**Algorithm 3.8** Combinaison linéaire  $\mathcal{L}_i \leftarrow \mathcal{L}_i + \alpha \mathcal{L}_j$  appliqué à une matrice et à un vecteur.

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

$j, i$  : entiers,  $1 \leq j, i \leq n$ .

alpha : scalaire de  $\mathbb{K}$

**Résultat :**  $\mathbb{A}$  et  $\mathbf{b}$  modifiés.

```

1: Fonction  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{CombLignesSys}(\mathbb{A}, \mathbf{b}, j, i, \alpha)$ 
2:   Pour  $k \leftarrow 1$  à  $n$  faire
3:      $\mathbb{A}(i, k) \leftarrow \mathbb{A}(i, k) + \alpha * \mathbb{A}(j, k)$ 
4:   Fin
5:    $\mathbf{b}(i) \leftarrow \mathbf{b}(i) + \alpha \mathbf{b}(j)$ 
6: Fin

```

---

## Ecriture algébrique

Sous forme d'exercice :

### EXERCICE 3.1.5

Soit  $A \in \mathcal{M}_n(\mathbb{C})$  inversible.

Q. 1

Montrer qu'il existe une matrice  $G \in \mathcal{M}_n(\mathbb{C})$  telle que  $|\det(G)| = 1$  et  $GA\mathbf{e}_1 = \alpha\mathbf{e}_1$  avec  $\alpha \neq 0$  et  $\mathbf{e}_1$  premier vecteur de la base canonique de  $\mathbb{C}^n$ .

Q. 2

a. Montrer par récurrence sur l'ordre des matrices que pour toute matrice  $A_n \in \mathcal{M}_n(\mathbb{C})$  inversible, il existe une matrice  $S_n \in \mathcal{M}_n(\mathbb{C})$  telle que  $|\det S_n| = 1$  et  $S_n A_n = U_n$  avec  $U_n$  matrice triangulaire supérieure inversible.

b. Soit  $\mathbf{b} \in \mathbb{C}^n$ . En supposant connue la décomposition précédente  $S_n A_n = U_n$ , expliquer comment résoudre le système  $A_n \mathbf{x} = \mathbf{b}$ .

Q. 3

Que peut-on dire si  $A$  est non inversible?

**Indication :** utiliser les résultats des exercices 3.1.3 et 3.1.4.

### Correction 3.1.5

R. 1

D'après le Lemme 3.1.1, si  $A_{1,1} \neq 0$ , le résultat est immédiat.

Dans l'énoncé rien ne vient corroborer cette hypothèse. Toutefois, comme la matrice  $A$  est inversible, il existe au moins un  $p \in \llbracket 1, n \rrbracket$  tel que  $A_{p,1} \neq 0$ . On peut même choisir le premier indice  $p$  tel que  $|A_{p,1}| = \max_{i \in \llbracket 1, n \rrbracket} |A_{i,1}| > 0$  (pivot de l'algorithme de Gauss-Jordan). On note  $P = P_n^{[1,p]}$  la matrice de permutation des lignes 1 et  $p$  (voir Lemme 3.1.1).

$$|\det P| = 1 \quad \text{et} \quad P^{-1} = P.$$

Par construction  $(PA)_{1,1} = A_{p,1} \neq 0$ , et on peut alors appliquer le Lemme 3.1.1 à la matrice  $(PA)$  pour obtenir l'existence d'une matrice  $E \in \mathcal{M}_n(\mathbb{C})$  vérifiant  $\det E = 1$  et telle que

$$E(PA)\mathbf{e}_1 = A_{p,1}\mathbf{e}_1.$$

En posant  $G = EP$  et  $\alpha = A_{p,1}$ , on obtient bien  $GA\mathbf{e}_1 = \alpha\mathbf{e}_1$ . De plus, on a

$$|\det G| = |\det(EP)| = |\det E \times \det P| = 1.$$

**Remarque.** La matrice  $G$  étant inversible, on a

$$A\mathbf{x} = \mathbf{b} \iff GA\mathbf{x} = G\mathbf{b}$$

ce qui correspond à la première permutation/élimination de l'algorithme de Gauss-Jordan.

R. 2

a. On veut démontrer, par récurrence sur  $n \geq 2$ , la propriété suivante

$(\mathcal{P}_n)$

$\forall A_n \in \mathcal{M}_n(\mathbb{C})$  inversible,  $\exists S_n \in \mathcal{M}_n(\mathbb{C})$ ,  $|\det S_n| = 1$ , tel que la matrice  $U_n \stackrel{\text{def}}{=} S_n A_n$  soit une triangulaire supérieure inversible.

**Initialisation :** Pour  $n = 2$ . Soit  $A_2 \in \mathcal{M}_2(\mathbb{C})$  inversible. En utilisant la question précédente il existe  $G_2 \in \mathcal{M}_2(\mathbb{C})$  telle que  $|\det G_2| = 1$  et  $G_2 A_2 \mathbf{e}_1 = \alpha \mathbf{e}_1$  avec  $\alpha \neq 0$  et  $\mathbf{e}_1$  premier vecteur de la base canonique de  $\mathbb{C}^2$ . On note  $U_2 = G_2 A_2$ . Cette matrice s'écrit donc sous la forme

$$U_2 = \begin{pmatrix} \alpha & \bullet \\ 0 & \bullet \end{pmatrix}$$

et elle est triangulaire supérieure. Les matrices  $\mathbb{G}_2$  et  $\mathbb{A}_2$  étant inversible, leur produit  $\mathbb{U}_2$  l'est aussi. La proposition  $(\mathcal{P}_2)$  est donc vérifiée avec  $\mathbb{S}_2 = \mathbb{G}_2$ .

**Hérédité :** Soit  $n \geq 3$ . On suppose que  $(\mathcal{P}_{n-1})$  est vraie. Montrons que  $(\mathcal{P}_n)$  est vérifiée.

Soit  $\mathbb{A}_n \in \mathcal{M}_n(\mathbb{C})$  inversible. En utilisant la question précédente il existe  $\mathbb{G}_n \in \mathcal{M}_n(\mathbb{C})$  telle que  $|\det \mathbb{G}_n| = 1$  et  $\mathbb{G}_n \mathbb{A}_n \mathbf{e}_1 = \alpha_n \mathbf{e}_1$  avec  $\alpha_n \neq 0$  et  $\mathbf{e}_1$  premier vecteur de la base canonique de  $\mathbb{C}^n$ . On note  $\mathbb{V}_n = \mathbb{G}_n \mathbb{A}_n$ . Cette matrice s'écrit donc sous la forme

$$\mathbb{V}_n = \left( \begin{array}{c|ccc} \alpha_n & \bullet & \dots & \bullet \\ \hline 0 & \bullet & \dots & \bullet \\ \vdots & \vdots & & \vdots \\ 0 & \bullet & \dots & \bullet \end{array} \right) \stackrel{\text{def}}{=} \left( \begin{array}{c|c} \alpha_n & \mathbf{c}_{n-1}^* \\ \hline 0 & \mathbb{B}_{n-1} \\ \vdots & \\ 0 & \end{array} \right)$$

où  $\mathbf{c}_{n-1} \in \mathbb{C}^{n-1}$  et  $\mathbb{B}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$ . Comme  $\mathbb{G}_n$  et  $\mathbb{A}_n$  sont inversibles,  $\mathbb{V}_n$  l'est aussi. On en déduit donc que  $\mathbb{B}_{n-1}$  est inversible car  $0 \neq \det \mathbb{V}_n = \alpha_n \times \det \mathbb{B}_{n-1}$  et  $\alpha_n \neq 0$ .

On peut donc utiliser la propriété  $(\mathcal{P}_{n-1})$  (hyp. de récurrence) sur la matrice  $\mathbb{B}_{n-1}$  : il existe donc  $\mathbb{S}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$ , avec  $|\det \mathbb{S}_{n-1}| = 1$ , tel que la matrice  $\mathbb{U}_{n-1} = \mathbb{S}_{n-1} \mathbb{B}_{n-1}$  soit une triangulaire supérieure inversible.

Soit  $\mathbb{Q}_n \in \mathcal{M}_n(\mathbb{C})$  la matrice définie par

$$\mathbb{Q}_n = \left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & \mathbb{S}_{n-1} & \\ 0 & & & \end{array} \right)$$

On a alors

$$\begin{aligned} \mathbb{Q}_n \mathbb{G}_n \mathbb{A}_n &= \mathbb{Q}_n \mathbb{V}_n = \left( \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & & & \\ \vdots & & \mathbb{S}_{n-1} & \\ 0 & & & \end{array} \right) \left( \begin{array}{c|c} \alpha_n & \mathbf{c}_{n-1}^* \\ \hline 0 & \mathbb{B}_{n-1} \\ \vdots & \\ 0 & \end{array} \right) \\ &= \left( \begin{array}{c|c} \alpha_n & \mathbf{c}_{n-1}^* \\ \hline 0 & \mathbb{S}_{n-1} \mathbb{B}_{n-1} \\ \vdots & \\ 0 & \end{array} \right) = \left( \begin{array}{c|c} \alpha_n & \mathbf{c}_{n-1}^* \\ \hline 0 & \mathbb{U}_{n-1} \\ \vdots & \\ 0 & \end{array} \right) \stackrel{\text{def}}{=} \mathbb{U}_n \end{aligned}$$

La matrice  $\mathbb{U}_n$  est triangulaire supérieure inversible car  $\mathbb{U}_{n-1}$  l'est aussi et  $\alpha_n \neq 0$ .

On pose  $\mathbb{S}_n = \mathbb{Q}_n \mathbb{G}_n$ . On a donc

$$\mathbb{S}_n \mathbb{A}_n = \mathbb{U}_n.$$

De plus, comme on a  $\det \mathbb{S}_n = \det \mathbb{Q}_n \times \det \mathbb{G}_n$ , et  $\det \mathbb{Q}_n = \det \mathbb{S}_{n-1}$ , on obtient, en utilisant  $|\det \mathbb{G}_n| = 1$  et l'hypothèse de récurrence  $|\det \mathbb{S}_{n-1}| = 1$ , que

$$|\det \mathbb{S}_n| = 1.$$

Ceci prouve la véracité de la proposition  $(\mathcal{P}_n)$ .

- b. Comme  $\mathbb{S}_n$  est inversible, on a en multipliant à gauche le système par  $\mathbb{S}_n$

$$\mathbb{A}_n \mathbf{x} = \mathbf{b} \iff \mathbb{S}_n \mathbb{A}_n \mathbf{x} = \mathbb{S}_n \mathbf{b} \iff \mathbb{U}_n \mathbf{x} = \mathbb{S}_n \mathbf{b}$$

Pour déterminer le vecteur  $\mathbf{x}$ , on peut alors résoudre le dernier système par l'algorithme de remontée.

**R. 3**

Si  $\mathbb{A}$  est non inversible, alors dans la première question nous ne sommes pas assurés d'avoir  $\alpha \neq 0$ . Cependant l'existence de la matrice  $\mathbb{G}$  reste avérée.

Pour la deuxième question, le seul changement vient du fait que la matrice  $U_n$  n'est plus inversible.

~~~~~ Fin correction ~~~~~  
On a donc démontré le théorème suivant

**Théorème 3.1.3.** *Soit  $A$  une matrice carrée, inversible ou non. Il existe (au moins) une matrice inversible  $G$  telle que  $GA$  soit triangulaire supérieure.*

*Preuve.* voir Exercice 3.1.5, page 71. □

### 3.1.4 Factorisation LU

Avant de citer le théorème principal, on va faire un "petit" exercice...

#### EXERCICE 3.1.6 ★★★★★

Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice dont les sous-matrices principales d'ordre  $i$ , notées  $\Delta_i$ ,  $i \in \llbracket 1, n \rrbracket$  (voir Définition B.2.26, page 225) sont inversibles.

Montrer qu'il existe des matrices  $E^{[k]} \in \mathcal{M}_n(\mathbb{C})$ ,  $k \in \llbracket 1, n-1 \rrbracket$ , triangulaires inférieures à diagonale unité telles que la matrice  $U$  définie par

$$U = E^{[n-1]} \dots E^{[1]} A$$

soit triangulaire supérieure avec  $U_{i,i} = \det \Delta_i / (U_{1,1} \times \dots \times U_{i-1,i-1})$ ,  $\forall i \in \llbracket 1, n \rrbracket$ .

**Correction 3.1.6** On note  $A^{[0]} = A$ . On va démontrer par récurrence finie sur  $k \in \llbracket 1, n-1 \rrbracket$ , qu'il existe une matrice  $E^{[k]} \in \mathcal{M}_n(\mathbb{C})$ , triangulaire inférieure à diagonale unité, telle que la matrice  $A^{[k]}$  définie itérativement par

$$A^{[k]} = E^{[k]} A^{[k-1]}$$

s'écrit sous la forme bloc

$$A^{[k]} = \left( \begin{array}{cccc|cccc} \alpha_1 & \bullet & \dots & \bullet & \bullet & \dots & \bullet & \\ 0 & \ddots & \ddots & \vdots & \vdots & & \vdots & \\ \vdots & \ddots & \ddots & \bullet & \vdots & & \vdots & \\ 0 & \dots & 0 & \alpha_k & \bullet & \dots & \bullet & \\ \hline 0 & \dots & \dots & 0 & \bullet & \dots & \bullet & \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots & \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots & \\ 0 & \dots & \dots & 0 & \bullet & \dots & \bullet & \end{array} \right) \quad (3.6)$$

avec  $\alpha_1 = A_{1,1}$  et  $\forall i \in \llbracket 2, k \rrbracket$ ,  $\alpha_i = \det \Delta_i / (\alpha_1 \times \dots \times \alpha_{i-1})$ .

**Initialisation** ( $k=1$ ): On a  $A_{1,1} \neq 0$  car  $\Delta_1 = A_{1,1}$  et  $\det \Delta_1 \neq 0$ . D'après le Lemme 3.1.1, il existe une matrice  $E^{[1]} \in \mathcal{M}_n(\mathbb{C})$ , triangulaire inférieure à diagonale unité, telle que  $E^{[1]} A e_1 = A_{1,1} e_1$  où  $e_1$  est le premier vecteur de la base canonique de  $\mathbb{C}^n$ . On a alors

$$A^{[1]} = E^{[1]} A = \left( \begin{array}{c|cccc} \alpha_1 & \bullet & \dots & \bullet & \\ \hline 0 & \bullet & \dots & \bullet & \\ \vdots & \vdots & & \vdots & \\ 0 & \bullet & \dots & \bullet & \end{array} \right)$$

avec  $\alpha_1 = A_{1,1} = \det \Delta_1$ .

**Hérédité** ( $k < n-1$ ): Supposons construite la matrice  $A^{[k]}$ . Il existe donc  $k$  matrices,  $E^{[1]}, \dots, E^{[k]}$ , triangulaires inférieures à diagonale unité telles que

$$A^{[k]} = E^{[k]} \dots E^{[1]} A.$$

- On va montrer que  $\alpha_{k+1} \stackrel{\text{def}}{=} A_{k+1,k+1}^{[k]} \neq 0$ . Pour cela, on réécrit la matrice  $\mathbb{A}^{[k]}$  sous forme bloc, avec comme premier bloc diagonale le bloc de dimension  $k+1$  :

$$\mathbb{A}^{[k]} = \begin{pmatrix} \overbrace{\alpha_1 \quad \bullet \quad \cdots \quad \bullet \quad \bullet}^{k+1} & \bullet & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots & \bullet \\ \vdots & \ddots & \ddots & \bullet & \bullet \\ 0 & \cdots & 0 & \alpha_k & \bullet \\ 0 & \cdots & \cdots & 0 & \alpha_{k+1} \\ \hline 0 & \cdots & \cdots & 0 & \bullet \\ \vdots & & & \vdots & \bullet \\ 0 & \cdots & \cdots & 0 & \bullet \end{pmatrix}$$

La matrice  $\mathbb{G}^{[k]} \stackrel{\text{def}}{=} \mathbb{E}^{[k]} \cdots \mathbb{E}^{[1]}$  est triangulaire inférieure à diagonale unité car produit de matrices triangulaires inférieures à diagonale unité (voir Exercice ??, page ??). Le produit de  $\mathbb{G}^{[k]} \mathbb{A}$  s'écrit alors sous forme bloc

$$\mathbb{G}^{[k]} \mathbb{A} = \begin{pmatrix} \overbrace{1 \quad 0 \quad \cdots \quad 0}^{k+1} & 0 & \cdots & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots & \bullet \\ \vdots & \ddots & \ddots & 0 & \bullet \\ \bullet & \cdots & \bullet & 1 & 0 \\ \bullet & \cdots & \cdots & \bullet & 1 \\ \hline \vdots & \ddots & \ddots & \vdots & \bullet \\ \vdots & \ddots & \ddots & \vdots & \bullet \\ \vdots & \ddots & \ddots & \vdots & 0 \\ \bullet & \cdots & \cdots & \bullet & 1 \end{pmatrix} \begin{pmatrix} \overbrace{\Delta_{k+1}}^{k+1} & \bullet & \cdots & \bullet \\ \vdots & \vdots & \vdots & \vdots \\ \hline \bullet & \cdots & \bullet & \bullet \\ \vdots & & \vdots & \vdots \\ \bullet & \cdots & \bullet & \bullet \end{pmatrix}$$

Comme  $\mathbb{A}^{[k]} = \mathbb{G}^{[k]} \mathbb{A}$ , en utilisant les règles de multiplication par blocs des matrices on obtient

$$\begin{pmatrix} \alpha_1 & \bullet & \cdots & \bullet & \bullet \\ 0 & \ddots & \ddots & \vdots & \bullet \\ \vdots & \ddots & \ddots & \bullet & \bullet \\ 0 & \cdots & 0 & \alpha_k & \bullet \\ 0 & \cdots & \cdots & 0 & \alpha_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \bullet & \cdots & \bullet & 1 \end{pmatrix} \begin{pmatrix} \Delta_{k+1} \\ \vdots \\ \vdots \end{pmatrix}$$

En prenant le déterminant de cette dernière équation, et en utilisant le fait que le déterminant d'une matrice triangulaire est le produit de ses coefficients diagonaux, on obtient

$$\prod_{i=1}^{k+1} \alpha_i = \det \Delta_{k+1}.$$

Par hypothèse  $\Delta_{k+1}$  inversible, ce qui entraîne  $\det \Delta_{k+1} \neq 0$  et donc  $\alpha_i \neq 0, \forall i \in \llbracket 1, k+1 \rrbracket$ . On a donc

$$\alpha_{k+1} = \frac{\det \Delta_{k+1}}{\prod_{i=1}^k \alpha_i} \neq 0.$$

- Montrons l'existence d'une matrice triangulaire inférieure à diagonale unité permettant d'éliminer les termes sous diagonaux de la colonne  $k+1$  de  $\mathbb{A}^{[k]}$ .

Revenons à l'écriture bloc de premier bloc diagonal de dimension  $k$ . On a

$$\mathbb{A}^{[k]} = \left( \begin{array}{cccc|cccc} \alpha_1 & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \bullet & \vdots & & & \vdots \\ 0 & \cdots & 0 & \alpha_k & \bullet & \cdots & \cdots & \bullet \\ \hline 0 & \cdots & \cdots & 0 & \alpha_{k+1} & \bullet & \cdots & \bullet \\ \vdots & \ddots & \ddots & \vdots & \bullet & \cdots & \cdots & \bullet \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 & \bullet & \cdots & \cdots & \bullet \end{array} \right) \stackrel{\text{def}}{=} \left( \begin{array}{c|c} \mathbb{U}^{[k]} & \mathbb{F}^{[k]} \\ \hline \mathbb{0} & \mathbb{V}^{[k]} \end{array} \right)$$

Nous sommes exactement dans le cas de figure étudié dans l'exercice 3.1.3, page 64. En effet, avec les notations de cet exercice et si l'on pose  $\mathbf{v} = \mathbb{V}_{:,1}^{[k]} = (A_{k+1,k+1}^{[k]}, \dots, A_{n,k+1}^{[k]})^t \in \mathbb{C}^{n-(k+1)}$  (en bleu dans l'expression de  $\mathbb{A}^{[k]}$  précédente) on a alors  $v_1 = A_{k+1,k+1}^{[k]} = \alpha_{k+1} \neq 0$  et l'on peut définir la matrice  $\mathbb{E}^{[k+1]} \in \mathcal{M}_n(\mathbb{C})$ , triangulaire inférieure à diagonale unité, par

$$\mathbb{E}^{[k+1]} = \mathbb{E}^{[k,\mathbf{v}]} \stackrel{\text{def}}{=} \left( \begin{array}{c|c} \mathbb{I}_k & \mathbb{0} \\ \hline \mathbb{0} & \mathbb{E}^{[\mathbf{v}]} \end{array} \right)$$

avec  $\mathbb{E}^{[\mathbf{v}]} \in \mathcal{M}_{n-k}(\mathbb{C})$  triangulaire inférieure à diagonale unité (définie dans l'exercice 3.1.3) telle que

$$\mathbb{E}^{[\mathbf{v}]} \mathbb{V}^{[k]} = \begin{pmatrix} \alpha_{k+1} & \bullet & \cdots & \bullet \\ 0 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & \bullet & \cdots & \bullet \end{pmatrix}$$

On a alors

$$\begin{aligned} \mathbb{A}^{[k+1]} \stackrel{\text{def}}{=} \mathbb{E}^{[k+1]} \mathbb{A}^{[k]} &= \left( \begin{array}{c|c} \mathbb{I}_k & \mathbb{0} \\ \hline \mathbb{0} & \mathbb{E}^{[\mathbf{v}]} \end{array} \right) \left( \begin{array}{c|c} \mathbb{U}^{[k]} & \mathbb{F}^{[k]} \\ \hline \mathbb{0} & \mathbb{V}^{[k]} \end{array} \right) \\ &= \left( \begin{array}{c|c} \mathbb{U}^{[k]} & \mathbb{F}^{[k]} \\ \hline \mathbb{0} & \mathbb{E}^{[\mathbf{v}]} \mathbb{V}^{[k]} \end{array} \right) \end{aligned}$$

et donc  $\mathbb{A}^{[k+1]}$  s'écrit bien sous la forme (3.6) au rang  $k+1$ .

**Final** ( $k = n-1$ ): On a donc

$$\mathbb{U} = \mathbb{A}^{[n-1]} \stackrel{\text{def}}{=} \mathbb{E}^{[n-1]} \times \cdots \times \mathbb{E}^{[1]} \times \mathbb{A} = \left( \begin{array}{cccc|c} \alpha_1 & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \alpha_{n-1} & \bullet \\ \hline 0 & \cdots & \cdots & 0 & \bullet \end{array} \right) \quad (3.7)$$

où pour tout  $k \in \llbracket 1, n-1 \rrbracket$  les matrices  $\mathbb{E}^{[k]}$  sont triangulaires inférieures à diagonale unité.

Pour achever l'exercice, il reste à démontrer que

$$U_{n,n} = \det \Delta_n / (U_{1,1} \times \cdots \times U_{n-1,n-1}).$$

En effet, en prenant le déterminant dans (3.7) on obtient

$$\det \left( \mathbb{E}^{[n-1]} \times \cdots \times \mathbb{E}^{[1]} \times \mathbb{A} \right) = \det \left( \begin{array}{cccc|c} U_{1,1} & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & U_{n-1,n-1} & \bullet \\ \hline 0 & \cdots & \cdots & 0 & U_{n,n} \end{array} \right)$$

Comme le déterminant d'un produit de matrices est égale au produit des déterminants des matrices on a

$$\det \left( \mathbb{E}^{[n-1]} \times \dots \times \mathbb{E}^{[1]} \times \mathbb{A} \right) = \det \mathbb{E}^{[n-1]} \times \dots \times \det \mathbb{E}^{[1]} \times \det \mathbb{A} \\ = \det \mathbb{A}$$

car les matrices  $\mathbb{E}^{[k]}$  sont triangulaires inférieures à diagonale unité et donc  $\det \mathbb{E}^{[k]} = 1, \forall k \in \llbracket 1, n-1 \rrbracket$ . De plus, le déterminant d'une matrice triangulaire supérieure est égale au produit de ses coefficients diagonaux et donc

$$\det \left( \begin{array}{cccc|ccc} U_{1,1} & \bullet & \cdots & \cdots & & & \\ 0 & \ddots & \ddots & & & & \\ \vdots & \ddots & \ddots & & & & \\ 0 & \cdots & 0 & U_{n-1,n-1} & & & \\ \hline 0 & \cdots & \cdots & 0 & & & \\ & & & & & & U_{n,n} \end{array} \right) = U_{n,n} \prod_{k=1}^{n-1} U_{k,k}.$$

On a alors

$$\det \mathbb{A} = \det \Delta_n = U_{n,n} \prod_{k=1}^{n-1} U_{k,k} \neq 0.$$

~~~~~ Fin correction ~~~~~

**Théorème 3.1.4** (Factorisation LU ★★★★★). Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice dont les sous-matrices principales sont inversibles alors il existe

- une unique matrice  $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure (lower triangular en anglais) à diagonale unité,
- une unique matrice  $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$  triangulaire supérieure (upper triangular en anglais) inversible

telles que

$$\mathbb{A} = \mathbb{L}\mathbb{U}.$$

*Preuve.* Nous allons voir deux méthodes pour démontrer l'existence.

**méthode 1** En utilisant le résultat de l'exercice 3.1.6, il existe des matrices  $\mathbb{E}^{[k]} \in \mathcal{M}_n(\mathbb{C}), k \in \llbracket 1, n-1 \rrbracket$ , triangulaires inférieures à diagonale unité telles que la matrice  $\mathbb{U}$  définie par

$$\mathbb{U} = \mathbb{E}^{[n-1]} \dots \mathbb{E}^{[1]} \mathbb{A}$$

soit triangulaire supérieure avec  $U_{i,i} \neq 0, \forall i \in \llbracket 1, n \rrbracket$ . On pose  $\mathbb{G} = \mathbb{E}^{[n-1]} \dots \mathbb{E}^{[1]}$ . La matrice  $\mathbb{G}$  est donc aussi triangulaire inférieure à diagonale unité. Elle est donc inversible et son inverse est triangulaire inférieure à diagonale unité (voir Proposition B.2.8, page 225). En notant  $\mathbb{L} = \mathbb{G}^{-1}$  on a  $\mathbb{A} = \mathbb{L}\mathbb{U}$ .

**méthode 2** Nous allons effectuer une démonstration par récurrence sur l'ordre  $n$  de la matrice  $\mathbb{A}$ .

**Propriété:** Pour tout  $n \in \mathbb{N}^*$ , si une matrice  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  a toutes ses sous-matrices principales inversibles alors il existe une unique matrice  $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure à diagonale unité et une unique matrice  $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$  triangulaire supérieure inversible telles que  $\mathbb{A} = \mathbb{L}\mathbb{U}$ .

**Initialisation:** Pour  $n = 1$ , c'est trivial  $\mathbb{L} = 1$  et  $\mathbb{U} = \mathbb{A}$ .

**Hérédité:** Supposons la propriété vraie au rang  $n$ , montrons qu'alors, la propriété est vraie au rang  $n+1$ .

Soit  $\mathbb{A} \in \mathcal{M}_{n+1}(\mathbb{C})$  telle que toutes les sous-matrices principales soient inversibles. On peut la décomposer sous la forme bloc suivante

$$\mathbb{A} = \left( \begin{array}{c|c} \mathbb{A} & \mathbf{f} \\ \hline \mathbf{e}^* & d \end{array} \right)$$

où

- $\underline{A}$  est la matrice de  $\mathcal{M}_n(\mathbb{C})$  telle que  $\underline{A}_{i,j} = A_{i,j}, \forall (i,j) \in \llbracket 1, n \rrbracket^2$ ,
- $\mathbf{f}$  est le vecteur de  $\mathbb{C}^n$  tel que  $f_i = A_{i,n+1}, \forall i \in \llbracket 1, n \rrbracket$ ,
- $\mathbf{e}$  est le vecteur de  $\mathbb{C}^n$  tel que  $e_i = \overline{A_{n+1,i}}, \forall i \in \llbracket 1, n \rrbracket$
- $d \in \mathbb{C}$  est le scalaire  $d = A_{n+1,n+1}$ .

Comme les sous-matrices principales de  $\underline{A}$  sont les  $n$  premières sous-matrices principales de  $A$ , elles sont donc inversibles. Par hypothèse de récurrence, il existe une unique matrice  $\underline{L} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure à diagonale unité et une unique matrice  $\underline{U} \in \mathcal{M}_n(\mathbb{C})$  triangulaire supérieure inversible telles que

$$\underline{A} = \underline{L} \underline{U}.$$

On va construire des matrices  $\mathbb{L}$  et  $\mathbb{U}$  vérifiant la propriété. Soient  $\mathbb{L} \in \mathcal{M}_{n+1}(\mathbb{C})$  et  $\mathbb{U} \in \mathcal{M}_{n+1}(\mathbb{C})$  décomposées sous la forme bloc

$$\mathbb{L} = \left( \begin{array}{c|c} \underline{L} & \mathbf{0} \\ \hline \mathbf{g}^* & 1 \end{array} \right) \text{ et } \mathbb{U} = \left( \begin{array}{c|c} \underline{U} & \mathbf{h} \\ \hline \mathbf{0}^* & \alpha \end{array} \right)$$

où  $\mathbf{g} \in \mathbb{C}^n$ ,  $\mathbf{h} \in \mathbb{C}^n$  and  $\alpha \in \mathbb{C}$ . Les deux matrices sont blocs compatibles pour la multiplication et on a alors

$$\mathbb{L}\mathbb{U} = \left( \begin{array}{c|c} \underline{L} & \mathbf{0} \\ \hline \mathbf{g}^* & 1 \end{array} \right) \left( \begin{array}{c|c} \underline{U} & \mathbf{h} \\ \hline \mathbf{0}^* & \alpha \end{array} \right) = \left( \begin{array}{c|c} \underline{L}\underline{U} & \underline{L}\mathbf{h} \\ \hline \mathbf{g}^*\underline{U} & \mathbf{g}^*\mathbf{h} + \alpha \end{array} \right)$$

Comme  $\underline{A} = \underline{L} \underline{U}$ , on va identifier bloc par bloc les matrices  $\mathbb{L}\mathbb{U}$  et  $\underline{A}$ , puis vérifier que le système est résoluble. On obtient donc par identification le système:

$$\begin{cases} \underline{L}\mathbf{h} & = \mathbf{f} \\ \mathbf{g}^*\underline{U} & = \mathbf{e}^* \\ \mathbf{g}^*\mathbf{h} + \alpha & = d \end{cases}$$

Par hypothèse de récurrence  $\underline{L}$  est triangulaire inférieure à diagonale unité et  $\underline{U}$  triangulaire supérieure inversible: le système précédant admet donc comme solution

$$\begin{cases} \mathbf{h} & = \underline{L}^{-1}\mathbf{f} \\ \mathbf{g} & = \underline{U}^{*-1}\mathbf{e} \\ \alpha & = d - \mathbf{g}^*\mathbf{h} \end{cases}$$

On a donc construit une matrice  $\mathbb{L}$  triangulaire inférieure à diagonale unité et une matrice  $\mathbb{U}$  triangulaire supérieure telles que  $\underline{A} = \mathbb{L}\mathbb{U}$ . Comme la matrice  $\underline{A}$  est inversible, on en déduit que

$$\det \underline{A} = \det(\mathbb{L}\mathbb{U}) = \det \mathbb{L} \times \det \mathbb{U} \neq 0$$

On obtient alors  $\det \mathbb{U} \neq 0$  ce qui entraîne l'inversibilité de la matrice  $\mathbb{U}$ .

On vient de démontrer l'existence d'une factorisation  $\mathbb{L}\mathbb{U}$  de la matrice  $\underline{A}$ . Pour démontrer l'unicité, on va supposer qu'il existe deux factorisations  $\mathbb{L}\mathbb{U}$  de  $\underline{A}$  i.e.

$$\underline{A} = \mathbb{L}_1\mathbb{U}_1 = \mathbb{L}_2\mathbb{U}_2.$$

avec  $\mathbb{L}_1, \mathbb{L}_2$  matrices triangulaires inférieures à diagonale unité et  $\mathbb{U}_1, \mathbb{U}_2$  matrices triangulaires supérieures (inversibles). En multipliant l'équation  $\mathbb{L}_1\mathbb{U}_1 = \mathbb{L}_2\mathbb{U}_2$  à gauche par  $\mathbb{L}_1^{-1}$  et à droite par  $\mathbb{U}_2^{-1}$  on obtient

$$\mathbb{U}_1\mathbb{U}_2^{-1} = \mathbb{L}_1^{-1}\mathbb{L}_2. \quad (3.8)$$

La matrice  $\mathbb{L}_1^{-1}\mathbb{L}_2$  est triangulaire inférieure à diagonale unité car produit de deux matrices triangulaires inférieures à diagonale unité. Elle est égale à la matrice  $\mathbb{U}_1\mathbb{U}_2^{-1}$  qui elle est triangulaire supérieure (car produit de deux matrices triangulaires supérieures). Donc  $\mathbb{L}_1^{-1}\mathbb{L}_2$  est à la fois une matrice triangulaire supérieure et inférieure : elle est donc diagonale. Comme elle est à diagonale unité on en déduit que  $\mathbb{L}_1^{-1}\mathbb{L}_2 = \mathbb{I}$  et donc  $\mathbb{L}_1 = \mathbb{L}_2$ . De l'équation (3.8), on tire alors  $\mathbb{U}_1 = \mathbb{U}_2$ .  $\square$

**EXERCICE 3.1.7**

Montrer que si  $\mathbb{A}$  inversible admet une factorisation LU alors toutes ses sous-matrices principales sont inversibles.

**Remarque 3.1.1.** Si la matrice  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  est inversible et si ses sous-matrices principales ne sont pas toutes inversibles, il est possible par des permutations préalables de lignes de la matrice de se ramener à une matrice telle que ses sous-matrices principales soient inversibles.

**Théorème 3.1.5** (Factorisation LU avec permutations ★★★★★). Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice inversible. Il existe une matrice  $\mathbb{P}$ , produit de matrices de permutation, une matrice  $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure à diagonale unité et une matrice  $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$  triangulaire supérieure inversible telles que

$$\mathbb{P}\mathbb{A} = \mathbb{L}\mathbb{U}. \quad (3.9)$$

**Corollaire 3.1.1** ( ★★★★★ ). Si  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  est une matrice hermitienne définie positive alors elle admet une unique factorisation LU.

*Preuve. (indication)* Si la matrice  $\mathbb{A}$  est hermitienne définie positive alors toutes ses sous-matrices principales sont définies positives et donc inversibles. □

**Résolution d'un système linéaire par factorisation LU**

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  une matrice dont les sous-matrices principales sont inversibles et  $\mathbf{b} \in \mathbb{K}^n$ . On veut résoudre le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ . Pour cela, grâce au théorème 3.1.4, on obtient :

Trouver  $\mathbf{x} \in \mathbb{K}^n$  tel que

$$\mathbb{A}\mathbf{x} = \mathbf{b}. \quad (3.10)$$

est équivalent à

Trouver  $\mathbf{x} \in \mathbb{K}^n$  solution de

$$\mathbb{U}\mathbf{x} = \mathbf{y} \quad (3.11)$$

avec  $\mathbf{y} \in \mathbb{K}^n$  solution de

$$\mathbb{L}\mathbf{y} = \mathbf{b}. \quad (3.12)$$

Ceci permet donc de découper le problème initial en trois sous-problèmes plus simples. De plus, ceux-ci peuvent se traiter de manière indépendante. On obtient alors l'algorithme suivant

**Algorithm 3.9** Fonction RSLFactLU permettant de résoudre, par une factorisation LU, le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  où  $\mathbb{A}$  est une matrice de  $\mathcal{M}_n(\mathbb{K})$ , dont toutes les sous-matrices principales sont inversibles, et  $\mathbf{b} \in \mathbb{K}^n$ .

, **Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les sous-matrices principales sont inversibles;  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{K}^n$ .

1: **Fonction**  $\mathbf{x} \leftarrow \text{RSLFactLU}(\mathbb{A}, \mathbf{b})$

2:  $[\mathbb{L}, \mathbb{U}] \leftarrow \text{FactLU}(\mathbb{A})$

3:  $\mathbf{y} \leftarrow \text{RSLTriInf}(\mathbb{L}, \mathbf{b})$

4:  $\mathbf{x} \leftarrow \text{RSLTriSup}(\mathbb{U}, \mathbf{y})$

5: **Fin**

▷ Factorisation LU

▷ Résolution du système  $\mathbb{L}\mathbf{y} = \mathbf{b}$

▷ Résolution du système  $\mathbb{U}\mathbf{x} = \mathbf{y}$

Il nous faut donc écrire la fonction **FactLU** (les deux fonctions **RSLTriInf** et **RSLTriSup** ayant déjà été écrites).

### Détermination des matrices $\mathbb{L}$ et $\mathbb{U}$

Soit  $A \in \mathcal{M}_n(\mathbb{K})$  une matrice dont les sous-matrices principales sont inversibles. D'après le théorème 3.1.4, il existe une unique matrice  $\mathbb{L} \in \mathcal{M}_n(\mathbb{K})$  triangulaire inférieure avec  $L_{i,i} = 1, \forall i \in \llbracket 1, n \rrbracket$ , et une unique matrice  $\mathbb{U} \in \mathcal{M}_n(\mathbb{K})$  triangulaire supérieure telles que

$$A = \mathbb{L}\mathbb{U} \quad (3.13)$$

c'est à dire

$$\begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ L_{2,1} & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ L_{n,1} & \cdots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & \cdots & \cdots & U_{n,1} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & U_{n,n} \end{pmatrix}. \quad (3.14)$$

Pour déterminer les matrices  $\mathbb{L}$  et  $\mathbb{U}$ , on remarque que la 1ère ligne de  $\mathbb{L}$  est déjà déterminée. On peut alors l'utiliser pour calculer la première ligne de  $\mathbb{U} : \forall j \in \llbracket 1, n \rrbracket$

$$\begin{aligned} A_{1,j} &= \sum_{k=1}^n L_{1,k} U_{k,j} \\ &= L_{1,1} U_{1,j} \text{ car } \mathbb{L} \text{ triangulaire inférieure} \\ &= U_{1,j} \end{aligned}$$

On a donc

$$U_{1,j} = A_{1,j}, \quad \forall j \in \llbracket 1, n \rrbracket. \quad (3.15)$$

La première colonne de  $\mathbb{U}$  est aussi déterminée, on peut alors l'utiliser pour calculer la première colonne de  $\mathbb{L} : \forall i \in \llbracket 2, n \rrbracket$

$$\begin{aligned} A_{i,1} &= \sum_{k=1}^n L_{i,k} U_{k,1} \\ &= L_{i,1} U_{1,1} \text{ car } \mathbb{U} \text{ triangulaire supérieure} \end{aligned}$$

On peut démontrer, de part les hypothèses sur la matrice  $A$ , que  $U_{1,1} \neq 0$  et alors

$$L_{1,j} = A_{i,j}/U_{1,1}, \quad \forall i \in \llbracket 2, n \rrbracket. \quad (3.16)$$

La première ligne de  $\mathbb{U}$  et la première colonne de  $\mathbb{L}$  sont donc déterminées par les formules (3.15) et (3.16). Par récurrence, on suppose connues les  $i-1$  premières lignes de  $\mathbb{U}$  et les  $i-1$  premières colonnes de  $\mathbb{L}$ . On va montrer que l'on peut expliciter la  $i$ -ème ligne de  $\mathbb{U}$  et la  $i$ -ème colonne de  $\mathbb{L}$ .

En effet,  $\forall j \in \llbracket i, n \rrbracket$ , on a

$$\begin{aligned} A_{i,j} &= \sum_{k=1}^n L_{i,k} U_{k,j} \\ &= \sum_{k=1}^{i-1} L_{i,k} U_{k,j} + L_{i,i} U_{i,j} + \sum_{k=i+1}^n L_{i,k} U_{k,j} \\ &= \sum_{k=1}^{i-1} L_{i,k} U_{k,j} + L_{i,i} U_{i,j} \text{ car } \mathbb{L} \text{ triangulaire inférieure} \end{aligned}$$

Dans l'expression  $\sum_{k=1}^{i-1} L_{i,k} U_{k,j}$  tous les termes sont connus (hypothèse de récurrence) et  $L_{i,i} = 1$ . On en déduit,

$$\text{Pour } i \text{ allant de } 1 \text{ à } n : U_{i,j} = \begin{cases} A_{i,j} - \sum_{k=1}^{i-1} L_{i,k} U_{k,j}, & \forall j \in \llbracket i, n \rrbracket. \\ 0, & \forall j \in \llbracket 1, i-1 \rrbracket. \end{cases} \quad (3.17)$$

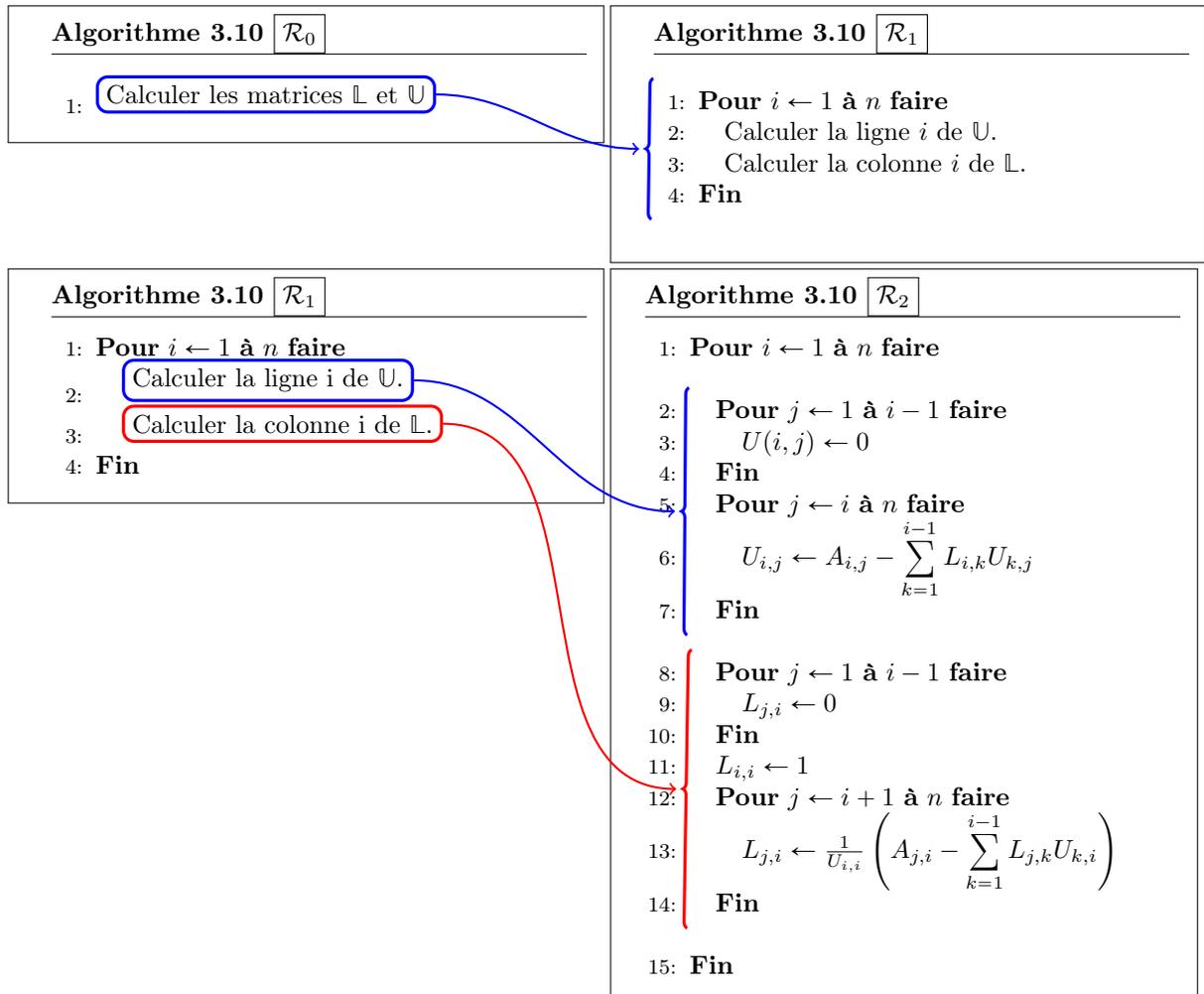
Maintenant, on calcule,  $\forall j \in \llbracket i + 1, n \rrbracket$ ,

$$\begin{aligned} A_{j,i} &= \sum_{k=1}^n L_{j,k} U_{k,i} \\ &= \sum_{k=1}^{i-1} L_{j,k} U_{k,i} + L_{j,i} U_{i,i} + \sum_{k=i+1}^n L_{j,k} U_{k,i} \\ &= \sum_{k=1}^{i-1} L_{j,k} U_{k,i} + L_{j,i} U_{i,i} \text{ car } \mathbb{U} \text{ triangulaire supérieure} \end{aligned}$$

Dans l'expression  $\sum_{k=1}^{i-1} L_{j,k} U_{k,i}$  tous les termes sont connus (hypothèse de récurrence). De plus  $U_{i,i}$  est donné par (3.17) et on peut démontrer, de part les hypothèses sur la matrice  $\mathbb{A}$ , que  $U_{i,i} \neq 0$ . On a alors

$$\text{Pour } i \text{ allant de } 1 \text{ à } n : L_{j,i} = \begin{cases} 0, & \forall j \in \llbracket 1, i-1 \rrbracket. \\ 1, & j = i \\ \frac{1}{U_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} L_{j,k} U_{k,i} \right), & \forall j \in \llbracket i+1, n \rrbracket, \end{cases} \quad (3.18)$$

On écrit en détail les raffinements successifs permettant d'aboutir à l'algorithme final de telle sorte que le passage entre deux raffinements successifs soit le plus compréhensible possible.



Algorithme 3.10  $\mathcal{R}_2$ 

```

1: Pour  $i \leftarrow 1$  à  $n$  faire
2:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
3:      $U(i, j) \leftarrow 0$ 
4:   Fin
5:   Pour  $j \leftarrow i$  à  $n$  faire
6:      $U_{i,j} \leftarrow A_{i,j} - \sum_{k=1}^{i-1} L_{i,k} U_{k,j}$ 
7:   Fin
8:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
9:      $L_{j,i} \leftarrow 0$ 
10:  Fin
11:   $L_{i,i} \leftarrow 1$ 
12:  Pour  $j \leftarrow i + 1$  à  $n$  faire
13:     $L_{j,i} \leftarrow \frac{1}{U_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} L_{j,k} U_{k,i} \right)$ 
14:  Fin
15: Fin

```

Algorithme 3.10  $\mathcal{R}_3$ 

```

1: Pour  $i \leftarrow 1$  à  $n$  faire
2:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
3:      $U(i, j) \leftarrow 0$ 
4:   Fin
5:   Pour  $j \leftarrow i$  à  $n$  faire
6:      $S_1 \leftarrow \sum_{k=1}^{i-1} L_{i,k} U_{k,j}$ 
7:      $U_{i,j} \leftarrow A_{i,j} - S_1$ 
8:   Fin
9:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
10:     $L_{j,i} \leftarrow 0$ 
11:  Fin
12:   $L_{i,i} \leftarrow 1$ 
13:  Pour  $j \leftarrow i + 1$  à  $n$  faire
14:     $S_2 \leftarrow \sum_{k=1}^{i-1} L_{j,k} U_{k,i}$ 
15:     $L_{j,i} \leftarrow \frac{1}{U_{i,i}} (A_{j,i} - S_2)$ 
16:  Fin
17: Fin

```

Algorithme 3.10  $\mathcal{R}_3$ 

```

1: Pour  $i \leftarrow 1$  à  $n$  faire
2:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
3:      $U(i, j) \leftarrow 0$ 
4:   Fin
5:   Pour  $j \leftarrow i$  à  $n$  faire
6:      $S_1 \leftarrow \sum_{k=1}^{i-1} L_{i,k} U_{k,j}$ 
7:      $U_{i,j} \leftarrow A_{i,j} - S_1$ 
8:   Fin
9:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
10:     $L_{j,i} \leftarrow 0$ 
11:  Fin
12:   $L_{i,i} \leftarrow 1$ 
13:  Pour  $j \leftarrow i + 1$  à  $n$  faire
14:     $S_2 \leftarrow \sum_{k=1}^{i-1} L_{j,k} U_{k,i}$ 
15:     $L_{j,i} \leftarrow \frac{1}{U_{i,i}} (A_{j,i} - S_2)$ 
16:  Fin
17: Fin

```

Algorithme 3.10  $\mathcal{R}_4$ 

```

1: Pour  $i \leftarrow 1$  à  $n$  faire
2:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
3:      $U(i, j) \leftarrow 0$ 
4:   Fin
5:   Pour  $j \leftarrow i$  à  $n$  faire
6:      $S_1 \leftarrow 0$ 
7:     Pour  $k \leftarrow 1$  à  $i - 1$  faire
8:        $S_1 \leftarrow S_1 + L_{i,k} * U_{k,j}$ 
9:     Fin
10:     $U_{i,j} \leftarrow A_{i,j} - S_1$ 
11:  Fin
12:  Pour  $j \leftarrow 1$  à  $i - 1$  faire
13:     $L_{j,i} \leftarrow 0$ 
14:  Fin
15:   $L_{i,i} \leftarrow 1$ 
16:  Pour  $j \leftarrow i + 1$  à  $n$  faire
17:     $S_2 \leftarrow 0$ 
18:    Pour  $k \leftarrow 1$  à  $i - 1$  faire
19:       $S_2 \leftarrow S_2 + L_{j,k} * U_{k,i}$ 
20:    Fin
21:     $L_{j,i} \leftarrow \frac{1}{U_{i,i}} (A_{j,i} - S_2)$ 
22:  Fin
23: Fin

```

L'algorithme peut être amélioré, pour gagner en lisibilité... En effet, il est possible d'initialiser la matrice  $\mathbb{U}$  par la matrice nulle et la matrice  $\mathbb{L}$  par la matrice identité, ce qui permet alors de supprimer les

boucles  $U(i, j) \leftarrow 0$  et  $L(j, i) \leftarrow 0$  ainsi que la commande  $L(i, i) \leftarrow 1$ . On obtient alors l'algorithme final

**Algorithm 3.10** Fonction **FactLU** permet de calculer les matrices  $\mathbb{L}$  et  $\mathbb{U}$  dites matrice de factorisation  $\mathbb{LU}$  associée à la matrice  $A$ , telle que

$$A = \mathbb{L}\mathbb{U}$$

**Données :**  $A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les sous-matrices principales sont inversibles.

**Résultat :**  $\mathbb{L}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  triangulaire inférieure avec  $L_{i,i} = 1, \forall i \in \llbracket 1, n \rrbracket$

$\mathbb{U}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  triangulaire supérieure.

```

1: Fonction  $[\mathbb{L}, \mathbb{U}] \leftarrow \text{FactLU}(A)$ 
2:    $\mathbb{U} \leftarrow \mathbb{O}_n$ 
3:    $\mathbb{L} \leftarrow \mathbb{I}_n$ 
4:   Pour  $i \leftarrow 1$  à  $n$  faire
5:     Pour  $j \leftarrow i$  à  $n$  faire
6:        $S_1 \leftarrow 0$ 
7:       Pour  $k \leftarrow 1$  à  $i - 1$  faire
8:          $S_1 \leftarrow S_1 + L(i, k) * U(k, j)$ 
9:       Fin
10:       $U(i, j) \leftarrow A(i, j) - S_1$ 
11:    Fin
12:    Pour  $j \leftarrow i + 1$  à  $n$  faire
13:       $S_2 \leftarrow 0$ 
14:      Pour  $k \leftarrow 1$  à  $i - 1$  faire
15:         $S_2 \leftarrow S_2 + L(j, k) * U(k, i)$ 
16:      Fin
17:       $L(j, i) \leftarrow (A_{j,i} - S_2) / U(i, i)$ .
18:    Fin
19:  Fin
20: Fin

```

$\triangleright \mathbb{O}_n$  matrice nulle  $n \times n$   
 $\triangleright \mathbb{I}_n$  matrice identité  $n \times n$   
 $\triangleright$  Calcul de la ligne  $i$  de  $\mathbb{U}$   
 $\triangleright$  Calcul de la colonne  $i$  de  $\mathbb{L}$

**Remarque 3.1.2.** Pour optimiser en mémoire cette fonction, il est possible de stocker les matrices  $\mathbb{L}$  et  $\mathbb{U}$  dans une même matrice de  $\mathcal{M}_n(\mathbb{K})$  ...

Pour faciliter la lecture d'un tel algorithme, il aurait pu être judicieux d'utiliser deux fonctions intermédiaires **FactLUigU** et **FactLUcolL** qui à l'étape  $i$  de l'algorithme calculent respectivement la ligne  $i$  de  $\mathbb{U}$  et la colonne  $i$  de  $\mathbb{L}$ .

**Algorithm 3.11** Fonction **FactLUligU** permet de calculer la ligne  $i$  de  $\mathbb{U}$  à partir de (3.17)

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les sous-matrices principales sont inversibles.  
 $\mathbb{L}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i-1$  premières colonnes ont été calculées  
 $\mathbb{U}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i-1$  premières lignes ont été calculées  
**Résultat :**  $\mathbb{U}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i$  premières lignes ont été calculées

```

1: Fonction  $\mathbb{U} \leftarrow \text{FactLUligU}(\mathbb{A}, \mathbb{L}, \mathbb{U}, i)$ 
2: Pour  $j \leftarrow 1$  à  $i-1$  faire
3:    $U(i, j) \leftarrow 0$ 
4: Fin
5: Pour  $j \leftarrow i$  à  $n$  faire
6:    $S_1 \leftarrow 0$ 
7:   Pour  $k \leftarrow 1$  à  $i-1$  faire
8:      $S_1 \leftarrow S_1 + L(i, k) * U(k, j)$ 
9:   Fin
10:   $U(i, j) \leftarrow A(i, j) - S_1$ 
11: Fin
12: Fin

```

**Algorithm 3.12** Fonction **FactLUcolL** permet de calculer la colonne  $i$  de  $\mathbb{L}$  à partir de (3.18)

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les sous-matrices principales sont inversibles.  
 $\mathbb{L}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i-1$  premières colonnes ont été calculées  
 $\mathbb{U}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i$  premières lignes ont été calculées  
**Résultat :**  $\mathbb{L}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  dont les  $i$  premières colonnes ont été calculées

```

1: Fonction  $\mathbb{L} \leftarrow \text{FactLUcolL}(\mathbb{A}, \mathbb{L}, \mathbb{U}, i)$ 
2: Pour  $j \leftarrow 1$  à  $i-1$  faire
3:    $L(j, i) \leftarrow 0$ 
4: Fin
5:  $L(i, i) \leftarrow 1$ 
6: Pour  $j \leftarrow i+1$  à  $n$  faire
7:    $S_2 \leftarrow 0$ 
8:   Pour  $k \leftarrow 1$  à  $i-1$  faire
9:      $S_2 \leftarrow S_2 + L(j, k) * U(k, i)$ 
10:  Fin
11:   $L(j, i) \leftarrow (A_{j,i} - S_2) / U(i, i)$ 
12: Fin
13: Fin

```

On a alors

**Algorithm 3.13** Fonction **FactLU** permet de calculer les matrices  $\mathbb{L}$  et  $\mathbb{U}$  dites matrice de factorisation  $\mathbb{L}\mathbb{U}$  associée à la matrice  $\mathbb{A}$ , telle que

$$\mathbb{A} = \mathbb{L}\mathbb{U}$$

en utilisant des fonctions intermédiaires.

```

1: Fonction  $[\mathbb{L}, \mathbb{U}] \leftarrow \text{FactLU}(\mathbb{A})$ 
2: Pour  $i \leftarrow 1$  à  $n$  faire
3:    $\mathbb{U} \leftarrow \text{FactLUligU}(\mathbb{A}, \mathbb{L}, \mathbb{U}, i)$ 
4:    $\mathbb{L} \leftarrow \text{FactLUcolL}(\mathbb{A}, \mathbb{L}, \mathbb{U}, i)$ 
5: Fin
6: Fin

```

### 3.1.5 Factorisation $\mathbb{L}\mathbb{D}\mathbb{L}^*$

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne inversible admettant une factorisation  $\mathbb{L}\mathbb{U}$ . On note  $\mathbb{D}$  la matrice diagonale inversible définie par  $\mathbb{D} = \text{diag } \mathbb{U}$  (i.e.  $D_{i,i} = U_{i,i} \neq 0, \forall i \in \llbracket 1, n \rrbracket$ ) et  $\mathbb{R} = \mathbb{D}^{-1}\mathbb{U}$ . La matrice  $\mathbb{R}$  est donc matrice triangulaire supérieure à diagonale unité car

$$R_{i,i} = (\mathbb{D}^{-1}\mathbb{U})_{i,i} = \sum_{k=1}^n (\mathbb{D}^{-1})_{i,k} U_{k,i} = (\mathbb{D}^{-1})_{i,i} U_{i,i} = \frac{1}{U_{i,i}} U_{i,i} = 1.$$

On a alors

$$\mathbb{A} = \mathbb{L}\mathbb{U} = \mathbb{L}\mathbb{D}\mathbb{D}^{-1}\mathbb{U} = \mathbb{L}\mathbb{D}\mathbb{R}.$$

De plus comme  $\mathbb{A}$  est hermitienne on a

$$\mathbb{A}^* = \mathbb{A} \iff \mathbb{R}^* \mathbb{D}^* \mathbb{L}^* = \mathbb{L}\mathbb{D}\mathbb{R}$$

Ce qui donne

$$\mathbb{A} = \mathbb{R}^* (\mathbb{D}^* \mathbb{L}^*) = \mathbb{L}(\mathbb{D}\mathbb{R})$$

et donc par unicité de la factorisation  $\mathbb{L}\mathbb{U}$  on a  $\mathbb{R}^* = \mathbb{L}$  et  $\mathbb{D}^* \mathbb{L}^* = \mathbb{D}\mathbb{R}$ . On obtient alors

$$\mathbb{R}^* = \mathbb{L} \text{ et } \mathbb{D}^* = \mathbb{D}$$

et on a le théorème suivant

**Théorème 3.1.6** (Factorisation LDL\* ★★★★★). Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne inversible admettant une factorisation LU. Alors  $A$  s'écrit sous la forme

$$A = LDL^* \quad (3.19)$$

où  $D = \text{diag } U$  est une matrice à coefficients réels.

**Corollaire 3.1.2** (★★★★★). Une matrice  $A \in \mathcal{M}_n(\mathbb{C})$  admet une factorisation LDL\* avec  $L \in \mathcal{M}_n(\mathbb{C})$  matrice triangulaire inférieure à diagonale unité et  $D \in \mathcal{M}_n(\mathbb{R})$  matrice diagonale à coefficients diagonaux strictement positifs si et seulement si la matrice  $A$  est hermitienne définie positive.

*Preuve.*  $\Rightarrow$  Soit  $A \in \mathcal{M}_n(\mathbb{C})$  admettant une factorisation LDL\* avec  $L \in \mathcal{M}_n(\mathbb{C})$  matrice triangulaire inférieure à diagonale unité et  $D \in \mathcal{M}_n(\mathbb{R})$  matrice diagonale à coefficients diagonaux strictement positifs.

La matrice  $A$  est alors hermitienne car

$$A^* = (LDL^*)^* = L^{**}D^*L^* = LDL^*.$$

De plus  $\forall \mathbf{x} \in \mathbb{C}^n \setminus \{0\}$  on a

$$\langle A\mathbf{x}, \mathbf{x} \rangle = \langle LDL^*\mathbf{x}, \mathbf{x} \rangle = \langle DL^*\mathbf{x}, L^*\mathbf{x} \rangle$$

On pose  $\mathbf{y} = L^*\mathbf{x} \neq 0$  car  $\mathbf{x} \neq 0$  et  $L^*$  inversible. On obtient alors

$$\langle A\mathbf{x}, \mathbf{x} \rangle = \langle D\mathbf{y}, \mathbf{y} \rangle = \sum_{i=1}^n D_{i,i}|y_i|^2 > 0$$

car  $D$  diagonale,  $D_{i,i} > 0$ ,  $\forall i \in \llbracket 1, n \rrbracket$  et  $\mathbf{y} \neq 0$ .

La matrice hermitienne  $A$  est donc bien définie positive.

$\Leftarrow$  Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne définie positive.

D'après le Corollaire 3.1.1, page 78, la matrice  $A$  admet une unique factorisation LU et donc d'après le Théorème 3.1.2, page 84, la matrice hermitienne  $A$  peut s'écrire sous la forme  $A = LDL^*$  où  $D$  est diagonale à coefficients réels et  $L$  triangulaire inférieure à diagonale unité. Il reste à démontrer que  $D_{i,i} > 0$ ,  $\forall i \in \llbracket 1, n \rrbracket$ .

Comme  $A$  est définie positive, on a  $\forall \mathbf{x} \in \mathbb{C}^n \setminus \{0\}$ ,  $\langle A\mathbf{x}, \mathbf{x} \rangle > 0$ . Or on a

$$\langle A\mathbf{x}, \mathbf{x} \rangle = \langle LDL^*\mathbf{x}, \mathbf{x} \rangle = \langle DL^*\mathbf{x}, L^*\mathbf{x} \rangle$$

On note  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ , la base canonique de  $\mathbb{C}^n$  et on rappelle que  $\forall i \in \llbracket 1, n \rrbracket$ ,  $\langle D\mathbf{e}_i, \mathbf{e}_i \rangle = D_{i,i}$ . Soit  $i \in \llbracket 1, n \rrbracket$ . En choisissant  $\mathbf{x} = (L^*)^{-1}\mathbf{e}_i \neq 0$ , on obtient alors

$$\langle DL^*\mathbf{x}, L^*\mathbf{x} \rangle = \langle D\mathbf{e}_i, \mathbf{e}_i \rangle = D_{i,i} > 0.$$

□

### 3.1.6 Factorisation de Cholesky

**Définition 3.1.1.** Une *factorisation régulière de Cholesky* d'une matrice  $A \in \mathcal{M}_n(\mathbb{C})$  est une factorisation  $A = BB^*$  où  $B$  est une matrice triangulaire inférieure inversible. Si les coefficients diagonaux de  $B$  sont positifs, on parle alors d'une *factorisation positive de Cholesky*.

**Théorème 3.1.7** (Factorisation de Cholesky ★★★★★). La matrice  $A \in \mathcal{M}_n(\mathbb{C})$  admet une factorisation régulière de Cholesky si et seulement si la matrice  $A$  est hermitienne définie positive. Dans ce cas, elle admet une unique factorisation positive.

*Preuve.*  $\boxed{\Rightarrow}$  Soit  $A \in \mathcal{M}_n(\mathbb{C})$  admettant une factorisation régulière de Cholesky  $A = BB^*$  avec  $B$  est une matrice triangulaire inférieure inversible.  
La matrice  $A$  est hermitienne car

$$A^* = (BB^*)^* = (B^*)^*B^* = BB^* = A.$$

Soit  $x \in \mathbb{C}^n \setminus \{0\}$ , on a

$$\langle Ax, x \rangle = \langle BB^*x, x \rangle = \langle B^*x, B^*x \rangle = \|B^*x\|^2 > 0$$

car  $B^*x \neq 0$  ( $B^*$  inversible et  $x \neq 0$ ). Donc la matrice  $A$  est bien hermitienne définie positive.

$\boxed{\Leftarrow}$  Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne définie positive.

D'après le Corollaire 3.1.2, page 84, il existe alors une matrice  $L \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure à diagonale unité et une matrice  $D \in \mathcal{M}_n(\mathbb{R})$  diagonale à coefficient strictement positifs telles que

$$A = LDL^*.$$

On note  $H \in \mathcal{M}_n(\mathbb{R})$  une matrice diagonale inversible vérifiant  $H^2 = D$  (i.e.  $H_{i,i} = \pm\sqrt{D_{i,i}} \neq 0$ ,  $\forall i \in \llbracket 1, n \rrbracket$ ). On a alors

$$A = LHHL^* = (LH)(LH)^*$$

En posant  $B = LH$ , la matrice  $B$  est bien triangulaire inférieure inversible car produit d'une matrice triangulaire inférieure inversible par une matrice diagonale inversible et on a  $A = BB^*$ .

Montrons qu'une factorisation positive de Cholesky est unique.

Soient  $B_1$  et  $B_2$  deux factorisations positives de la matrice  $A$ , on a donc

$$A = B_1B_1^* = B_2B_2^*.$$

En multipliant à gauche par  $B_2^{-1}$  et à droite par  $(B_1^*)^{-1}$  cette équation on obtient

$$B_2^{-1}B_1 = B_2^*(B_1^*)^{-1} = B_2^*(B_1^{-1})^* = (B_1^{-1}B_2)^*$$

En notant  $G = B_2^{-1}B_1$ , on tire de l'équation précédente

$$G = (G^{-1})^*. \quad (3.20)$$

On déduit de la (voir Proposition B.2.8, page 225), que l'inverse d'une matrice triangulaire inférieure à coefficients diagonaux réels strictement positifs est aussi une matrice triangulaire inférieure à coefficients diagonaux réels strictement positifs. De la (voir Proposition B.2.7, page 225), on obtient que le produit de matrices triangulaires inférieures à coefficients diagonaux réels strictement positifs reste triangulaire inférieure à coefficients diagonaux réels strictement positifs, on en déduit que les matrices  $G = B_2^{-1}B_1$  et  $G^{-1} = B_1^{-1}B_2$  sont triangulaires inférieures à coefficients diagonaux réels strictement positifs. Or l'équation (3.20) identifie la matrice triangulaire inférieure  $G$  à la matrice triangulaire supérieure  $(G^{-1})^*$  : ce sont donc des matrices diagonales à coefficients diagonaux réels strictement positifs et on a alors  $(G^{-1})^* = G^{-1}$ . De l'équation (3.20), on obtient alors  $G = G^{-1}$ , c'est à dire  $G = I = B_2^{-1}B_1$  et donc  $B_1 = B_2$ .  $\square$

### Résolution d'un système linéaire par la factorisation de Cholesky

Pour commencer, avec la factorisation de Cholesky point de salut sans matrice hermitienne définie positive!

 N'utiliser la factorisation de Cholesky pour la résolution d'un système linéaire que si la matrice du système est hermitienne définie positive.

Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne définie positive et  $b \in \mathbb{C}^n$ . Grâce au théorème 3.1.7, on obtient :

Trouver  $x \in \mathbb{C}^n$  tel que

$$Ax = b. \quad (3.21)$$

est équivalent à

Trouver  $\mathbf{x} \in \mathbb{C}^n$  solution de

$$\mathbb{B}^* \mathbf{x} = \mathbf{y} \quad (3.22)$$

avec  $\mathbb{B}$  la matrice de factorisation positive de Cholesky de la matrice  $\mathbb{A}$  avec  $\mathbf{y} \in \mathbb{C}^n$  solution de

$$\mathbb{B} \mathbf{y} = \mathbf{b}. \quad (3.23)$$

On est donc ramené à

**Algorithm 3.14** Algorithme de base permettant de résoudre, par une factorisation de Cholesky positive, le système linéaire

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

où  $\mathbb{A}$  une matrice de  $\mathcal{M}_n(\mathbb{C})$  hermitienne définie positive et  $\mathbf{b} \in \mathbb{C}^n$ .

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{C})$  hermitienne définie positive,  
 $\mathbf{b}$  : vecteur de  $\mathbb{C}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{C}^n$ .

- 1: Trouver la factorisation positive de Cholesky  $\mathbb{B}$  de la matrice  $\mathbb{A}$ ,
- 2: résoudre le système triangulaire inférieur  $\mathbb{B} \mathbf{y} = \mathbf{b}$ ,
- 3: résoudre le système triangulaire supérieur  $\mathbb{B}^* \mathbf{x} = \mathbf{y}$ .

Ceci permet donc de découper le problème initial en trois sous-problèmes plus simples. De plus, ceux-ci peuvent se traiter de manière indépendante.

**Algorithm 3.15** Fonction `RSLCholesky` permettant de résoudre, par une factorisation de Cholesky positive, le système linéaire

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

où  $\mathbb{A}$  une matrice hermitienne de  $\mathcal{M}_n(\mathbb{C})$  définie positive et  $\mathbf{b} \in \mathbb{C}^n$ .

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{C})$  hermitienne définie positive,  
 $\mathbf{b}$  : vecteur de  $\mathbb{C}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{C}^n$ .

- 1: **Fonction**  $\mathbf{x} \leftarrow \text{RSLCholesky}(\mathbb{A}, \mathbf{b})$
- 2:  $\mathbb{B} \leftarrow \text{Cholesky}(\mathbb{A})$  ▷ Factorisation positive de Cholesky
- 3:  $\mathbf{y} \leftarrow \text{RSLTriInf}(\mathbb{B}, \mathbf{b})$  ▷ Résolution du système  $\mathbb{B} \mathbf{y} = \mathbf{b}$
- 4:  $\mathbb{U} \leftarrow \text{MatAdjointe}(\mathbb{B})$  ▷ Calcul de la matrice adjointe de  $\mathbb{B}$
- 5:  $\mathbf{x} \leftarrow \text{RSLTriSup}(\mathbb{U}, \mathbf{y})$  ▷ Résolution du système  $\mathbb{B}^* \mathbf{x} = \mathbf{y}$
- 6: **Fin**

Il nous faut donc écrire la fonction `Cholesky` (les deux fonctions `RSLTriInf` et `RSLTriSup` ayant déjà été écrites et la fonction `MatAdjointe` étant simple à écrire).

### Factorisation positive de Cholesky : écriture de l'algorithme

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne définie positive. D'après le Théorème 3.1.7, il existe une unique matrice  $\mathbb{B} \in \mathcal{M}_n(\mathbb{C})$  triangulaire inférieure avec  $B_{i,i} \in \mathbb{R}^{+*}$ ,  $\forall i \in \llbracket 1, n \rrbracket$ , telle que

$$\mathbb{A} = \mathbb{B} \mathbb{B}^* \quad (3.24)$$

c'est à dire

$$\begin{pmatrix} A_{1,1} & \dots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} B_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ B_{n,1} & \dots & \dots & B_{n,n} \end{pmatrix} \begin{pmatrix} \overline{B_{1,1}} & \dots & \dots & \overline{B_{n,1}} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \overline{B_{n,n}} \end{pmatrix}. \quad (3.25)$$

Pour déterminer la matrice  $\mathbb{B}$ , on commence par calculer  $B_{1,1}$  (la 1ère ligne de  $\mathbb{B}$  est donc déterminée) ce qui nous permet de calculer la 1ère colonne de  $\mathbb{B}$ .

Ensuite, on calcule  $B_{2,2}$  (la 2ème ligne de  $\mathbb{B}$  est donc déterminée) ce qui nous permet de calculer la 2ème colonne de  $\mathbb{B}$ . Etc ...

On écrit en détail les raffinements successifs permettant d'aboutir à l'algorithme final de telle manière que le passage entre deux raffinements successifs soit le plus simple possible.

**Algorithme 3.16**  $\mathcal{R}_0$

1: Calculer la matrice  $\mathbb{B}$

**Algorithme 3.16**  $\mathcal{R}_1$

1: **Pour**  $i \leftarrow 1$  à  $n$  **faire**  
 2: Calculer  $B_{i,i}$ , connaissant les  $i - 1$  premières colonnes de  $\mathbb{B}$ .  
 3: Calculer la  $i^{\text{ème}}$  colonne de  $\mathbb{B}$ .  
 4: **Fin**

Pour calculer  $B_{i,i}$ , connaissant les  $i - 1$  premières colonnes de  $\mathbb{B}$ , on utilise (3.24) :

$$A_{i,i} = \sum_{j=1}^n B_{i,j} (\mathbb{B}^*)_{j,i} = \sum_{j=1}^n |B_{i,j}|^2,$$

et comme  $\mathbb{B}$  est triangulaire inférieure on obtient

$$A_{i,i} = \sum_{j=1}^i |B_{i,j}|^2 = \sum_{j=1}^{i-1} |B_{i,j}|^2 + |B_{i,i}|^2.$$

On a donc

$$B_{i,i} = \left( A_{i,i} - \sum_{j=1}^{i-1} |B_{i,j}|^2 \right)^{1/2}, \quad \forall i \in \llbracket 1, n \rrbracket, \quad (3.26)$$

Comme les  $i - 1$  premières colonnes de  $\mathbb{B}$  ont déjà été calculées,  $B_{i,i}$  est parfaitement déterminée par la formule précédente.

**Remarque 3.1.3.** Les hypothèses sur la matrice  $\mathbb{A}$  permettent d'affirmer que,  $\forall i \in \llbracket 1, n \rrbracket$ ,  $A_{i,i} - \sum_{j=1}^{i-1} |B_{i,j}|^2 > 0$  et donc  $B_{i,i} > 0$ .

Pour calculer la  $i^{\text{ème}}$  colonne de  $\mathbb{B}$ , il suffit de déterminer  $B_{j,i}$ ,  $\forall j \in \llbracket i + 1, n \rrbracket$ . Pour cela on utilise (3.24)

$$A_{j,i} = \sum_{k=1}^n B_{j,k} (\mathbb{B}^*)_{k,i} = \sum_{k=1}^n B_{j,k} \overline{B_{i,k}}, \quad \forall j \in \llbracket i + 1, n \rrbracket$$

Comme  $\mathbb{L}$  est triangulaire inférieure on obtient

$$A_{j,i} = \sum_{k=1}^i B_{j,k} \overline{B_{i,k}} = \sum_{k=1}^{i-1} B_{j,k} \overline{B_{i,k}} + B_{j,i} \overline{B_{i,i}}, \quad \forall j \in \llbracket i + 1, n \rrbracket$$

Or  $B_{i,i} > 0$  vient d'être calculé et les  $i - 1$  premières colonnes de  $\mathbb{B}$  ont déjà été calculées, ce qui donne

$$B_{j,i} = \frac{1}{B_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} B_{j,k} \overline{B_{i,k}} \right), \quad \forall j \in \llbracket i + 1, n \rrbracket \quad (3.27)$$

$$B_{j,i} = 0, \quad \forall j \in \llbracket 1, i - 1 \rrbracket. \quad (3.28)$$

Avec (3.26) et (3.27), on a

Algorithme 3.16  $\mathcal{R}_1$ 

- 1: **Pour**  $i \leftarrow 1$  à  $n$  **faire**
- 2: Calculer  $B_{i,i}$ , connaissant les  $i - 1$  premières colonnes de  $\mathbb{B}$ .
- 3: Calculer la  $i^{\text{ème}}$  colonne de  $\mathbb{B}$ .
- 4: **Fin**

Algorithme 3.16  $\mathcal{R}_2$ 

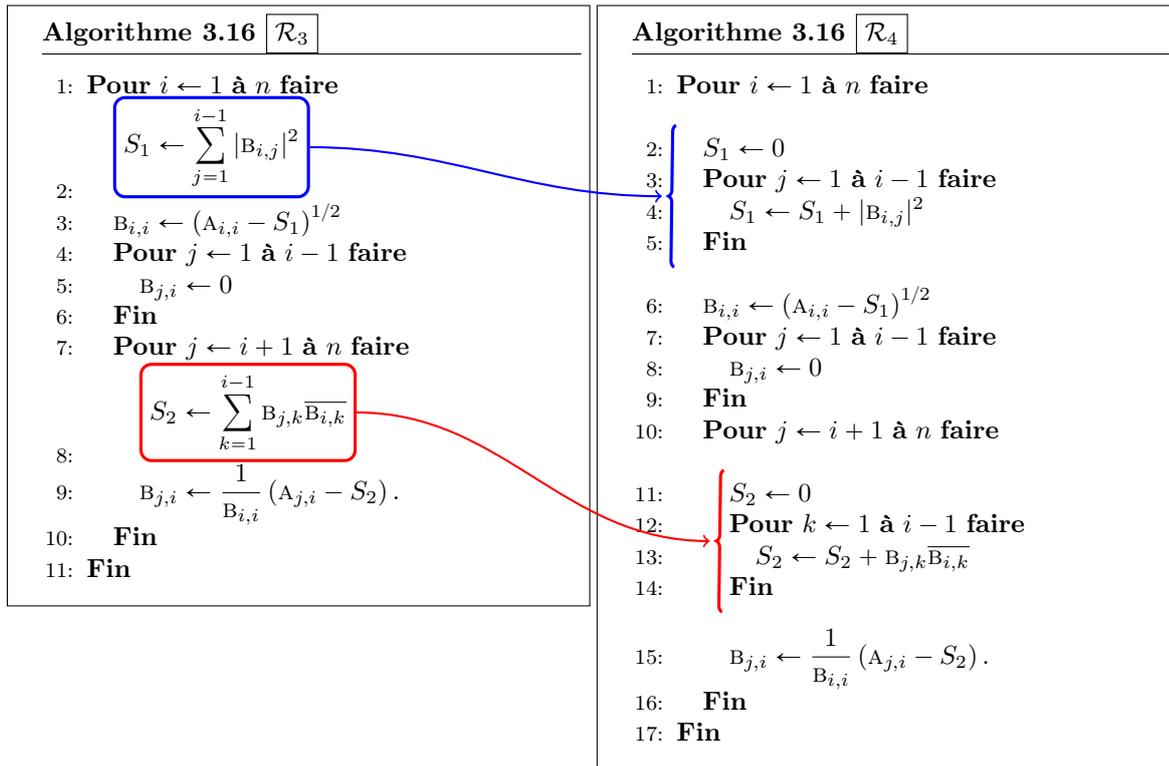
- 1: **Pour**  $i \leftarrow 1$  à  $n$  **faire**
- 2:  $B_{i,i} \leftarrow \left( A_{i,i} - \sum_{j=1}^{i-1} |B_{i,j}|^2 \right)^{1/2}$
- 3: **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**
- 4:  $B_{j,i} \leftarrow 0$
- 5: **Fin**
- 6: **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**
- 7:  $B_{j,i} \leftarrow \frac{1}{B_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} B_{j,k} \overline{B_{i,k}} \right)$ .
- 8: **Fin**
- 9: **Fin**

Algorithme 3.16  $\mathcal{R}_2$ 

- 1: **Pour**  $i \leftarrow 1$  à  $n$  **faire**
- 2:  $B_{i,i} \leftarrow \left( A_{i,i} - \sum_{j=1}^{i-1} |B_{i,j}|^2 \right)^{1/2}$
- 3: **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**
- 4:  $B_{j,i} \leftarrow 0$
- 5: **Fin**
- 6: **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**
- 7:  $B_{j,i} \leftarrow \frac{1}{B_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} B_{j,k} \overline{B_{i,k}} \right)$ .
- 8: **Fin**
- 9: **Fin**

Algorithme 3.16  $\mathcal{R}_3$ 

- 1: **Pour**  $i \leftarrow 1$  à  $n$  **faire**
- 2:  $S_1 \leftarrow \sum_{j=1}^{i-1} |B_{i,j}|^2$
- 3:  $B_{i,i} \leftarrow (A_{i,i} - S_1)^{1/2}$
- 4: **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**
- 5:  $B_{j,i} \leftarrow 0$
- 6: **Fin**
- 7: **Pour**  $j \leftarrow i + 1$  à  $n$  **faire**
- 8:  $S_2 \leftarrow \sum_{k=1}^{i-1} B_{j,k} \overline{B_{i,k}}$
- 9:  $B_{j,i} \leftarrow \frac{1}{B_{i,i}} (A_{j,i} - S_2)$ .
- 10: **Fin**
- 11: **Fin**



On obtient alors l'algorithme final

**Algorithme 3.16** Fonction **Cholesky** permettant de calculer la matrice  $\mathbb{B}$ , dites matrice de factorisation positive de Cholesky associée à la matrice  $\mathbb{A}$ , telle que  $\mathbb{A} = \mathbb{B}\mathbb{B}^*$ .

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{C})$  hermitienne définie positive.

**Résultat :**  $\mathbb{B}$  : matrice de  $\mathcal{M}_n(\mathbb{C})$  triangulaire inférieure  
avec  $B(i, i) > 0, \forall i \in \llbracket 1, n \rrbracket$

```

1: Fonction  $\mathbb{B} \leftarrow \text{Cholesky}(\mathbb{A})$ 
2: Pour  $i \leftarrow 1$  à  $n$  faire
3:    $S_1 \leftarrow 0$ 
4:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
5:      $S_1 \leftarrow S_1 + |B(i, j)|^2$ 
6:   Fin
7:    $B(i, i) \leftarrow \text{sqrt}(A(i, i) - S_1)$ 
8:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
9:      $B(j, i) \leftarrow 0$ 
10:  Fin
11:  Pour  $j \leftarrow i + 1$  à  $n$  faire
12:     $S_2 \leftarrow 0$ 
13:    Pour  $k \leftarrow 1$  à  $i - 1$  faire
14:       $S_2 \leftarrow S_2 + B(j, k) * \overline{B(i, k)}$ 
15:    Fin
16:     $B(j, i) \leftarrow (A(j, i) - S_2) / B(i, i)$ 
17:  Fin
18: Fin
19: Fin

```

## 3.1.7 Factorisation QR

## La transformation de Householder

**Définition 3.1.2** (Matrice élémentaire de Householder). Soit  $\mathbf{u} \in \mathbb{C}^n$  tel que  $\|\mathbf{u}\|_2 = 1$ . On appelle *matrice élémentaire de Householder* la matrice  $\mathbb{H}(\mathbf{u}) \in \mathcal{M}_n(\mathbb{C})$  définie par

$$\mathbb{H}(\mathbf{u}) = \mathbb{I} - 2\mathbf{u}\mathbf{u}^*. \quad (3.29)$$

**Propriété 3.1.1.** Toute matrice élémentaire de Householder est hermitienne et unitaire.

*Preuve.* Pour simplifier, on note  $\mathbb{H} = \mathbb{H}(\mathbf{u})$ . Cette matrice est hermitienne car

$$\mathbb{H}^* = (\mathbb{I} - 2\mathbf{u}\mathbf{u}^*)^* = \mathbb{I} - 2(\mathbf{u}\mathbf{u}^*)^* = \mathbb{I} - 2\mathbf{u}\mathbf{u}^* = \mathbb{H}.$$

Montrons qu'elle est unitaire (i.e.  $\mathbb{H}^*\mathbb{H} = \mathbb{I}$ ). On a

$$\begin{aligned} \mathbb{H}^*\mathbb{H} &= \mathbb{H}\mathbb{H} = (\mathbb{I} - 2\mathbf{u}\mathbf{u}^*)(\mathbb{I} - 2\mathbf{u}\mathbf{u}^*) \\ &= \mathbb{I} - 4\mathbf{u}\mathbf{u}^* + 4\mathbf{u}\mathbf{u}^*\mathbf{u}\mathbf{u}^*. \end{aligned}$$

Or on a  $\mathbf{u}^*\mathbf{u} = \|\mathbf{u}\|_2^2 = 1$  par hypothèse et donc

$$\mathbb{H}^*\mathbb{H} = \mathbb{I} - 4\mathbf{u}\mathbf{u}^* + 4\mathbf{u}(\mathbf{u}^*\mathbf{u})\mathbf{u}^* = \mathbb{I}.$$

□

**Propriété 3.1.2.** Soient  $\mathbf{x} \in \mathbb{K}^n$  et  $\mathbf{u} \in \mathbb{K}^n$ ,  $\|\mathbf{u}\|_2 = 1$ . On note  $\mathbf{x}_{\parallel} = \text{proj}_{\mathbf{u}}(\mathbf{x}) \stackrel{\text{def}}{=} \langle \mathbf{u}, \mathbf{x} \rangle \mathbf{u}$  et  $\mathbf{x}_{\perp} = \mathbf{x} - \mathbf{x}_{\parallel}$ . On a alors

$$\mathbb{H}(\mathbf{u})(\mathbf{x}_{\perp} + \mathbf{x}_{\parallel}) = \mathbf{x}_{\perp} - \mathbf{x}_{\parallel}. \quad (3.30)$$

et

$$\mathbb{H}(\mathbf{u})\mathbf{x} = \mathbf{x}, \quad \text{si } \langle \mathbf{x}, \mathbf{u} \rangle = 0. \quad (3.31)$$

*Preuve.* On note que par construction  $\langle \mathbf{u}, \mathbf{x}_{\perp} \rangle = 0$ . En effet, on a

$$\begin{aligned} \langle \mathbf{u}, \mathbf{x}_{\perp} \rangle &= \langle \mathbf{u}, \mathbf{x} - \mathbf{x}_{\parallel} \rangle \\ &= \langle \mathbf{u}, \mathbf{x} \rangle - \langle \mathbf{u}, \mathbf{x}_{\parallel} \rangle \\ &= \langle \mathbf{u}, \mathbf{x} \rangle - \langle \mathbf{u}, \langle \mathbf{u}, \mathbf{x} \rangle \mathbf{u} \rangle \\ &= \langle \mathbf{u}, \mathbf{x} \rangle - \langle \mathbf{u}, \mathbf{x} \rangle \langle \mathbf{u}, \mathbf{u} \rangle \\ &= 0 \quad \text{car } \langle \mathbf{u}, \mathbf{u} \rangle = \|\mathbf{u}\|_2^2 = 1. \end{aligned}$$

On a alors

$$\begin{aligned} \mathbb{H}(\mathbf{u})(\mathbf{x}_{\perp} + \mathbf{x}_{\parallel}) &= (\mathbb{I} - 2\mathbf{u}\mathbf{u}^*)(\mathbf{x}_{\perp} + \mathbf{x}_{\parallel}) = \mathbf{x}_{\perp} + \mathbf{x}_{\parallel} - 2\mathbf{u} \underbrace{\mathbf{u}^*\mathbf{x}_{\perp}}_{=0} - 2\mathbf{u}\mathbf{u}^*\mathbf{x}_{\parallel} \\ &= \mathbf{x}_{\perp} + \mathbf{x}_{\parallel} - 2\mathbf{u}\mathbf{u}^*(\langle \mathbf{u}, \mathbf{x} \rangle \mathbf{u}) = \mathbf{x}_{\perp} + \mathbf{x}_{\parallel} - 2\langle \mathbf{u}, \mathbf{x} \rangle \mathbf{u} \underbrace{\mathbf{u}^*\mathbf{u}}_{=1} \\ &= \mathbf{x}_{\perp} + \mathbf{x}_{\parallel} - 2\mathbf{x}_{\parallel} \\ &= \mathbf{x}_{\perp} - \mathbf{x}_{\parallel}. \end{aligned}$$

Si  $\langle \mathbf{x}, \mathbf{u} \rangle = 0$  alors  $\mathbf{x}_{\parallel} = 0$  et  $\mathbf{x} = \mathbf{x}_{\perp}$ . □

**Théorème 3.1.8.** Soient  $\mathbf{a}, \mathbf{b}$  deux vecteurs non nuls et non colinéaires de  $\mathbb{C}^n$  avec  $\|\mathbf{b}\|_2 = 1$ . Soit

$\alpha \in \mathbb{C}$  tel que  $|\alpha| = \|\mathbf{a}\|_2$  et  $\arg \alpha = -\arg \langle \mathbf{a}, \mathbf{b} \rangle \pmod{\pi}$ . On a alors

$$\mathbb{H} \left( \frac{\mathbf{a} - \alpha \mathbf{b}}{\|\mathbf{a} - \alpha \mathbf{b}\|_2} \right) \mathbf{a} = \alpha \mathbf{b}. \quad (3.32)$$

Preuve. (voir Exercice 3.1.8, page 91)  $\square$

### EXERCICE 3.1.8

Soient  $\mathbf{a}$  et  $\mathbf{b}$  deux vecteurs non colinéaires de  $\mathbb{C}^n$  avec  $\|\mathbf{b}\|_2 = 1$ . On va chercher  $\alpha \in \mathbb{C}$  et  $\mathbf{u} \in \mathbb{C}^n$ ,  $\|\mathbf{u}\|_2 = 1$ , vérifiant

$$\mathbb{H}(\mathbf{u})\mathbf{a} = \alpha \mathbf{b}. \quad (3.1)$$

Q. 1

a. Montrer que si  $\alpha$  vérifie (3.1) alors  $|\alpha| = \|\mathbf{a}\|_2$ .

b. Montrer que si  $\arg \alpha = -\arg \langle \mathbf{a}, \mathbf{b} \rangle \pmod{\pi}$  alors  $\alpha \langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}$ .

Q. 2

Soient  $\alpha$  et  $\mathbf{u}$  vérifiant (3.1).

a. Montrer que

$$|\langle \mathbf{u}, \mathbf{a} \rangle|^2 = \frac{\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle}{2} \quad (3.2)$$

b. Montrer que si  $\arg \alpha = -\arg \langle \mathbf{a}, \mathbf{b} \rangle \pmod{\pi}$  alors  $\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}^{*+}$ .

c. En déduire que

$$\mathbf{u} = \frac{1}{2\lambda}(\mathbf{a} - \alpha \mathbf{b}), \quad \text{avec } \lambda = \pm \left( \frac{\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle}{2} \right)^{1/2} \quad (3.3)$$

### Correction 3.1.8

R. 1

Dans la suite de l'exercice, on pose  $\mathbb{H} = \mathbb{H}(\mathbf{u})$  pour alléger les notations.

a. On a

$$\begin{aligned} \|\mathbf{a}\|_2^2 &= \langle \mathbf{a}, \mathbf{a} \rangle = \langle \mathbb{H}^* \mathbb{H} \mathbf{a}, \mathbf{a} \rangle \quad \text{car } \mathbb{H} \text{ unitaire} \\ &= \langle \mathbb{H} \mathbf{a}, \mathbb{H} \mathbf{a} \rangle \quad \text{par définition du produit scalaire} \\ &= \|\mathbb{H} \mathbf{a}\|_2^2 = \|\alpha \mathbf{b}\|_2^2 = |\alpha|^2 \|\mathbf{b}\|_2^2 = |\alpha|^2. \end{aligned}$$

b. On a par définition de l'argument  $\alpha = |\alpha|e^{i \arg \alpha}$  et  $\langle \mathbf{a}, \mathbf{b} \rangle = |\langle \mathbf{a}, \mathbf{b} \rangle|e^{i \arg \langle \mathbf{a}, \mathbf{b} \rangle}$  ce qui donne

$$\alpha \langle \mathbf{a}, \mathbf{b} \rangle = |\alpha| |\langle \mathbf{a}, \mathbf{b} \rangle| e^{i(\arg \alpha + \arg \langle \mathbf{a}, \mathbf{b} \rangle)} \quad (R8.4)$$

et donc  $\alpha \langle \mathbf{a}, \mathbf{b} \rangle$  est réel si  $\arg \alpha + \arg \langle \mathbf{a}, \mathbf{b} \rangle = 0 \pmod{\pi}$ .

R. 2

a. On a

$$\begin{aligned} \mathbb{H}(\mathbf{u})\mathbf{a} = \alpha \mathbf{b} &\iff (\mathbb{I} - 2\mathbf{u}\mathbf{u}^*)\mathbf{a} = \alpha \mathbf{b} \\ &\iff \mathbf{a} - 2\mathbf{u}(\mathbf{u}^*\mathbf{a}) = \alpha \mathbf{b} \end{aligned}$$

et donc

$$\mathbf{a} - 2\langle \mathbf{u}, \mathbf{a} \rangle \mathbf{u} = \alpha \mathbf{b} \quad (R8.5)$$

En effectuant le produit scalaire avec  $\mathbf{a}$  de cette dernière équation, on obtient

$$\langle \mathbf{a}, \mathbf{a} \rangle - 2\langle \mathbf{u}, \mathbf{a} \rangle \langle \mathbf{a}, \mathbf{u} \rangle = \alpha \langle \mathbf{a}, \mathbf{b} \rangle$$

ce qui prouve (3.2).

b. On a montré en Q.1 que  $\alpha \langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}$  et donc  $\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}$ . Il reste donc à montrer que  $\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle > 0$ .

- Si  $\arg \alpha = -\arg(\langle \mathbf{a}, \mathbf{b} \rangle) + \pi [2\pi]$ , alors de (R8.4) on obtient  $\alpha \langle \mathbf{a}, \mathbf{b} \rangle \leq 0$  et donc  $\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle \geq \|\mathbf{a}\|_2 > 0$  car  $\mathbf{a} \neq 0$ .
- Si  $\arg \alpha = -\arg(\langle \mathbf{a}, \mathbf{b} \rangle) [2\pi]$ , alors de (R8.4) on obtient  $\alpha \langle \mathbf{a}, \mathbf{b} \rangle \geq 0$ . Comme les vecteurs  $\mathbf{a}$  et  $\mathbf{b}$  ne sont pas colinéaires, on a inégalité stricte dans Cauchy-Schwarz :

$$|\langle \mathbf{a}, \mathbf{b} \rangle| < \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 = \|\mathbf{a}\|_2.$$

On obtient donc

$$0 \leq \alpha \langle \mathbf{a}, \mathbf{b} \rangle \leq |\alpha| |\langle \mathbf{a}, \mathbf{b} \rangle| < |\alpha| \|\mathbf{a}\|_2 = \|\mathbf{a}\|_2^2$$

Attention, dans ce cas  $\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle$  peut-être très petit.

c. De (R8.5), on en déduit immédiatement (3.3).

Vérifions que  $\|\mathbf{u}\|_2 = 1$ . On a

$$\|\mathbf{u}\|_2^2 = \langle \mathbf{u}, \mathbf{u} \rangle = \frac{1}{4|\lambda|^2} \langle \mathbf{a} - \alpha \mathbf{b}, \mathbf{a} - \alpha \mathbf{b} \rangle$$

Or

$$\begin{aligned} \langle \mathbf{a} - \alpha \mathbf{b}, \mathbf{a} - \alpha \mathbf{b} \rangle &= \langle \mathbf{a}, \mathbf{a} \rangle - \bar{\alpha} \langle \mathbf{b}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle + |\alpha|^2 \langle \mathbf{b}, \mathbf{b} \rangle = \|\mathbf{a}\|_2^2 - \bar{\alpha} \langle \mathbf{b}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle + |\alpha|^2 \\ &= 2\|\mathbf{a}\|_2^2 - \bar{\alpha} \langle \mathbf{b}, \mathbf{a} \rangle - \alpha \langle \mathbf{a}, \mathbf{b} \rangle \\ &= 2\|\mathbf{a}\|_2^2 - 2\alpha \langle \mathbf{a}, \mathbf{b} \rangle \quad \text{car } \alpha \langle \mathbf{a}, \mathbf{b} \rangle = \overline{(\alpha \langle \mathbf{a}, \mathbf{b} \rangle)} = \bar{\alpha} \langle \mathbf{b}, \mathbf{a} \rangle \in \mathbb{R} \end{aligned}$$

De plus

$$\begin{aligned} 4|\lambda|^2 &= 2(\langle \mathbf{a}, \mathbf{a} \rangle - \alpha \langle \mathbf{b}, \mathbf{a} \rangle) \in \mathbb{R} \\ &= 2\|\mathbf{a}\|_2^2 - 2\alpha \langle \mathbf{b}, \mathbf{a} \rangle \end{aligned}$$

Fin correction

### EXERCICE 3.1.9

Soient  $\mathbf{a}$  et  $\mathbf{b}$  deux vecteurs non nuls et non colinéaires de  $\mathbb{C}^n$  avec  $\|\mathbf{b}\|_2 = 1$ .

**Q. 1** Ecrire la fonction algorithmique `Householder` permettant de retourner une matrice de Householder  $\mathbb{H}$  et  $\alpha \in \mathbb{C}$  tels que  $\mathbb{H}(\mathbf{u})\mathbf{a} = \alpha \mathbf{b}$ . Le choix du  $\alpha$  est fait par le paramètre  $\delta$  (0 ou 1) de telle sorte que  $\arg \alpha = -\arg(\langle \mathbf{a}, \mathbf{b} \rangle) + \delta\pi$  avec  $|\alpha| = \|\mathbf{a}\|_2$ . Des fonctions comme `dot(a, b)` (produit scalaire de deux vecteurs), `norm(a)` (norme 2 d'un vecteur), `arg(z)` (argument d'un nombre complexe), `matprod(A, B)` (produit de deux matrices), `ctranspose(A)` (adjoint d'une matrice), ... pourront être utilisées

**Q. 2** Proposer un programme permettant de tester cette fonction. On pourra utiliser la fonction `vecrand(n)` retournant un vecteur aléatoire de  $\mathbb{C}^n$ , les parties réelles et imaginaires de chacune de ses composantes étant dans  $]0, 1[$  (loi uniforme).

**Q. 3** Proposer un programme permettant de vérifier que  $\delta = 1$  est le "meilleur" choix.

### Correction 3.1.9

**R. 1**

Soient  $\mathbf{a}$  et  $\mathbf{b}$  deux vecteurs non nuls et non colinéaires de  $\mathbb{C}^n$ .

Les données du problème sont  $\mathbf{a}$ ,  $\mathbf{b}$  et  $\delta$ . On veut calculer  $\alpha$  et la matrice  $\mathbb{H}(\mathbf{u})$ .

---

**Algorithme 3.17** Calcul du  $\alpha$  et de la matrice de Householder  $\mathbb{H}(\mathbf{u})$  telle que  $\mathbb{H}(\mathbf{u})\mathbf{a} = \alpha\mathbf{b}$ .

---

**Données :**  $\mathbf{a}, \mathbf{b}$  : deux vecteurs de  $\mathbb{C}^n$  non nuls et non colinéaires.  
 $\delta$  : 0 ou 1, permet de déterminer  $\alpha$ .

**Résultat :**  $\mathbb{H}$  : matrice de Householder dans  $\mathcal{M}_n(\mathbb{C})$ ,  
 $\alpha$  : nombre complexe, de module  $\|\mathbf{a}\|_2$  et d'argument  $-\arg(\langle \mathbf{a}, \mathbf{b} \rangle) + \delta\pi$ .

```

1: Fonction [ $\mathbb{H}, \alpha$ ]  $\leftarrow$  Householder(  $\mathbf{a}, \mathbf{b}, \delta$  )
2:  $ab \leftarrow \text{dot}(\mathbf{a}, \mathbf{b})$   $\triangleright \text{dot}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^n \bar{a}_i b_i$ 
3:  $\alpha \leftarrow \text{norm}(\mathbf{a}) * \exp(i * (\delta * \pi - \arg(ab)))$ 
4:  $\mathbf{u} \leftarrow \mathbf{a} - \alpha * \mathbf{b}$ 
5:  $\mathbf{u} \leftarrow \mathbf{u} / \text{norm}(\mathbf{u})$ 
6:  $\mathbb{H} \leftarrow \text{eye}(n) - 2 * \text{matprod}(\mathbf{u}, \text{ctranspose}(\mathbf{u}))$ 
7: Fin

```

---

**R. 2**

```

1:  $n \leftarrow 100$ 
2:  $\mathbf{a} \leftarrow \text{vecrand}(n)$ 
3:  $\mathbf{b} \leftarrow \text{vecrand}(n)$ 
4:  $\mathbf{b} \leftarrow \mathbf{b} / \text{norm}(\mathbf{b}, 2)$ 
5: [ $\mathbb{H}, \alpha$ ]  $\leftarrow$  Householder( $\mathbf{a}, \mathbf{b}, 0$ )
6:  $\text{error} \leftarrow \text{norm}(\mathbb{H} * \mathbf{a} - \alpha * \mathbf{b}, 2)$ 

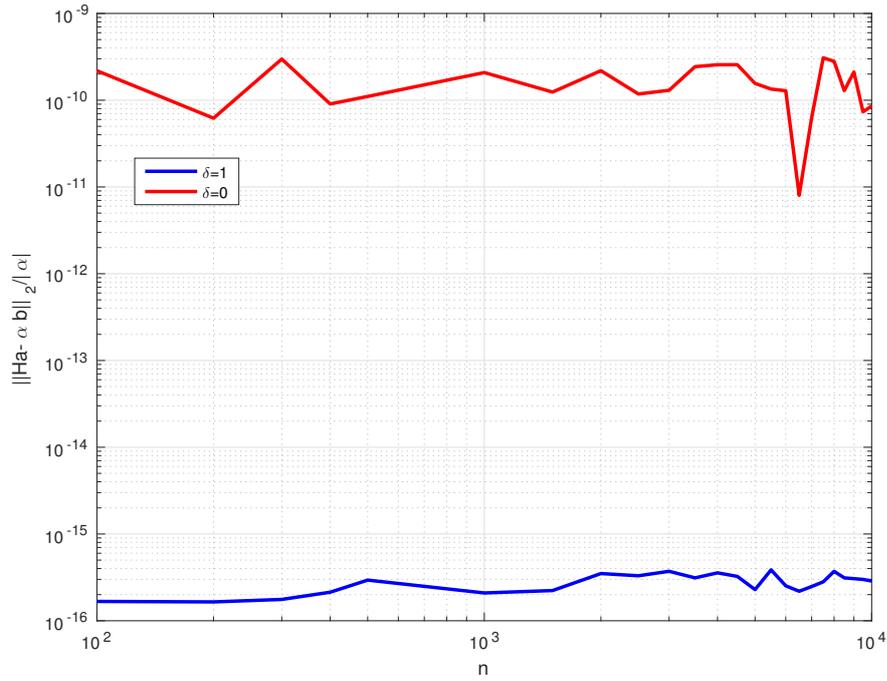
```

**R. 3**

```

1:  $n \leftarrow 100$ 
2:  $\mathbf{a} \leftarrow \text{vecrand}(n)$ 
3:  $\mathbf{b} \leftarrow \mathbf{a} + 1e - 6 * \text{vecrand}(n)$ 
4:  $\mathbf{b} \leftarrow \mathbf{b} / \text{norm}(\mathbf{b}, 2)$ 
5: [ $\mathbb{H}_1, \alpha_1$ ]  $\leftarrow$  Householder( $\mathbf{a}, \mathbf{b}, 1$ )
6: [ $\mathbb{H}_0, \alpha_0$ ]  $\leftarrow$  Householder( $\mathbf{a}, \mathbf{b}, 0$ )
7:  $\text{error}_0 \leftarrow \text{norm}(\mathbb{H}_0 * \mathbf{a} - \alpha_0 * \mathbf{b}, 2) / (1 + \text{abs}(\alpha_0))$ 
8:  $\text{error}_1 \leftarrow \text{norm}(\mathbb{H}_1 * \mathbf{a} - \alpha_1 * \mathbf{b}, 2) / (1 + \text{abs}(\alpha_1))$ 

```

Figure 3.2: Choix de  $\alpha$  dans Householder : erreur relative en norme  $L_2$ 

~~~~~Fin correction~~~~~

**NOTE**

Si l'on souhaite calculer le produit matrice-vecteur  $\mathbb{H}(\mathbf{u})\mathbf{x}$ , il n'est pas nécessaire de calculer explicitement la matrice  $\mathbb{H}(\mathbf{u})$ . En effet, on pose  $\mathbf{v} = 2\lambda\mathbf{u} = \mathbf{a} - \alpha\mathbf{b}$  et  $\beta = 2\lambda^2 = \|\mathbf{a}\|_2^2 - \alpha\langle\mathbf{a}, \mathbf{b}\rangle > 0$ . On choisit  $\alpha$  de manière à maximiser  $\beta$  pour éviter une division par un nombre trop petit. On prend donc

$$\alpha = \|\mathbf{a}\|_2 e^{i(\pi - \arg(\langle\mathbf{a}, \mathbf{b}\rangle))}$$

On obtient alors

$$\mathbb{H}(\mathbf{u})\mathbf{x} = \mathbf{x} - \frac{1}{\beta}\mathbf{v}\mathbf{v}^*\mathbf{x} = \mathbf{x} - \frac{\langle\mathbf{v}, \mathbf{x}\rangle}{\beta}\mathbf{v}. \quad (3.33)$$

**Corollaire 3.1.3.** Soit  $\mathbf{a} \in \mathbb{C}^n$  avec  $a_1 \neq 0$  et  $\exists j \in \llbracket 2, n \rrbracket$  tel que  $a_j \neq 0$ . Soient  $\theta = \arg a_1$  et

$$\mathbf{u}_{\pm} = \frac{\mathbf{a} \pm \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1}{\|\mathbf{a} \pm \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1\|}$$

Alors

$$\mathbb{H}(\mathbf{u}_{\pm})\mathbf{a} = \mp \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1 \quad (3.34)$$

où  $\mathbf{e}_1$  désigne le premier vecteur de la base canonique de  $\mathbb{C}^n$ .

*Preuve.* On va utiliser le Théorème 3.1.8.

On pose  $\alpha = \pm \|\mathbf{a}\|_2 e^{i\theta}$  et  $\mathbf{b} = \mathbf{e}_1$ . Comme  $\arg(\langle\mathbf{a}, \mathbf{b}\rangle) = \arg \bar{a}_1 = -\arg a_1 = -\theta$ , on a  $\arg \alpha = \theta [\pi] = -\arg(\langle\mathbf{a}, \mathbf{b}\rangle) [\pi]$ .

Les autres hypothèses du Théorème 3.1.8 sont vérifiées puisque le vecteur  $\mathbf{a}$  n'est pas colinéaire à  $\mathbf{e}_1$  et que  $\|\mathbf{e}_1\|_2 = 1$ . On a donc

$$\mathbb{H}\left(\frac{\mathbf{a} \pm \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1}{\|\mathbf{a} \pm \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1\|}\right)\mathbf{a} = \mp \|\mathbf{a}\|_2 e^{i\theta} \mathbf{e}_1.$$

□

Sur le même principe que l'écriture algébrique de la méthode de Gauss, nous allons transformer la matrice  $A \in \mathcal{M}_n(\mathbb{K})$  en une matrice triangulaire supérieure à l'aide de matrices de Householder. On a le théorème

**Théorème 3.1.9.** Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice. Il existe une matrice unitaire  $Q \in \mathcal{M}_n(\mathbb{C})$  et une matrice triangulaire supérieure  $R \in \mathcal{M}_n(\mathbb{C})$  telles que

$$A = QR. \quad (3.35)$$

Si  $A$  est réelle alors  $Q$  et  $R$  sont aussi réelles et l'on peut choisir  $Q$  de telle sorte que les coefficients diagonaux de  $R$  soient positifs. De plus, si  $A$  est inversible alors la factorisation est unique.

### EXERCICE 3.1.10

Soit  $B \in \mathcal{M}_{m+n}(\mathbb{K})$  la matrice bloc

$$B = \left( \begin{array}{c|c} B_{1,1} & B_{1,2} \\ \hline 0 & S \end{array} \right)$$

où  $B_{1,1} \in \mathcal{M}_m(\mathbb{K})$  et  $S \in \mathcal{M}_n(\mathbb{K})$ . On note  $\mathbf{s} \in \mathbb{K}^n$  le premier vecteur colonne de  $S$  et on suppose que  $\mathbf{s} \neq 0$  et  $\mathbf{s}$  non colinéaire à  $\mathbf{e}_1^n$  premier vecteur de la base canonique de  $\mathbb{K}^n$ .

Q. 1

a. Montrer qu'il existe une matrice de Householder  $H = H(\mathbf{u}) \in \mathcal{M}_n(\mathbb{K})$  et  $\alpha \in \mathbb{K}^*$  tel que

$$HS = \left( \begin{array}{c|ccc} \pm\alpha & \bullet & \cdots & \bullet \\ \hline 0 & \bullet & \cdots & \bullet \\ \vdots & \vdots & & \vdots \\ 0 & \bullet & \cdots & \bullet \end{array} \right).$$

b. On note  $\mathbf{u} \in \mathbb{K}^{m+n}$ , le vecteur défini par  $u_i = 0, \forall i \in \llbracket 1, m \rrbracket$  et  $u_{m+i} = \underline{u}_i, \forall i \in \llbracket 1, n \rrbracket$ . Montrer que

$$H(\mathbf{u})B = \left( \begin{array}{c|c} B_{1,1} & B_{1,2} \\ \hline 0 & HS \end{array} \right).$$

Soient  $k \in \llbracket 0, n-1 \rrbracket$  et  $A^{[k]} \in \mathcal{M}_n(\mathbb{K})$  la matrice bloc définie par

$$A^{[k]} = \left( \begin{array}{c|c} R^{[k]} & F^{[k]} \\ \hline 0 & A^{[k]} \end{array} \right)$$

où  $R^{[k]}$  est une matrice triangulaire supérieure d'ordre  $k$  et  $A^{[k]}$  une matrice d'ordre  $n-k$ .

Q. 2

a. Sous certaines hypothèses, montrer qu'il existe une matrice de Householder  $H^{[k+1]}$  telle que  $H^{[k+1]}A^{[k]} = A^{[k+1]}$ .

b. Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . Montrer qu'il existe une matrice unitaire  $Q \in \mathcal{M}_n(\mathbb{K})$ , produit d'au plus  $n-1$  matrices de Householder, et une matrice triangulaire supérieure  $R$  telles que  $A = QR$ .

c. Montrer que si  $A$  est réelle alors les coefficients diagonaux de  $R$  peuvent être choisis positifs ou nuls.

d. Montrer que si  $A$  est réelle inversible alors la factorisation  $QR$ , avec  $R$  à coefficient diagonaux strictement positifs, est unique.

### Correction 3.1.10

R. 1

a. D'après le Corollaire 3.21, page 93, avec  $\mathbf{a} = \mathbf{s}$ , en posant  $\alpha = \pm \|\mathbf{s}\|_2 e^{2 \arg s_1}$  et

$$\underline{\mathbf{u}} = \frac{\mathbf{s} - \alpha \mathbf{e}_1^n}{\|\mathbf{s} - \alpha \mathbf{e}_1^n\|}$$

on obtient  $\mathbb{H}(\underline{\mathbf{u}}) = \alpha \mathbf{e}_1^n$ .

On pose  $\underline{\mathbb{H}} = \mathbb{H}(\underline{\mathbf{u}})$ . On a alors sous forme bloc

$$\underline{\mathbb{H}}\mathbf{s} = \underline{\mathbb{H}} \begin{pmatrix} \bullet & \cdots & \bullet \\ \vdots & & \vdots \\ \bullet & \cdots & \bullet \end{pmatrix} = \begin{pmatrix} \pm\alpha & \bullet & \cdots & \bullet \\ 0 & \bullet & \cdots & \bullet \\ \vdots & \vdots & & \vdots \\ 0 & \bullet & \cdots & \bullet \end{pmatrix}$$

b. On a  $\mathbf{u} = \begin{pmatrix} \mathbf{0}_m \\ \underline{\mathbf{u}} \end{pmatrix}$  et

$$\begin{aligned} \mathbb{H}(\mathbf{u}) &= \mathbb{I} - 2\mathbf{u}\mathbf{u}^* = \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \mathbb{I}_n \end{pmatrix} - 2 \begin{pmatrix} \mathbf{0}_m \\ \underline{\mathbf{u}} \end{pmatrix} \begin{pmatrix} \mathbf{0}_m^* & \underline{\mathbf{u}}^* \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \mathbb{I}_n \end{pmatrix} - 2 \begin{pmatrix} \mathbf{0}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \underline{\mathbf{u}}\underline{\mathbf{u}}^* \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \mathbb{I}_n - 2\underline{\mathbf{u}}\underline{\mathbf{u}}^* \end{pmatrix} = \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \underline{\mathbb{H}} \end{pmatrix} \end{aligned}$$

Ce qui donne

$$\mathbb{H}(\mathbf{u})\mathbb{B} = \begin{pmatrix} \mathbb{I}_m & \mathbf{0}_{m,n} \\ \mathbf{0}_{n,m} & \underline{\mathbb{H}} \end{pmatrix} \begin{pmatrix} \mathbb{B}_{1,1} & \mathbb{B}_{1,2} \\ \mathbf{0} & \mathbb{S} \end{pmatrix} = \begin{pmatrix} \mathbb{B}_{1,1} & \mathbb{B}_{1,2} \\ \mathbf{0} & \underline{\mathbb{H}}\mathbb{S} \end{pmatrix}.$$

R. 2

a. On note  $\underline{\mathbf{s}} \in \mathbb{K}^{n-k}$  le premier vecteur colonne de  $\underline{\mathbb{A}}^{[k]}$ , et  $\mathbf{u} = \begin{pmatrix} \mathbf{0}_k \\ \underline{\mathbf{s}} \end{pmatrix}$ . D'après la question précédente si  $\underline{\mathbf{s}} \neq \mathbf{0}$  et  $\underline{\mathbf{s}}$  non colinéaire à  $\mathbf{e}_1^{n-k}$  premier vecteur de la base canonique de  $\mathbb{K}^{n-k}$  alors il existe une matrice de Householder  $\mathbb{H}^{[k+1]} = \mathbb{H}(\mathbf{u})$  et  $\alpha \in \mathbb{K}^*$  tels que

$$\underline{\mathbb{A}}^{[k+1]} \stackrel{\text{def}}{=} \mathbb{H}^{[k+1]}\underline{\mathbb{A}}^{[k]} = \begin{pmatrix} \mathbb{R}^{[k]} & \mathbb{F}^{[k]} \\ \mathbf{0} & \begin{pmatrix} \pm\alpha & \bullet & \cdots & \bullet \\ 0 & \bullet & \cdots & \bullet \\ \vdots & \vdots & & \vdots \\ 0 & \bullet & \cdots & \bullet \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \mathbb{R}^{[k+1]} & \mathbb{F}^{[k+1]} \\ \mathbf{0} & \underline{\mathbb{A}}^{[k+1]} \end{pmatrix}$$

On peut remarquer que si  $\underline{\mathbf{s}} = \mathbf{0}$  ou  $\underline{\mathbf{s}}$  colinéaire à  $\mathbf{e}_1^{n-k}$  alors  $\underline{\mathbb{A}}^{[k]}$  est déjà sous la forme  $\underline{\mathbb{A}}^{[k+1]}$  et donc  $\mathbb{H}^{[k+1]} = \mathbb{I}$ .

b. il suffit d'appliquer itérativement le résultat précédent  $n-1$  fois en posant  $\underline{\mathbb{A}}^{[0]} = \underline{\mathbb{A}}$  et  $\underline{\mathbb{A}}^{[k+1]} = \mathbb{H}^{[k+1]}\underline{\mathbb{A}}^{[k]}$  où  $\mathbb{H}^{[k+1]}$  est soit une matrice de Householder soit la matrice identité. Par construction la matrice  $\underline{\mathbb{A}}^{[n-1]}$  est triangulaire supérieure et l'on a

$$\underline{\mathbb{A}}^{[n-1]} = \mathbb{H}^{[n-1]} \times \cdots \times \mathbb{H}^{[1]}\underline{\mathbb{A}}$$

On pose  $\mathbb{H} = \mathbb{H}^{[n-1]} \times \cdots \times \mathbb{H}^{[1]}$  et  $\mathbb{R} = \underline{\mathbb{A}}^{[n-1]}$ . La matrice  $\mathbb{H}$  est unitaire car produit de matrices unitaires. On note  $\mathbb{Q} = \mathbb{H}^*$ . On a

$$\mathbb{Q} = \mathbb{H}^{[1]} \times \cdots \times \mathbb{H}^{[n-1]}$$

car les matrices de Householder et matrice identité sont unitaires et hermitiennes.

- c. Si  $A$  est réelle alors par construction  $Q$  et  $R$  sont réelles. Les coefficients diagonaux peuvent alors être choisis positifs lors de la construction de chaque matrice de Householder.
- d. Pour montrer l'unicité d'une telle factorisation, on note  $Q_1, Q_2$ , deux matrices orthogonales et  $R_1, R_2$ , deux matrices triangulaires à coefficients diagonaux positifs telles que

$$A = Q_1 R_1 = Q_2 R_2.$$

Comme  $A$  est inversible les coefficients diagonaux de  $R_1$  et  $R_2$  sont strictement positifs. On a alors

$$I = AA^{-1} = Q_1 R_1 R_2^{-1} Q_2^{-1}$$

et donc

$$Q_1^{-1} Q_2 = R_1 R_2^{-1} \stackrel{\text{def}}{=} T.$$

Comme  $Q_1$  est orthogonale on a  $T = Q_1^t Q_2$  et

$$T^t T = (Q_1^t Q_2)^t Q_1^t Q_2 = Q_2^t Q_1 Q_1^t Q_2 = I.$$

La matrice  $T$  est donc orthogonale. De plus  $T = R_1 R_2^{-1}$  est une matrice triangulaire supérieure à coefficients diagonaux strictement positifs puisque produit de triangulaire supérieure à coefficients diagonaux strictement positifs. La matrice  $I$  étant symétrique définie positive, d'après le Théorème 3.15 (factorisation positive de Cholesky) il existe une unique matrice  $L$  triangulaire inférieure à coefficients diagonaux strictement positifs telle que  $LL^t = I$ . Cette matrice  $L$  est évidemment la matrice identité. On en déduit que  $T = L^t = I$  et donc  $Q_1 = Q_2$  et  $R_1 = R_2$ .

Fin correction

### EXERCICE 3.1.11 Algorithmique

Q. 1

Écrire une fonction `FactQR` permettant de calculer la factorisation QR d'une matrice  $A \in \mathcal{M}_n(\mathbb{C})$ .

On pourra utiliser la fonction `Householder` (voir Exercice ??, page ??).

Q. 2

Écrire un programme permettant de tester cette fonction.

### Correction 3.1.11

R. 1

L'objectif est de déterminer les matrices  $Q$ , matrice unitaire, et  $R$  matrice triangulaire supérieure telle que  $A = QR$ .

**Données :**  $A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .

**Résultat :**  $Q$  : matrice unitaire de  $\mathcal{M}_n(\mathbb{K})$ .

$R$  : matrice triangulaire supérieure de  $\mathcal{M}_n(\mathbb{K})$ .

On rappelle la technique utilisée dans la correction de l'exercice 3.1.10 pour déterminer l'ensemble des matrices de Householder permettant de transformer la matrice  $A$  en une matrice triangulaire supérieure. On pose

$$A^{[0]} = A, \quad A^{[k+1]} = H^{[k+1]} A^{[k]}, \quad \forall k \in [0, n-2]$$

où  $H^{[k+1]}$  est soit une matrice de Householder soit la matrice identité. Plus précisément, on note  $\underline{s} \in \mathbb{K}^{n-k}$  le vecteur composé des  $n-k$  dernières composantes de la  $k+1$ -ème colonne de  $A^{[k]}$  et

$$\underline{a} = \begin{pmatrix} \mathbf{0}_k \\ \underline{s} \end{pmatrix}.$$

- Si  $s_1 = 0$  ou  $\underline{s}$  colinéaire à  $\mathbf{e}_1^{n-k}$  premier vecteur de la base canonique de  $\mathbb{K}^{n-k}$  alors

$$H^{[k+1]} = I.$$

En notant  $e_{k+1}^n$  le  $k+1$ -ème vecteur de la base canonique de  $\mathbb{K}^n$ , cette matrice peut-être calculée avec la fonction `Householder` par

$$[\mathbb{H}^{[k+1]}, \alpha] \leftarrow \text{Householder}(\mathbf{a}, e_{k+1}^n, 1)$$

- sinon  $\mathbb{H}^{[k+1]} = \mathbb{I}$ .

On a vu que dans ce cas  $\mathbb{A}^{[n-1]}$  est triangulaire supérieure. On pose  $\mathbb{H} = \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$  qui est une matrice unitaire. On a alors  $\mathbb{R} = \mathbb{A}^{[n-1]} = \mathbb{H}\mathbb{A}$  et  $\mathbb{Q} = \mathbb{H}^*$ .

#### Algorithme 3.18 $\mathcal{R}_0$

1: Calculer  $\mathbb{Q}$  et  $\mathbb{R}$

#### Algorithme 3.18 $\mathcal{R}_1$

1:  $\mathbb{H} \leftarrow \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$   
 2:  $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$   
 3:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

#### Algorithme 3.18 $\mathcal{R}_1$

1:  $\mathbb{H} \leftarrow \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$   
 2:  $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$   
 3:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

#### Algorithme 3.18 $\mathcal{R}_2$

1:  $\mathbb{H} \leftarrow \mathbb{I}$   
 2:  $\mathbb{A}^{[0]} \leftarrow \mathbb{A}$   
 3: **Pour**  $k \leftarrow 0$  à  $n-2$  **faire**  
 4:   Calculer  $\mathbb{H}^{[k+1]}$  à partir de  $\mathbb{A}^{[k]}$   
 5:    $\mathbb{A}^{[k+1]} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{A}^{[k]}$   
 6:    $\mathbb{H} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{H}$   
 7: **Fin**  
 8:  $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$                    ▷ ou  $\mathbb{R} \leftarrow \mathbb{A}^{[n-1]}$   
 9:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

Il est inutile de stocker les matrices  $\mathbb{A}^{[k]}$  et  $\mathbb{H}^{[k+1]}$ :

#### Algorithme 3.18 $\mathcal{R}_2$

1:  $\mathbb{H} \leftarrow \mathbb{I}$ ,  $\mathbb{A}^{[0]} \leftarrow \mathbb{A}$   
 2: **Pour**  $k \leftarrow 0$  à  $n-2$  **faire**  
 3:   Calculer  $\mathbb{H}^{[k+1]}$  à partir de  $\mathbb{A}^{[k]}$   
 4:    $\mathbb{A}^{[k+1]} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{A}^{[k]}$   
 5:    $\mathbb{H} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{H}$   
 6: **Fin**  
 7:  $\mathbb{R} \leftarrow \mathbb{A}^{[n-1]}$   
 8:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

#### Algorithme 3.18 $\mathcal{R}_3$

1:  $\mathbb{H} \leftarrow \mathbb{I}$ ,  $\mathbb{R} \leftarrow \mathbb{A}$   
 2: **Pour**  $k \leftarrow 0$  à  $n-2$  **faire**  
 3:   Calculer  $\mathbb{S} (= \mathbb{H}^{[k+1]})$  à partir de  $\mathbb{R} (= \mathbb{A}^{[k]})$   
 4:    $\mathbb{R} \leftarrow \mathbb{S} * \mathbb{R}$                    ▷ compute  $\mathbb{A}^{[k+1]}$   
 5:    $\mathbb{H} \leftarrow \mathbb{S} * \mathbb{H}$   
 6: **Fin**  
 7:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

#### Algorithme 3.18 $\mathcal{R}_3$

1:  $\mathbb{H} \leftarrow \mathbb{I}$ ,  $\mathbb{R} \leftarrow \mathbb{A}$   
 2: **Pour**  $k \leftarrow 0$  à  $n-2$  **faire**  
 3:   Calculer  $\mathbb{S} (= \mathbb{H}^{[k+1]})$  à partir de  $\mathbb{R}$   
 4:    $\mathbb{R} \leftarrow \mathbb{S} * \mathbb{R}$   
 5:    $\mathbb{H} \leftarrow \mathbb{S} * \mathbb{H}$   
 6: **Fin**  
 7:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

#### Algorithme 3.18 $\mathcal{R}_4$

1:  $\mathbb{H} \leftarrow \mathbb{I}$ ,  $\mathbb{R} \leftarrow \mathbb{A}$   
 2: **Pour**  $k \leftarrow 0$  à  $n-2$  **faire**  
 3:    $\mathbf{a} \leftarrow [\mathbf{0}_k; \mathbb{R}(k+1:n, k+1)]$   
 4:    $e_{k+1}^n \in \mathbb{C}^n$ ,  $e_{k+1}^n(i) = \delta_{k+1,i}$ ,  $\forall i \in \llbracket 1, n \rrbracket$ .  
 5:    $[\mathbb{R}, \alpha] \leftarrow \text{Householder}(\mathbf{a}, e_{k+1}^n, 1)$   
 6:    $\mathbb{R} \leftarrow \mathbb{S} * \mathbb{R}$   
 7:    $\mathbb{H} \leftarrow \mathbb{S} \mathbb{H}$   
 8: **Fin**  
 9:  $\mathbb{Q} \leftarrow \mathbb{H}^*$

**Algorithme 3.18** Fonction **FactQR****Données :**  $A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .**Résultat :**  $Q$  : matrice unitaire de  $\mathcal{M}_n(\mathbb{K})$ . $R$  : matrice triangulaire supérieure de  $\mathcal{M}_n(\mathbb{K})$ .

```

1: Fonction  $[Q, R] \leftarrow \text{FactQR}(A)$ 
2:  $H \leftarrow I$ 
3:  $R \leftarrow A$ 
4: Pour  $k \leftarrow 0$  à  $n - 2$  faire
5:    $a \leftarrow 0_n$ 
6:   Pour  $i \leftarrow k + 1$  à  $n$  faire
7:      $a(i) \leftarrow R(i, k + 1)$ 
8:   Fin
9:    $e \leftarrow 0_n$ ,  $e(k + 1) \leftarrow 1$ 
10:   $[S, \alpha] \leftarrow \text{Householder}(a, e, 1)$ 
11:   $R \leftarrow S * R$ 
12:   $H \leftarrow S * H$ 
13: Fin
14:  $Q \leftarrow H^*$ 
15: Fin

```

**R. 2**

A faire!

Fin correction

## 3.2 Normes vectorielles et normes matricielles

### 3.2.1 Normes vectorielles

**Définition 3.2.1.** Une *norme* sur un espace vectoriel  $V$  est une application  $\|\bullet\| : V \rightarrow \mathbb{R}^+$  qui vérifie les propriétés suivantes

- ◇  $\|\mathbf{v}\| = 0 \iff \mathbf{v} = 0$ ,
- ◇  $\|\alpha\mathbf{v}\| = |\alpha| \|\mathbf{v}\|$ ,  $\forall \alpha \in \mathbb{K}$ ,  $\forall \mathbf{v} \in V$ ,
- ◇  $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$ ,  $\forall (\mathbf{u}, \mathbf{v}) \in V^2$  (inégalité triangulaire).

Une norme sur  $V$  est également appelée *norme vectorielle*. On appelle *espace vectoriel normé* un espace vectoriel muni d'une norme.

Les trois normes suivantes sont les plus couramment utilisées :

$$\|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|$$

$$\|\mathbf{v}\|_2 = \left( \sum_{i=1}^n |v_i|^2 \right)^{1/2}$$

$$\|\mathbf{v}\|_\infty = \max_{i \in [1, n]} |v_i|.$$

**Proposition 3.2.1.** Soit  $\mathbf{v} \in \mathbb{K}^n$ . Pour tout nombre réel  $p \geq 1$ , l'application  $\|\bullet\|_p$  définie par

$$\|\mathbf{v}\|_p = \left( \sum_{i=1}^n |v_i|^p \right)^{1/p}$$

est une norme sur  $\mathbb{K}^n$ .

**Lemme 3.2.1** (Inégalité de Cauchy-Schwarz).  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{K}^n$

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2. \quad (3.36)$$

Cette inégalité s'appelle l'**inégalité de Cauchy-Schwarz**. On a égalité si et seulement si  $\mathbf{x}$  et  $\mathbf{y}$  sont colinéaires.

Preuve. Exercice ??, page ??

□

**Lemme 3.2.2** (Inégalité de Hölder). Pour  $p > 1$  et  $\frac{1}{p} + \frac{1}{q} = 1$ , on a  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{K}^n$

$$\sum_{i=1}^n |x_i y_i| \leq \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \left( \sum_{i=1}^n |y_i|^q \right)^{1/q} = \|\mathbf{x}\|_p \|\mathbf{y}\|_q. \quad (3.37)$$

Cette inégalité s'appelle l'**inégalité de Hölder**.

Preuve. Exercice ??, page ??

□

**Définition 3.2.2.** Deux normes  $\|\bullet\|$  et  $\|\bullet\|'$ , définies sur un même espace vectoriel  $V$ , sont **équivalentes** s'il existe deux constantes  $C$  et  $C'$  telles que

$$\|\mathbf{x}\|' \leq C \|\mathbf{x}\| \quad \text{et} \quad \|\mathbf{x}\| \leq C' \|\mathbf{x}\|' \quad \text{pour tout } \mathbf{x} \in V. \quad (3.38)$$

**Proposition 3.2.2.** Sur un espace vectoriel de dimension finie toutes les normes sont équivalentes.

### 3.2.2 Normes matricielles

**Définition 3.2.3.** Une **norme matricielle** sur  $\mathcal{M}_n(\mathbb{K})$  est une application  $\|\bullet\| : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  vérifiant

- $\|\mathbb{A}\| = 0 \iff \mathbb{A} = 0$ ,
- $\|\alpha \mathbb{A}\| = |\alpha| \|\mathbb{A}\|, \forall \alpha \in \mathbb{K}, \forall \mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ ,
- $\|\mathbb{A} + \mathbb{B}\| \leq \|\mathbb{A}\| + \|\mathbb{B}\|, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$  (inégalité triangulaire)
- $\|\mathbb{A}\mathbb{B}\| \leq \|\mathbb{A}\| \|\mathbb{B}\|, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$

**Proposition 3.2.3.** Etant donné une norme vectorielle  $\|\bullet\|$  sur  $\mathbb{K}^n$ , l'application  $\|\bullet\|_s : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  définie par

$$\|\mathbb{A}\|_s \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} \quad (3.39)$$

est une norme matricielle, appelée **norme matricielle subordonnée** (à la norme vectorielle donnée).

Elle vérifie

$$\|A\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \|\mathbf{v}\| \leq 1}} \|A\mathbf{v}\| = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \|\mathbf{v}\|=1}} \|A\mathbf{v}\| = \inf \{ \alpha \in \mathbb{R} : \|A\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n \}. \quad (3.40)$$

De plus, pour tout  $\mathbf{v} \in \mathbb{K}^n$  on a

$$\|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\| \quad (3.41)$$

et il existe au moins un vecteur  $\mathbf{u} \in \mathbb{K}^n \setminus \{0\}$  tel que

$$\|A\mathbf{u}\| = \|A\|_s \|\mathbf{u}\|. \quad (3.42)$$

Soit  $\mathbb{I}$  la matrice identité d'ordre  $n$ , on a

$$\|\mathbb{I}\|_s = 1. \quad (3.43)$$

*Preuve.* On note  $\mathcal{B} = \{\mathbf{v} \in \mathbb{K}^n ; \|\mathbf{v}\| \leq 1\}$  la boule unité de  $\mathbb{K}^n$  et  $\mathcal{S} = \{\mathbf{v} \in \mathbb{K}^n ; \|\mathbf{v}\| = 1\}$  la sphère unité de  $\mathbb{K}^n$ . On note que les ensembles  $\mathcal{B}$  et  $\mathcal{S}$  sont des compacts car image réciproque de l'application continue  $\mathbf{v} \mapsto \|\mathbf{v}\|$  par le fermé borné  $[0, 1]$  (pour la boule) et le singleton  $\{1\}$  (pour la sphère).

- Vérifions que les égalités suivantes sont vraies :

$$\|A\|_s \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\mathbf{v} \in \mathcal{B}} \|A\mathbf{v}\| = \sup_{\mathbf{v} \in \mathcal{S}} \|A\mathbf{v}\|$$

On a

$$\|A\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \left\| A \frac{\mathbf{v}}{\|\mathbf{v}\|} \right\| = \sup_{\mathbf{v} \in \mathcal{S}} \|A\mathbf{v}\|$$

Comme  $\mathcal{S} \subset \mathcal{B}$  on a aussi

$$\sup_{\mathbf{v} \in \mathcal{B}} \|A\mathbf{v}\| \geq \sup_{\mathbf{v} \in \mathcal{S}} \|A\mathbf{v}\|. \quad (3.44)$$

On peut aussi remarquer que

$$\sup_{\mathbf{v} \in \mathcal{B}} \|A\mathbf{v}\| = \sup_{\substack{\mathbf{v} \in \mathcal{B} \\ \mathbf{v} \neq 0}} \|A\mathbf{v}\| \quad (3.45)$$

De plus,  $\forall \mathbf{w} \in \mathcal{B} \setminus \{0\}$ , en posant  $\mathbf{u} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \in \mathcal{S}$ , on a  $\mathbf{w} = \|\mathbf{w}\| \mathbf{u}$  et

$$\|A\mathbf{w}\| = \|\mathbf{w}\| \|A\mathbf{u}\| \leq \|A\mathbf{u}\| \text{ car } \|\mathbf{w}\| \leq 1.$$

Or on a

$$\|A\mathbf{u}\| \leq \sup_{\mathbf{v} \in \mathcal{S}} \|A\mathbf{v}\|.$$

et on obtient alors

$$\sup_{\substack{\mathbf{w} \in \mathcal{B} \\ \mathbf{w} \neq 0}} \|A\mathbf{w}\| \leq \sup_{\mathbf{v} \in \mathcal{S}} \|A\mathbf{v}\|.$$

En utilisant (R31.31) et (R31.32), on en déduit

$$\sup_{\mathbf{w} \in \mathcal{B}} \|A\mathbf{w}\| = \sup_{\mathbf{u} \in \mathcal{S}} \|A\mathbf{u}\|.$$

- Vérifions que l'application  $\|\cdot\|_s$  est bien définie sur  $\mathcal{M}_n(\mathbb{K})$  i.e.  $\forall A \in \mathcal{M}_n(\mathbb{K}), \|A\|_s < +\infty$ .  
L'application  $\mathbf{v} \mapsto \|A\mathbf{v}\|$  est continue donc son sup sur la sphère unité qui est compacte est atteint.
- Montrons que  $\forall \mathbf{v} \in \mathbb{K}^n, \|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\|$ .  
On a par définition du sup

$$\|A\|_s = \sup_{\substack{\mathbf{u} \in \mathbb{K}^n \\ \mathbf{u} \neq 0}} \frac{\|A\mathbf{u}\|}{\|\mathbf{u}\|} \geq \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|}, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\}.$$

et donc

$$\|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\|, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\}.$$

- Montrons qu'il existe  $\mathbf{u} \in \mathbb{K}^n$  tel que

$$\mathbf{u} \neq \mathbf{0} \text{ et } \|\mathbb{A}\mathbf{u}\| = \|\mathbb{A}\|_s \|\mathbf{u}\|.$$

On a

$$\|\mathbb{A}\|_s = \sup_{\mathbf{v} \in \mathcal{S}} \|\mathbb{A}\mathbf{v}\|$$

La sphère unité étant compacte et l'application  $\mathbf{v} \mapsto \|\mathbb{A}\mathbf{v}\|$  étant continue, il existe  $\mathbf{w} \in \mathcal{S}$  tel que  $\|\mathbb{A}\|_s = \|\mathbb{A}\mathbf{w}\|$ . Soit  $\lambda \in \mathbb{K}^*$  et  $\mathbf{u} = \lambda\mathbf{w} \neq \mathbf{0}$ . On a  $\|\mathbf{u}\| = |\lambda|$  et

$$\|\mathbb{A}\|_s = \|\mathbb{A}\mathbf{w}\| = \left\| \mathbb{A} \frac{\mathbf{u}}{\|\mathbf{u}\|} \right\| = \frac{1}{\|\mathbf{u}\|} \|\mathbb{A}\mathbf{u}\|.$$

- On a immédiatement

$$\|\mathbb{I}\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{I}\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbf{v}\|}{\|\mathbf{v}\|} = 1.$$

- Montrons que  $\|\bullet\|_s$  est une norme matricielle.

- a.  $\|\mathbb{A}\|_s = 0 \iff \mathbb{A}_s = \mathbb{O}$  ?

$\longleftarrow$  trivial.

$\longrightarrow$  Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ .

$$\begin{aligned} \|\mathbb{A}\|_s = 0 &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} \implies \|\mathbb{A}\mathbf{v}\| = 0, \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \\ &\implies \mathbb{A}\mathbf{v} = \mathbf{0}, \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \end{aligned}$$

Soit  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$  la base canonique de  $\mathbb{K}^n$ . On a alors  $\forall j \in \llbracket 1, n \rrbracket, \mathbb{A}\mathbf{e}_j = \mathbf{0}$  et on en déduit que

$$A_{i,j} = \langle \mathbf{e}_i, \mathbb{A}\mathbf{e}_j \rangle = 0, \forall (i, j) \in \llbracket 1, n \rrbracket.$$

et donc  $\mathbb{A} = \mathbb{O}$ .

- b. Montrons que  $\|\alpha\mathbb{A}\| = |\alpha| \|\mathbb{A}\|, \forall \alpha \in \mathbb{K}, \forall \mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ .

Soient  $\alpha \in \mathbb{K}$  et  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ . On a  $\alpha\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  (car  $\mathcal{M}_n(\mathbb{K})$  est un espace vectoriel) et

$$\begin{aligned} \|\alpha\mathbb{A}\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\alpha\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{|\alpha| \|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} \text{ car } \|\alpha\mathbf{u}\| = |\alpha| \|\mathbf{u}\| \\ &= |\alpha| \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} = |\alpha| \|\mathbb{A}\|_s. \end{aligned}$$

- c. Montrons que  $\|\mathbb{A} + \mathbb{B}\|_s \leq \|\mathbb{A}\|_s + \|\mathbb{B}\|_s, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$

Soient  $\mathbb{A}$  et  $\mathbb{B}$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$ . On a  $\mathbb{A} + \mathbb{B} \in \mathcal{M}_n(\mathbb{K})$  car  $\mathcal{M}_n(\mathbb{K})$  est un espace vectoriel et

$$\begin{aligned} \|\mathbb{A} + \mathbb{B}\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|(\mathbb{A} + \mathbb{B})\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{A}\mathbf{v} + \mathbb{B}\mathbf{v}\|}{\|\mathbf{v}\|} \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{A}\mathbf{v}\| + \|\mathbb{B}\mathbf{v}\|}{\|\mathbf{v}\|} \text{ par inégalité triangulaire dans } \mathbb{K}^n \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} + \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{B}\mathbf{v}\|}{\|\mathbf{v}\|} = \|\mathbb{A}\|_s + \|\mathbb{B}\|_s. \end{aligned}$$

- d. Montrons que  $\|\mathbb{A}\mathbb{B}\|_s \leq \|\mathbb{A}\|_s \|\mathbb{B}\|_s, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$ .

Soient  $\mathbb{A}$  et  $\mathbb{B}$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$ . On a  $\mathbb{A}\mathbb{B} \in \mathcal{M}_n(\mathbb{K})$  par définition du produit matriciel

et

$$\begin{aligned}\|AB\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|(AB)\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A(B\mathbf{v})\|}{\|\mathbf{v}\|} \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\|_s \|B\mathbf{v}\|}{\|\mathbf{v}\|} \quad \text{car } \|A\mathbf{u}\| \leq \|A\|_s \|\mathbf{u}\| \quad \forall \mathbf{u} \in \mathbb{K}^n \\ &\leq \|A\|_s \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|B\mathbf{v}\|}{\|\mathbf{v}\|} = \|A\|_s \|B\|_s.\end{aligned}$$

□

**Théorème 3.2.1.** Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . On a

$$\|A\|_1 \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|_1}{\|\mathbf{v}\|_1} = \max_{j \in [1, n]} \sum_{i=1}^n |a_{ij}| \quad (3.46)$$

$$\|A\|_2 \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sqrt{\rho(A^*A)} \quad (3.47)$$

$$\|A\|_\infty \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|A\mathbf{v}\|_\infty}{\|\mathbf{v}\|_\infty} = \max_{i \in [1, n]} \sum_{j=1}^n |a_{ij}| \quad (3.48)$$

La norme  $\|\bullet\|_2$  est invariante par transformation unitaire :

$$UU^* = \mathbb{I} \implies \|A\|_2 = \|AU\|_2 = \|UA\|_2 = \|U^*AU\|_2. \quad (3.49)$$

*Preuve.* Exercice ??, page ??

□

**Corollaire 3.2.1.** a. Si une matrice  $A$  est hermitienne, on a  $\|A\|_2 = \rho(A)$ .

b. Si une matrice  $A$  est unitaire, on a  $\|A\|_2 = 1$ .

*Preuve.* a. Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne. Elle est donc normale. D'après le théorème 3.1.2 (réduction des matrice) page 63, il existe une matrice unitaire  $U$  et une matrice diagonale  $D$  telles que

$$A = UDU^*.$$

Les matrices  $A$  et  $D$  sont semblables: elles ont les mêmes valeurs propres et donc

$$\rho(A) = \rho(D).$$

De plus, comme  $A$  est hermitienne, ses valeurs propres sont réelles et donc  $D \in \mathcal{M}_n(\mathbb{R})$ . Comme la norme 2 est invariante par transformation unitaire, on a

$$\|A\|_2 = \|UDU^*\|_2 = \|D\|_2.$$

De plus,  $D$  étant réelle, on obtient

$$\|D\|_2 = \sqrt{\rho(D^*D)} = \sqrt{\rho(D^2)}$$

La matrice  $D$  étant diagonale on en déduit que

$$\rho(D^2) = \rho(D)^2.$$

On obtient donc

$$\|A\|_2 = \rho(D) = \rho(A).$$

b. Si  $A \in \mathcal{M}_n(\mathbb{C})$  est unitaire ou si  $A \in \mathcal{M}_n(\mathbb{R})$  est orthogonale alors  $AA^* = \mathbb{I}$  et donc

$$\|A\|_2 = \sqrt{\rho(AA^*)} = \sqrt{\rho(\mathbb{I})} = 1.$$

□

**Théorème 3.2.2.** a. Soit  $\mathbb{A}$  une matrice carrée quelconque et  $\|\bullet\|$  une norme matricielle subordonnée ou non, quelconque. Alors

$$\rho(\mathbb{A}) \leq \|\mathbb{A}\|. \quad (3.50)$$

b. Etant donné une matrice  $\mathbb{A}$  et un nombre  $\varepsilon > 0$ , il existe au moins une norme matricielle subordonnée telle que

$$\|\mathbb{A}\| \leq \rho(\mathbb{A}) + \varepsilon. \quad (3.51)$$

*Preuve.* a. Soient  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  et  $(\lambda, \mathbf{u})$  un élément propre de  $\mathbb{A}$  :

$$\mathbf{u} \in \mathbb{C}^n \setminus \{\mathbf{0}\}, \quad \text{et} \quad \mathbb{A}\mathbf{u} = \lambda\mathbf{u}.$$

- Si la norme matricielle est subordonnée à la norme vectorielle  $\|\bullet\|$  alors

$$\|\lambda\mathbf{u}\| = |\lambda| \|\mathbf{u}\| = \|\mathbb{A}\mathbf{u}\| \leq \|\mathbb{A}\| \|\mathbf{u}\|$$

Or  $\mathbf{u} \in \mathbb{C}^n \setminus \{\mathbf{0}\}$  et donc on en déduit que

$$\forall \lambda \in \rho(\mathbb{A}), \quad |\lambda| \leq \|\mathbb{A}\|.$$

- Si la norme est quelconque (non forcément subordonnée), l'inégalité suivante n'est plus vérifiée

$$\|\mathbb{A}\mathbf{u}\| \leq \|\mathbb{A}\| \|\mathbf{u}\|.$$

Par contre on a pour toute matrice  $\mathbb{B} \in \mathcal{M}_n(\mathbb{K})$

$$\|\mathbb{A}\mathbb{B}\| \leq \|\mathbb{A}\| \|\mathbb{B}\|.$$

Comme  $(\lambda, \mathbf{u})$  un élément propre de  $\mathbb{A}$ , on a

$$\mathbb{A}\mathbf{u}\mathbf{u}^* = \lambda\mathbf{u}\mathbf{u}^*$$

et donc

$$\|\mathbb{A}(\mathbf{u}\mathbf{u}^*)\| = \|\lambda(\mathbf{u}\mathbf{u}^*)\| = |\lambda| \|\mathbf{u}\mathbf{u}^*\|.$$

De plus on a

$$\|\mathbb{A}(\mathbf{u}\mathbf{u}^*)\| \leq \|\mathbb{A}\| \|\mathbf{u}\mathbf{u}^*\|$$

ce qui donne

$$|\lambda| \|\mathbf{u}\mathbf{u}^*\| \leq \|\mathbb{A}\| \|\mathbf{u}\mathbf{u}^*\|$$

Comme la matrice  $\mathbb{B} = \mathbf{u}\mathbf{u}^*$  est non nulle car  $\mathbf{u}$  non nul et

$$(\mathbf{u}\mathbf{u}^*)\mathbf{u} = \mathbf{u}(\mathbf{u}^*\mathbf{u}) = \mathbf{u}\langle \mathbf{u}, \mathbf{u} \rangle \neq \mathbf{0}.$$

on obtient

$$|\lambda| \leq \|\mathbb{A}\|$$

et on en déduit alors

$$\forall \lambda \in \rho(\mathbb{A}), \quad |\lambda| \leq \|\mathbb{A}\|.$$

b. voir [1], théorème 1.4-3 page 18-19. □

**Théorème 3.2.3.** L'application  $\|\bullet\|_E : \mathcal{M}_n \rightarrow \mathbb{R}^+$  définie par

$$\|\mathbb{A}\|_E = \left( \sum_{(i,j) \in \llbracket 1, n \rrbracket^2} |a_{ij}|^2 \right)^{1/2} = \sqrt{\text{tr}(\mathbb{A}^* \mathbb{A})}, \quad (3.52)$$

pour toute matrice  $\mathbb{A} = (a_{ij})$  d'ordre  $n$ , est une norme matricielle non subordonnée (pour  $n \geq 2$ ),

invariante par transformation unitaire et qui vérifie

$$\|A\|_2 \leq \|A\|_E \leq \sqrt{n} \|A\|_2, \quad \forall A \in \mathcal{M}_n. \quad (3.53)$$

De plus  $\|\mathbb{1}\|_E = \sqrt{n}$ .

**Théorème 3.2.4.** a. Soit  $\|\bullet\|$  une norme matricielle subordonnée, et  $\mathbb{B}$  une matrice vérifiant

$$\|\mathbb{B}\| < 1.$$

Alors la matrice  $(\mathbb{1} + \mathbb{B})$  est inversible, et

$$\|(\mathbb{1} + \mathbb{B})^{-1}\| \leq \frac{1}{1 - \|\mathbb{B}\|}.$$

b. Si une matrice de la forme  $(\mathbb{1} + \mathbb{B})$  est singulière, alors nécessairement

$$\|\mathbb{B}\| \geq 1$$

pour toute norme matricielle, subordonnée ou non.

*Preuve.* a. Par l'absurde, on suppose la matrice  $\mathbb{1} + \mathbb{B}$  singulière (non inversible). Alors 0 est une de ses valeurs propres. On note  $(0, \mathbf{u})$  un élément propre de  $\mathbb{1} + \mathbb{B}$ . On a donc

$$\begin{aligned} (\mathbb{1} + \mathbb{B})\mathbf{u} = \mathbf{0} &\Leftrightarrow \mathbb{B}\mathbf{u} = -\mathbf{u} \\ &\Leftrightarrow (-1, \mathbf{u}) \text{ élément propre de } \mathbb{B} \end{aligned}$$

ce qui est en contradiction avec  $\rho(\mathbb{B}) \leq \|\mathbb{B}\| < 1$ .

La matrice  $\mathbb{1} + \mathbb{B}$  est donc inversible.

Par définition de l'inverse

$$(\mathbb{1} + \mathbb{B})(\mathbb{1} + \mathbb{B})^{-1} = \mathbb{1}$$

et donc

$$(\mathbb{1} + \mathbb{B})^{-1} + \mathbb{B}(\mathbb{1} + \mathbb{B})^{-1} = \mathbb{1}$$

ce qui s'écrit aussi

$$(\mathbb{1} + \mathbb{B})^{-1} = \mathbb{1} - \mathbb{B}(\mathbb{1} + \mathbb{B})^{-1}.$$

En prenant sa norme on obtient

$$\begin{aligned} \|(\mathbb{1} + \mathbb{B})^{-1}\| &= \|\mathbb{1} - \mathbb{B}(\mathbb{1} + \mathbb{B})^{-1}\| \\ &\leq \|\mathbb{1}\| + \|\mathbb{B}(\mathbb{1} + \mathbb{B})^{-1}\| \quad \text{par l'inégalité triangulaire} \\ &\leq 1 + \|\mathbb{B}(\mathbb{1} + \mathbb{B})^{-1}\| \quad \text{car pour une norme subordonnée } \|\mathbb{1}\| = 1 \\ &\leq 1 + \|\mathbb{B}\| \|(\mathbb{1} + \mathbb{B})^{-1}\| \end{aligned}$$

Ce qui donne

$$\|(\mathbb{1} + \mathbb{B})^{-1}\| - \|\mathbb{B}\| \|(\mathbb{1} + \mathbb{B})^{-1}\| \leq 1$$

et donc l'inégalité est démontrée.

b. On a démontré lors de la démonstration par l'absurde que si  $\mathbb{1} + \mathbb{B}$  est singulière (non inversible) alors  $\rho(\mathbb{B}) \geq 1$ . Comme pour toute norme  $\rho(\mathbb{B}) \leq \|\mathbb{B}\|$  (voir théorème 3.2.2, page 104), on obtient

$$\|\mathbb{B}\| \geq 1.$$

□

### 3.2.3 Suites de vecteurs et de matrices

**Définition 3.2.4.** Soit  $V$  un espace vectoriel muni d'une norme  $\|\bullet\|$ , on dit qu'une suite  $(v_k)$  d'éléments de  $V$  **converge vers un élément**  $v \in V$ , si

$$\lim_{k \rightarrow \infty} \|v_k - v\| = 0$$

et on écrit

$$v = \lim_{k \rightarrow \infty} v_k.$$

**Théorème 3.2.5** (admis, voir [1] Théorème 1.5-1 page 21-22). Soit  $\mathbb{B}$  une matrice carrée. Les conditions suivantes sont équivalentes :

- $\lim_{k \rightarrow \infty} \mathbb{B}^k = 0$ ,
- $\lim_{k \rightarrow \infty} \mathbb{B}^k v = 0$  pour tout vecteur  $v$ ,
- $\rho(\mathbb{B}) < 1$ ,
- $\|\mathbb{B}\| < 1$  pour au moins une norme matricielle subordonnée  $\|\bullet\|$ .

**Théorème 3.2.6** (admis, voir [1] Théorème 1.5-2 page 22). Soit  $\mathbb{B}$  une matrice carrée, et  $\|\bullet\|$  une norme matricielle quelconque. Alors

$$\lim_{k \rightarrow \infty} \|\mathbb{B}^k\|^{1/k} = \rho(\mathbb{B}).$$

### 3.3 Conditionnement d'un système linéaire

Pour la résolution numérique d'un système linéaire  $Ax = b$ , il est rare que les données  $A$  et  $b$  du problème ne soient pas entachées d'erreurs (aussi minimes soient-elles). La question qui se pose alors est de savoir si de petites perturbations sur les données ne peuvent pas entraîner des erreurs importantes sur le calcul de la solution.

#### Exemple de R.S. Wilson

Soient

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad \Delta A = \begin{pmatrix} 0 & 0 & \frac{1}{10} & \frac{1}{5} \\ \frac{2}{25} & \frac{1}{25} & 0 & 0 \\ 0 & -\frac{1}{50} & -\frac{11}{100} & 0 \\ -\frac{1}{100} & -\frac{1}{100} & 0 & -\frac{1}{50} \end{pmatrix}$$

et  $b^t = (32, 23, 33, 31)$ ,  $(\Delta b)^t = (\frac{1}{100}, -\frac{1}{100}, \frac{1}{100}, -\frac{1}{100})$ . Des calculs exacts donnent

$$\begin{aligned} Ax = b & \iff x^t = (1, 1, 1, 1) \\ Au = (b + \Delta b) & \iff u^t = \left( \frac{91}{50}, -\frac{9}{25}, \frac{27}{20}, \frac{79}{100} \right) \\ & \approx (1.8, -0.36, 1.3, 0.79) \\ (A + \Delta A)v = b & \iff v^t = (-81, 137, -34, 22) \\ (A + \Delta A)y = (b + \Delta b) & \iff y^t = \left( -\frac{18283543}{461600}, \frac{31504261}{461600}, -\frac{3741501}{230800}, \frac{5235241}{461600} \right) \\ & \approx (-39.61, 68.25, -16.21, 11.34) \end{aligned}$$

Il est clair sur cet exemple que de petites perturbations sur les données peuvent entraîner des erreurs importantes sur la solution exacte.

On dit que le système linéaire précédent est **mal conditionné** ou qu'il a un **mauvais conditionnement** car il est sujet à de fortes variations de la solution pour de petites perturbations des données. A contrario,

on dit qu'il est **bien conditionné** ou qu'il a un **bon conditionnement** si de petites perturbations des données n'entraînent qu'une variation *raisonnable* de la solution.  
Une nouvelle question : est-il possible de "mesurer" le **conditionnement** d'une matrice?

### Définitions et résultats

**Définition 3.3.1.** Soit  $\|\cdot\|$  une norme matricielle subordonnée, le conditionnement d'une matrice régulière  $A$ , associé à cette norme, est le nombre

$$\text{cond}(A) = \|A\| \|A^{-1}\|.$$

Nous noterons  $\text{cond}_p(A) = \|A\|_p \|A^{-1}\|_p$ .

**Proposition 3.3.1.** Soit  $A$  une matrice régulière. On a les propriétés suivantes

- $\forall \alpha \in \mathbb{K}^*$ ,  $\text{cond}(\alpha A) = \text{cond}(A)$ .
- $\text{cond}_p(A) \geq 1$ ,  $\forall p \in \llbracket 1, +\infty \rrbracket$ .
- $\text{cond}_2(A) = 1$  si et seulement si  $A = \alpha Q$  avec  $\alpha \in \mathbb{K}^*$  et  $Q$  matrice unitaire

*Preuve.* Soit  $A$  une matrice régulière.

- Soit  $\alpha \in \mathbb{K}^*$ , on a

$$\begin{aligned} \text{cond}(\alpha A) &\stackrel{\text{def}}{=} \|\alpha A\| \|(\alpha A)^{-1}\| = |\alpha| \|A\| \left\| \frac{1}{\alpha} A^{-1} \right\| \\ &= |\alpha| \|A\| \frac{1}{|\alpha|} \|A^{-1}\| = \|A\| \|A^{-1}\| \\ &= \text{cond}(A) \end{aligned}$$

- On a  $\|I\| = \|AA^{-1}\|$ . Or pour toute norme subordonnée, on a  $\|I\| = 1$  et donc  $1 = \|AA^{-1}\| \leq \|A\| \|A^{-1}\| = \text{cond}(A)$ .
- Admis.** Voir par exemple [7] Théorème 2 page 142-143.

□

**Théorème 3.3.1.** Soit  $A$  une matrice inversible. Soient  $\mathbf{x}$  et  $\mathbf{x} + \Delta\mathbf{x}$  les solutions respectives de

$$A\mathbf{x} = \mathbf{b} \quad \text{et} \quad A(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}.$$

Supposons  $\mathbf{b} \neq \mathbf{0}$ , alors l'inégalité

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(A) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

est satisfaite, et c'est la meilleure possible : pour une matrice  $A$  donnée, on peut trouver des vecteurs  $\mathbf{b} \neq \mathbf{0}$  et  $\Delta\mathbf{b} \neq \mathbf{0}$  tels qu'elle devienne une égalité.

*Preuve.* On a

$$A\mathbf{x} = \mathbf{b} \quad \text{et} \quad A(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

or  $A(\mathbf{x} + \Delta\mathbf{x}) = A\mathbf{x} + A\Delta\mathbf{x}$  et donc  $A\Delta\mathbf{x} = \Delta\mathbf{b}$  ou encore  $\Delta\mathbf{x} = A^{-1}\Delta\mathbf{b}$ . Ceci donne

$$\|\Delta\mathbf{x}\| = \|A^{-1}\Delta\mathbf{b}\| \leq \|A^{-1}\| \|\Delta\mathbf{b}\|$$

et

$$\|\mathbf{b}\| = \|A\mathbf{x}\| \leq \|A\| \|\mathbf{x}\|.$$

On en déduit

$$\|\Delta\mathbf{x}\| \|\mathbf{b}\| \leq \|A\| \|\mathbf{x}\| \|A^{-1}\| \|\Delta\mathbf{b}\|.$$

Comme  $\mathbf{b} \neq \mathbf{0}$ , on a  $\mathbf{x} = \mathbb{A}^{-1}\mathbf{b} \neq \mathbf{0}$  et, les normes étant positives, on obtient

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

D'après la Proposition 3.2.3, pour toute norme matricielle subordonnée il existe au moins un vecteur  $\mathbf{u} \in \mathbb{K}^n \setminus \{0\}$  et un vecteur  $\mathbf{v} \in \mathbb{K}^n \setminus \{0\}$  tel que

$$\|\mathbb{A}^{-1}\mathbf{u}\| = \|\mathbb{A}^{-1}\| \|\mathbf{u}\| \text{ et } \|\mathbb{A}\mathbf{v}\| = \|\mathbb{A}\| \|\mathbf{v}\|.$$

En posant  $\mathbf{b} = \mathbb{A}\mathbf{v}$  et  $\Delta\mathbf{b} = \mathbf{u}$  on a bien égalité.  $\square$

**Théorème 3.3.2.** Soient  $\mathbb{A}$  et  $\mathbb{A} + \Delta\mathbb{A}$  deux matrices inversibles. Soient  $\mathbf{x}$  et  $\mathbf{x} + \Delta\mathbf{x}$  les solutions respectives de

$$\mathbb{A}\mathbf{x} = \mathbf{b} \text{ et } (\mathbb{A} + \Delta\mathbb{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}.$$

Supposons  $\mathbf{b} \neq \mathbf{0}$ , alors on a

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x} + \Delta\mathbf{x}\|} \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbb{A}\|}{\|\mathbb{A}\|}.$$

*Preuve.* On a

$$\mathbb{A}\mathbf{x} = \mathbf{b} = (\mathbb{A} + \Delta\mathbb{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbb{A}\mathbf{x} + \mathbb{A}\Delta\mathbf{x} + \Delta\mathbb{A}(\mathbf{x} + \Delta\mathbf{x})$$

et donc

$$\mathbb{A}\Delta\mathbf{x} + \Delta\mathbb{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{0} \iff \Delta\mathbf{x} = -\mathbb{A}^{-1}\Delta\mathbb{A}(\mathbf{x} + \Delta\mathbf{x})$$

On en déduit alors

$$\|\Delta\mathbf{x}\| = \|\mathbb{A}^{-1}\Delta\mathbb{A}(\mathbf{x} + \Delta\mathbf{x})\| \leq \|\mathbb{A}^{-1}\| \|\Delta\mathbb{A}\| \|\mathbf{x} + \Delta\mathbf{x}\|$$

De plus, on a  $\text{cond}(\mathbb{A}) \stackrel{\text{def}}{=} \|\mathbb{A}\| \|\mathbb{A}^{-1}\|$  et donc  $\|\mathbb{A}^{-1}\| = \frac{\text{cond}(\mathbb{A})}{\|\mathbb{A}\|}$  ce qui donne

$$\|\Delta\mathbf{x}\| \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbb{A}\|}{\|\mathbb{A}\|} \|\mathbf{x} + \Delta\mathbf{x}\|.$$

Comme  $\mathbf{b} \neq \mathbf{0}$ , on a  $\mathbf{x} + \Delta\mathbf{x} = (\mathbb{A} + \Delta\mathbb{A})^{-1}\mathbf{b} \neq \mathbf{0}$  et de l'inégalité précédente, on déduit alors

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x} + \Delta\mathbf{x}\|} \leq \text{cond}(\mathbb{A}) \frac{\|\Delta\mathbb{A}\|}{\|\mathbb{A}\|}.$$

$\square$

**Remarque 3.3.1.** a. Plus le conditionnement d'une matrice est proche de 1, meilleur il est.

b. Si une matrice  $\mathbb{A}$  est mal conditionnée, il est possible de trouver une matrice  $\mathbb{P}$  inversible tel que la matrice  $\mathbb{P}^{-1}\mathbb{A}$  soit mieux conditionnée... Résoudre  $\mathbb{A}\mathbf{x} = \mathbf{b}$  est alors équivalent à résoudre  $\mathbb{P}^{-1}\mathbb{A}\mathbf{x} = \mathbb{P}^{-1}\mathbf{b}$ . La matrice  $\mathbb{P}$  est appelé **préconditionneur**. Le choix  $\mathbb{P} = \mathbb{A}$  est idéal mais n'est pas réellement utilisable. Si  $\mathbb{A}$  est une matrice dominante alors on peut utiliser le preconditionneur de Jacobi:  $\mathbb{P} = \text{diag}(\mathbb{A})$ .

## 3.4 Méthodes itératives

### 3.4.1 Principe

On souhaite résoudre le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$  par des **méthodes itératives**. Ces dernières consistent en la détermination d'une **matrice d'itération**  $\mathbb{B}$  et d'un vecteur  $\mathbf{c}$  tels que la suite de vecteurs  $\mathbf{x}^{[k]}$  définie par

$$\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}, \quad k \geq 0, \quad \mathbf{x}^{[0]} \text{ arbitraire}$$

soit telle que  $\lim_{k \rightarrow \infty} \mathbf{x}^{[k]} = \underline{\mathbf{x}}$  et  $\underline{\mathbf{x}}$  solution de  $\mathbb{A}\mathbf{x} = \mathbf{b}$ . Bien évidemment les matrices  $\mathbb{B}$  et les vecteurs  $\mathbf{c}$  dépendront de  $\mathbb{A}$  et  $\mathbf{b}$ .

Nous allons étudier plusieurs méthodes itératives et pour chacune d'entre elles nous expliciterons la matrice d'itération  $\mathbb{B}$  et le vecteur  $\mathbf{c}$  associé.

### 3.4.2 Présentation des méthodes usuelles

Soit  $A \in \mathcal{M}_n(\mathbb{K})$  une matrice régulière, d'éléments diagonaux non-nuls, et  $\mathbf{b} \in \mathbb{K}^n$ . On note  $\mathbb{D}$  la **matrice diagonale** telle que  $\mathbb{D} = \text{diag}(A)$ ,  $\mathbb{E}$  la **matrice triangulaire inférieure** à diagonale nulle définie par

$$\begin{cases} E_{ij} = 0, & i \leq j \\ E_{ij} = -A_{ij} & i > j \end{cases} \quad (3.54)$$

et  $\mathbb{F}$  la **matrice triangulaire supérieure** à diagonale nulle définie par

$$\begin{cases} F_{ij} = 0, & i \geq j \\ F_{ij} = -A_{ij} & i < j \end{cases} \quad (3.55)$$

On a alors

$$\begin{aligned} A &= \begin{pmatrix} A_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{n,n} \end{pmatrix} + \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ A_{2,1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n-1} & 0 \end{pmatrix} + \begin{pmatrix} 0 & A_{1,2} & \cdots & A_{1,n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & A_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \\ &= \mathbb{D} - \mathbb{E} - \mathbb{F} = \begin{pmatrix} \ddots & & & \\ & \mathbb{D} & & \\ -\mathbb{E} & & \ddots & \end{pmatrix} \end{aligned} \quad (3.56)$$

On rappelle que la  $i$ -ème équation du système linéaire  $A\mathbf{x} = \mathbf{b}$  s'écrit

$$b_i = \sum_{j=1}^n A_{i,j}x_j = \sum_{j=1}^{i-1} A_{i,j}x_j + A_{i,i}x_i + \sum_{j=i+1}^n A_{i,j}x_j. \quad (3.57)$$

Il faut aussi noter que la matrice diagonale  $\mathbb{D}$  est inversible car les éléments diagonaux de  $A$  sont non nuls par hypothèse.

#### Méthode de Jacobi

Pour obtenir la méthode itérative de Jacobi, il suffit de *mettre*, dans la formule (3.4.2), l'itéré  $k+1$  sur le terme diagonale et l'itéré  $k$  sur les autres termes pour avoir

$$b_i = \sum_{j=1}^{i-1} A_{i,j}x_j^{[k]} + A_{i,i}x_i^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]}.$$

ce qui donne

$$x_i^{[k+1]} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij}x_j^{[k]} \right) \quad \forall i \in \llbracket 1, n \rrbracket \quad (3.58)$$

#### Méthode de Gauss-Seidel

Pour obtenir la méthode itérative de Gauss-Seidel, il suffit de *mettre*, dans la formule (3.4.2), l'itéré  $k+1$  sur la partie *triangulaire inférieure* et l'itéré  $k$  sur les autres termes pour avoir

$$b_i = \sum_{j=1}^{i-1} A_{i,j}x_j^{[k+1]} + A_{i,i}x_i^{[k+1]} + \sum_{j=i+1}^n A_{i,j}x_j^{[k]}.$$

ce qui donne

$$x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij}x_j^{[k]} \right) \quad \forall i \in \llbracket 1, n \rrbracket \quad (3.59)$$

### Méthodes de relaxation

Ces méthodes sont basées sur un paramètre de relaxation  $w \in \mathbb{R}^*$  et sont données par

$$x_i^{[k+1]} = w \underline{x}_i^{[k+1]} + (1-w)x_i^{[k]}$$

où  $\underline{x}_i^{[k+1]}$  est obtenu à partir de l'une des deux méthodes précédentes.

Avec la méthode de Jacobi

$$x_i^{(k+1)} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket.$$

Avec la méthode de Gauss-Seidel

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

Cette dernière méthode de relaxation, utilisant la méthode de Gauss-Seidel, est appelée méthode S.O.R. (successive over relaxation)

#### EXERCICE 3.4.1

En écrivant  $A$  sous la forme  $A = D - E - F$ , montrer que les méthodes itératives de Jacobi, Gauss-Seidel et S.O.R. s'écrivent sous la forme  $\mathbf{x}^{[k+1]} = B\mathbf{x}^{[k]} + \mathbf{c}$ , où l'on exprimera les matrices  $B$  et les vecteurs  $\mathbf{c}$  en fonction de  $D$ ,  $E$ ,  $F$  et  $\mathbf{b}$ .

**Correction 3.4.1** On rappelle que la  $i$ -ème équation du système linéaire  $A\mathbf{x} = \mathbf{b}$  s'écrit

$$b_i = \sum_{j=1}^n A_{i,j} x_j = \sum_{j=1}^{i-1} A_{i,j} x_j + A_{i,i} x_i + \sum_{j=i+1}^n A_{i,j} x_j.$$

et que ceci correspond à l'écriture matricielle

$$\mathbf{b} = D\mathbf{x} - E\mathbf{x} - F\mathbf{x}$$

- Pour la **méthode de Jacobi** on a,  $\forall i \in \llbracket 1, n \rrbracket$ ,

$$b_i = A_{i,i} x_i^{[k+1]} + \sum_{j=1}^{i-1} A_{i,j} x_j^{[k]} + \sum_{j=i+1}^n A_{i,j} x_j^{[k]}.$$

ce qui s'écrit matriciellement

$$\mathbf{b} = D\mathbf{x}^{[k+1]} - E\mathbf{x}^{[k]} - F\mathbf{x}^{[k]}.$$

On a donc

$$D\mathbf{x}^{[k+1]} = \mathbf{b} + (E + F)\mathbf{x}^{[k]}$$

Comme la matrice  $D$  est inversible, on obtient

$$\mathbf{x}^{[k+1]} = D^{-1}(E + F)\mathbf{x}^{[k]} + D^{-1}\mathbf{b}$$

La matrice d'itération de Jacobi est  $B = D^{-1}(E + F)$  et le vecteur  $\mathbf{c} = D^{-1}\mathbf{b}$ .

- Pour la **méthode de Gauss-Seidel** on a,  $\forall i \in \llbracket 1, n \rrbracket$ ,

$$b_i = A_{i,i} x_i^{[k+1]} + \sum_{j=1}^{i-1} A_{i,j} x_j^{[k+1]} + \sum_{j=i+1}^n A_{i,j} x_j^{[k]}.$$

ce qui s'écrit matriciellement

$$\mathbf{b} = D\mathbf{x}^{[k+1]} - E\mathbf{x}^{[k+1]} - F\mathbf{x}^{[k]}.$$

On a donc

$$(D - E)\mathbf{x}^{[k+1]} = \mathbf{b} + F\mathbf{x}^{[k]}$$

Comme la matrice  $D - E$  est inversible (matrice triangulaire inférieure d'éléments diagonaux non nuls), on obtient

$$\mathbf{x}^{[k+1]} = (D - E)^{-1}F\mathbf{x}^{[k]} + (D - E)^{-1}\mathbf{b}$$

La matrice d'itération de Gauss-Seidel est  $B = (D - E)^{-1}F$  et le vecteur  $\mathbf{c} = (D - E)^{-1}\mathbf{b}$ .

•

**R. 1**

Pour la **méthode S.O.R.** on a ,  $\forall i \in \llbracket 1, n \rrbracket$ ,

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]}$$

ce qui s'écrit aussi

$$\frac{A_{ii}}{w} x_i^{[k+1]} + \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} = b_i - \sum_{j=i+1}^n A_{ij} x_j^{[k]} + \frac{1-w}{w} A_{ii} x_i^{[k]}$$

et matriciellement on obtient

$$\left( \frac{\mathbb{D}}{w} - \mathbb{E} \right) \mathbf{x}^{[k+1]} = \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) \mathbf{x}^{[k]} + \mathbf{b}.$$

Comme la matrice  $\left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)$  est inversible (car triangulaire inférieure à éléments diagonaux non nuls), on a

$$\mathbf{x}^{[k+1]} = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) \mathbf{x}^{[k]} + \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \mathbf{b}$$

La matrice d'itération de S.O.R. est  $\mathbb{B} = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right)$  et le vecteur  $\mathbf{c} = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \mathbf{b}$ .

~~~~~ Fin correction ~~~~~

**Proposition 3.4.1.** Soit  $\mathbb{A}$  une matrice régulière telle que tous ses éléments diagonaux soient non nuls. On note  $\mathbb{D} = \text{diag}(\mathbb{A})$  et  $\mathbb{E}, \mathbb{F}$ , les matrices à diagonales nulles respectivement triangulaire inférieure et supérieure telles que  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ . On pose  $\mathbb{L} = \mathbb{D}^{-1}\mathbb{E}$  et  $\mathbb{U} = \mathbb{D}^{-1}\mathbb{F}$ .

La matrice d'itération de la méthode de Jacobi, notée  $\mathbb{J}$ , est donnée par

$$\mathbb{J} = \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F}) = \mathbb{L} + \mathbb{U}, \quad (3.60)$$

La matrice d'itération de la méthode S.O.R., notée  $\mathcal{L}_w$ , est donnée par

$$\mathcal{L}_w = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right) = (\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U}). \quad (3.61)$$

et elle vérifie

$$\rho(\mathcal{L}_w) \geq |w-1|. \quad (3.62)$$

La matrice d'itération de Gauss-Seidel est  $\mathcal{L}_1$  et elle correspond à

$$\mathcal{L}_1 = (\mathbb{D} - \mathbb{E})^{-1} \mathbb{F} = (\mathbb{I} - \mathbb{L})^{-1} \mathbb{U}. \quad (3.63)$$

*Preuve.* Les résultats découlent de l'Exercice 3.4.1 pour l'écriture en fonction des matrices  $\mathbb{D}, \mathbb{E}$  et  $\mathbb{F}$ . Pour l'écriture en fonction des matrices  $\mathbb{L}$  et  $\mathbb{U}$  seule l'équation (3.61) n'est pas forcément immédiate. On a vu que

$$\mathcal{L}_w = \left( \frac{\mathbb{D}}{w} - \mathbb{E} \right)^{-1} \left( \frac{1-w}{w} \mathbb{D} + \mathbb{F} \right)$$

et comme  $\mathbb{E} = \mathbb{D}\mathbb{L}$  et  $\mathbb{F} = \mathbb{D}\mathbb{U}$  on obtient

$$\begin{aligned}\mathcal{L}_w &= \left(\frac{\mathbb{D}}{w} - \mathbb{D}\mathbb{L}\right)^{-1} \left(\frac{1-w}{w}\mathbb{D} + \mathbb{D}\mathbb{U}\right) \\ &= \left(\frac{1}{w}\mathbb{D}[\mathbb{I} - w\mathbb{L}]\right)^{-1} \left(\frac{1}{w}\mathbb{D}[(1-w)\mathbb{I} + w\mathbb{U}]\right) \\ &= (\mathbb{I} - w\mathbb{L})^{-1} \left(\frac{1}{w}\mathbb{D}\right)^{-1} \left(\frac{1}{w}\mathbb{D}\right) ((1-w)\mathbb{I} + w\mathbb{U}) \\ &= (\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U})\end{aligned}$$

Il reste à démontrer l'inégalité (3.62). La matrice  $\mathbb{L}$  est triangulaire inférieure à diagonale nulle car elle est le produit d'une matrice diagonale (et donc triangulaire inférieure)  $\mathbb{D}^{-1}$  et d'une matrice triangulaire inférieure  $\mathbb{E}$  à diagonale nulle. De même la matrice  $\mathbb{U}$  est triangulaire supérieure à diagonale nulle.

On sait que le déterminant d'une matrice est égale aux produits de ses valeurs propres comptées avec leurs multiplicités. En notant  $n$  la dimension de la matrice  $\mathcal{L}_w$ , et en notant  $\lambda_i(\mathcal{L}_w)$  ses  $n$  valeurs propres, on a donc

$$\det(\mathcal{L}_w) = \prod_{i=1}^n \lambda_i(\mathcal{L}_w).$$

Le rayon spectral de  $\mathcal{L}_w$ , noté  $\rho(\mathcal{L}_w)$ , correspond au plus grand des modules des valeurs propres. On a alors

$$\rho(\mathcal{L}_w) = \max_{i \in \llbracket 1, n \rrbracket} |\lambda_i(\mathcal{L}_w)| \geq |\det(\mathcal{L}_w)|^{1/n}$$

De plus on a

$$\det(\mathcal{L}_w) = \det\left((\mathbb{I} - w\mathbb{L})^{-1} ((1-w)\mathbb{I} + w\mathbb{U})\right) = \det\left((\mathbb{I} - w\mathbb{L})^{-1}\right) \det\left(((1-w)\mathbb{I} + w\mathbb{U})\right)$$

La matrice  $\mathbb{I} - w\mathbb{L}$  est triangulaire inférieure à diagonale unité donc son inverse aussi. On en déduit  $\det\left((\mathbb{I} - w\mathbb{L})^{-1}\right) = 1$ . La matrice  $(1-w)\mathbb{I} + w\mathbb{U}$  est triangulaire supérieure avec tous ses éléments diagonaux valant  $1-w$  et donc  $\det\left(((1-w)\mathbb{I} + w\mathbb{U})\right) = (1-w)^n$ . On a alors  $|\det(\mathcal{L}_w)| = |1-w|^n$  et

$$\rho(\mathcal{L}_w) \geq |\det(\mathcal{L}_w)|^{1/n} = |1-w|.$$

□

### 3.4.3 Etude de la convergence

Les méthodes de Jacobi, Gauss-Seidel et S.O.R. peuvent s'écrire sous la forme

$$\mathbb{M}\mathbf{x}^{[k+1]} = \mathbb{N}\mathbf{x}^{[k]} + \mathbf{b}$$

avec  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  et, dans ce cas, la matrice d'itération est  $\mathbb{B} = \mathbb{M}^{-1}\mathbb{N}$ .

- Jacobi :  $\mathbb{M} = \mathbb{D}$  et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$  car

$$\mathbf{x}^{[k+1]} = \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})\mathbf{x}^{[k]} + \mathbb{D}^{-1}\mathbf{b}$$

- Gauss-Seidel :  $\mathbb{M} = \mathbb{D} - \mathbb{E}$  et  $\mathbb{N} = \mathbb{F}$  car

$$\mathbf{x}^{[k+1]} = (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}^{[k]} + (\mathbb{D} - \mathbb{E})^{-1}\mathbf{b}$$

- S.O.R. :  $\mathbb{M} = \frac{\mathbb{D}}{w} - \mathbb{E}$  et  $\mathbb{N} = \frac{1-w}{w}\mathbb{D} + \mathbb{F}$  car

$$\mathbf{x}^{[k+1]} = \left(\frac{\mathbb{D}}{w} - \mathbb{E}\right)^{-1} \left(\frac{1-w}{w}\mathbb{D} + \mathbb{F}\right)\mathbf{x}^{[k]} + \left(\frac{\mathbb{D}}{w} - \mathbb{E}\right)^{-1}\mathbf{b}$$

**Théorème 3.4.1.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  une matrice inversible décomposée sous la forme  $\mathbb{A} = \mathbb{M} - \mathbb{N}$

avec  $\mathbb{M}$  inversible. On pose

$$\mathbb{B} = \mathbb{M}^{-1}\mathbb{N} \quad \text{et} \quad \mathbf{c} = \mathbb{M}^{-1}\mathbf{b}.$$

Alors la suite définie par

$$\mathbf{x}^{[0]} \in \mathbb{K}^n \quad \text{et} \quad \mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

converge vers  $\bar{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$  quelque soit  $\mathbf{x}^{[0]}$  si et seulement si  $\rho(\mathbb{B}) < 1$ .

*Preuve.* Comme  $\bar{\mathbf{x}} = \mathbb{A}^{-1}\mathbf{b}$  (sans présupposer la convergence) on a  $\mathbb{M}\bar{\mathbf{x}} = \mathbb{N}\bar{\mathbf{x}} + \mathbf{b}$  et alors

$$\bar{\mathbf{x}} = \mathbb{M}^{-1}\mathbb{N}\bar{\mathbf{x}} + \mathbb{M}^{-1}\mathbf{b} = \mathbb{B}\bar{\mathbf{x}} + \mathbf{c}$$

On obtient donc

$$\bar{\mathbf{x}} - \mathbf{x}^{[k+1]} = \mathbb{B}(\bar{\mathbf{x}} - \mathbf{x}^{[k]})$$

Or la suite  $\mathbf{x}^{[k]}$  converge vers  $\bar{\mathbf{x}}$  si et seulement si la suite  $\mathbf{e}^{[k]} \stackrel{\text{def}}{=} \bar{\mathbf{x}} - \mathbf{x}^{[k]}$  converge vers  $\mathbf{0}$ . On a

$$\mathbf{e}^{[k]} = \mathbb{B}^k \mathbf{e}^{[0]}, \quad \forall k \in \mathbb{N}.$$

D'après le Théorème 3.2.5, page 106, on a  $\lim_{k \rightarrow +\infty} \mathbb{B}^k \mathbf{e}^{[0]} = \mathbf{0}$ ,  $\forall \mathbf{e}^{[0]} \in \mathbb{K}^n$  si et seulement si  $\rho(\mathbb{B}) < 1$ .  $\square$

On introduit maintenant la notion de **taux de convergence** pour permettre la comparaison des *performances* de différentes méthodes itératives.

Pour simplifier, on suppose la matrice d'itération  $\mathbb{B}$  normale. D'après le théorème de réduction des matrices (Théorème 3.1.2, page 63), il existe une matrice unitaire  $\mathbb{U}$  et une matrice diagonale  $\mathbb{D}$  telles que

$$\mathbb{B} = \mathbb{U}\mathbb{D}\mathbb{U}^*.$$

De plus, les éléments diagonaux de  $\mathbb{D}$  sont les valeurs propres de  $\mathbb{B}$  et on a alors, pour tout  $p \in \llbracket 1, \infty \rrbracket$

$$\|\mathbb{D}\|_p = \rho(\mathbb{D}) = \rho(\mathbb{B}).$$

Avec les notations de la preuve du Théorème 3.4.1, on a alors

$$\mathbf{e}^{[k]} = \mathbb{B}^k \mathbf{e}^{[0]} = (\mathbb{U}\mathbb{D}\mathbb{U}^*)^k \mathbf{e}^{[0]} = \mathbb{U}\mathbb{D}^k \mathbb{U}^* \mathbf{e}^{[0]} \quad \text{car } \mathbb{U} \text{ unitaire}$$

En posant  $\mathbf{E}^{[k]} = \mathbb{U}^* \mathbf{e}^{[k]}$ , on a

$$\mathbf{E}^{[k]} = \mathbb{D}^k \mathbf{E}^{[0]}$$

et on obtient

$$\|\mathbf{E}^{[k]}\|_p \leq \|\mathbb{D}\|_p^k \|\mathbf{E}^{[0]}\|_p = \rho(\mathbb{B})^k \|\mathbf{E}^{[0]}\|_p.$$

On déduit alors

$$\|\mathbf{e}^{[k]}\|_p \leq \text{cond}_p(\mathbb{U}) \rho(\mathbb{B})^k \|\mathbf{e}^{[0]}\|_p.$$

On est alors assuré de la convergence si  $\rho(\mathbb{B}) < 1$ . Le **facteur asymptotique de convergence** est  $\rho(\mathbb{B})$  et le nombre  $\sigma \stackrel{\text{def}}{=} -\ln \rho(\mathbb{B})$  est appelé **taux asymptotique de convergence**.  
Lorsque  $\rho(\mathbb{B}) < 1$ , pour réduire l'erreur initiale d'un facteur  $\epsilon > 0$ , on doit avoir

$$\|\mathbf{E}^{[k]}\|_p \leq \rho(\mathbb{B})^k \|\mathbf{E}^{[0]}\|_p \leq \|\mathbf{E}^{[0]}\|_p \epsilon$$

c'est à dire

$$k \geq \frac{\ln(\epsilon)}{\ln(\rho(\mathbb{B}))}.$$

Bien évidemment, plus le rayon spectral de  $\mathbb{B}$  est proche de 0, plus rapide est la convergence

**Proposition 3.4.2.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  une matrice tridiagonale, i.e.  $A_{i,j} = 0$ , si  $|i - j| > 1$ . Avec les notations de la Proposition 3.4.1, on a

$$\rho(\mathcal{L}_1) = \rho(\mathbb{J})^2.$$

La méthode de Gauss-Seidel converge donc plus vite que la méthode de Jacobi.

Preuve. voir Exercice 3.4.4, page 128.  $\square$

**Proposition 3.4.3.** Soit  $A$  une matrice vérifiant  $A_{i,i} \neq 0 \forall i$ . Une condition nécessaire de convergence pour la méthode S.O.R. est que  $0 < w < 2$ .

Preuve. On a vu en (voir Proposition 3.4.1, page 111) que  $\rho(\mathcal{L}_w) \geq |w - 1|$ . Donc si  $\rho(\mathcal{L}_w) \geq 1$ , la non-convergence est certaine d'après le Théorème 3.2.5, page 106. Une condition nécessaire (mais non suffisante) de convergence est que  $|w - 1| < 1$  i.e.  $w \in ]0, 2[$ .  $\square$

**Proposition 3.4.4.** Soit  $A \in \mathcal{M}_n(\mathbb{R})$  une matrice tridiagonale dont les éléments diagonaux sont non nuls. On suppose que les valeurs propres de la matrice d'itération de Jacobi  $J$  sont réelles et que  $\rho(J) < 1$ .

On note  $w_0$  le paramètre optimal de la méthode S.O.R. vérifiant

$$\rho(\mathcal{L}_{w_0}) = \min_{w \in ]0, 2[} (\rho(\mathcal{L}_w)).$$

et donné par

$$w_0 = \frac{2}{1 + \sqrt{1 - \rho(J)^2}} > 1. \quad (3.64)$$

On a alors

$$\rho(\mathcal{L}_{w_0}) = w_0 - 1 \text{ et } \rho(\mathcal{L}_{w_0}) \leq \rho(\mathcal{L}_1) = \rho(J)^2 < \rho(J).$$

Preuve. voir [1], Théorème 5.3-5, pages 106 à 109.  $\square$

Pour illustrer cette proposition, on génère une matrice tridiagonale aléatoire inversible à éléments diagonaux non nuls de telle sorte que les valeurs propres de la matrice d'itération de Jacobi  $J$  sont réelles et que  $\rho(J) < 1$ .

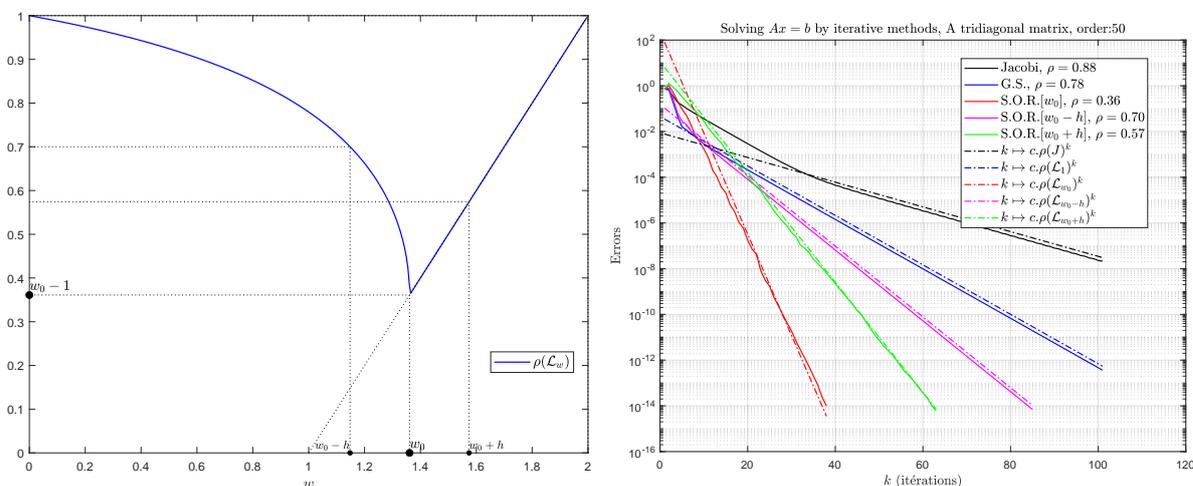


Figure 3.3: to do

**Théorème 3.4.2.** Soit  $A$  une matrice à diagonale strictement dominante ou une matrice inversible à diagonale fortement dominante alors

- la méthode de Jacobi est convergente,
- si  $w \in ]0, 1]$  la méthode S.O.R. est convergente.

Preuve. voir [7], vol.2, Théorème 19 et 20, pages 346 à 349.  $\square$

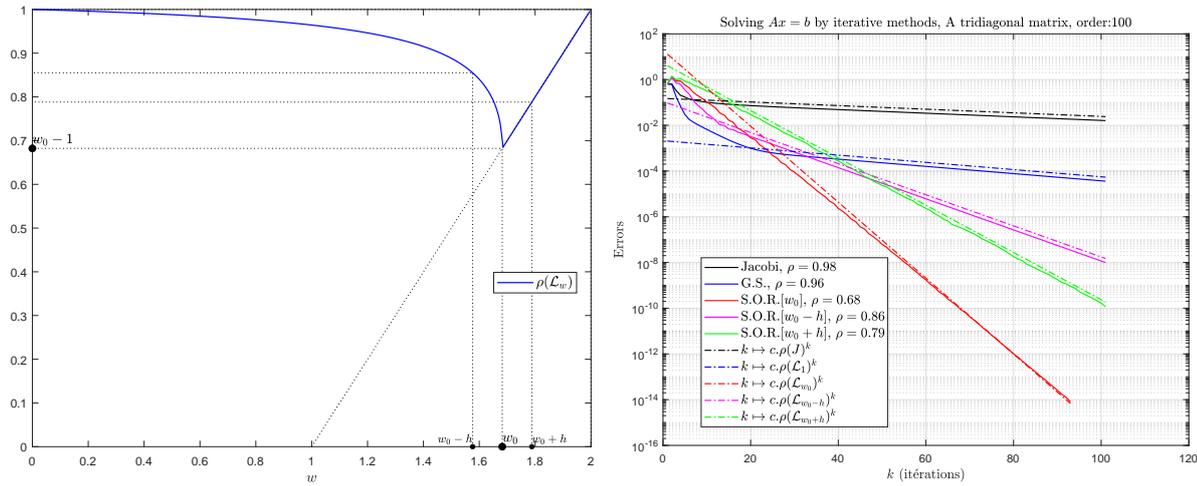


Figure 3.4: to do

**Théorème 3.4.3.** Soient  $A$  une matrice hermitienne inversible décomposée en  $A = M - N$  avec  $M$  inversible et  $M^* + N$  hermitienne définie positive. On a alors

$$\rho(M^{-1}N) < 1 \text{ si et seulement si } A \text{ est définie positive.}$$

Preuve. (voir Exercice 3.4.2, page 115) □

#### EXERCICE 3.4.2

Soit  $A \in \mathcal{M}_{n,n}(\mathbb{C})$  une matrice hermitienne inversible décomposée en  $A = M - N$  où  $M$  est inversible. On note  $B = I - M^{-1}A$ .

Q. 1

Montrer que la matrice  $M^* + N$  est hermitienne.

On suppose maintenant que  $M^* + N$  est définie positive.

Q. 2

Soit  $x$  un vecteur quelconque de  $\mathbb{C}^n$  et  $y = Bx$ .

a. Montrer que

$$\langle x, Ax \rangle - \langle y, Ay \rangle = \langle x, AM^{-1}Ax \rangle + \langle M^{-1}Ax, Ax \rangle - \langle M^{-1}Ax, AM^{-1}Ax \rangle \quad (3.1)$$

et

$$x - y = M^{-1}Ax. \quad (3.2)$$

b. En déduire que

$$\langle x, Ax \rangle - \langle y, Ay \rangle = \langle (x - y), (M^* + N)(x - y) \rangle. \quad (3.3)$$

Q. 3

Montrer que si  $A$  est définie positive alors  $\rho(B) < 1$ .

Q. 4

Démontrer par l'absurde que si  $\rho(B) < 1$  alors  $A$  est définie positive.

#### Correction 3.4.2

R. 1

On a

$$\begin{aligned} (M^* + N)^* &= M + N^* \\ &= A + N + N^* = A^* + N + N^* \text{ car } A \text{ est hermitienne} \\ &= M^* + N. \end{aligned}$$

La matrice  $M^* + N$  est donc hermitienne.

R. 2

a. On a  $\mathbf{y} = \mathbb{B}\mathbf{x}$  avec  $\mathbb{B} = \mathbb{I} - \mathbb{M}^{-1}\mathbb{A}$  ce qui donne

$$\mathbf{x} - \mathbf{y} = \mathbf{x} - \mathbb{B}\mathbf{x} = (\mathbb{I} - \mathbb{B})\mathbf{x} = \mathbb{M}^{-1}\mathbb{A}\mathbf{x}.$$

L'équation (3.2) est donc démontrée. Pour prouver (3.1), on note que

$$\mathbf{y} = \mathbf{x} - \mathbb{M}^{-1}\mathbb{A}\mathbf{x}$$

et donc

$$\begin{aligned} \langle \mathbf{y}, \mathbb{A}\mathbf{y} \rangle &= \langle \mathbf{x} - \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}(\mathbf{x} - \mathbb{M}^{-1}\mathbb{A}\mathbf{x}) \rangle \\ &= \langle \mathbf{x}, \mathbb{A}\mathbf{x} \rangle - \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbf{x} \rangle - \langle \mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle + \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle. \end{aligned}$$

On en déduit immédiatement (3.1).

b. En utilisant (3.2), on obtient

$$\begin{aligned} \langle \mathbf{x} - \mathbf{y}, (\mathbb{M}^* + \mathbb{N})(\mathbf{x} - \mathbf{y}) \rangle &= \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, (\mathbb{M}^* + \mathbb{N})\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle \\ &= \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, (\mathbb{M} + \mathbb{N}^*)\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle \quad \text{car } \mathbb{M}^* + \mathbb{N} \text{ hermitienne} \\ &= \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, (\mathbb{M} + \mathbb{M}^* - \mathbb{A}^*)\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle \quad \text{car } \mathbb{N} = \mathbb{M} - \mathbb{A} \\ &= \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbf{x} \rangle + \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{M}^*\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle - \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle \quad \text{car } \mathbb{A} \text{ hermitienne} \end{aligned}$$

Or, par propriété du produit scalaire, on a

$$\langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{M}^*\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle = \langle \mathbb{M}\mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle = \langle \mathbb{A}\mathbf{x}, \mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle = \langle \mathbf{x}, \mathbb{A}^*\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle.$$

Comme  $\mathbb{A}$  est hermitienne, on obtient

$$\langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{M}^*\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle = \langle \mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle.$$

On abouti alors à

$$\langle \mathbf{x} - \mathbf{y}, (\mathbb{M}^* + \mathbb{N})(\mathbf{x} - \mathbf{y}) \rangle = \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbf{x} \rangle + \langle \mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle - \langle \mathbb{M}^{-1}\mathbb{A}\mathbf{x}, \mathbb{A}\mathbb{M}^{-1}\mathbb{A}\mathbf{x} \rangle.$$

L'équation (3.3) est obtenue en utilisant (3.1).

R. 3

On veut démontrer que sous les hypothèses  $\mathbb{A}$  hermitienne définie positive et  $\mathbb{M}^* + \mathbb{N}$  (hermitienne) définie positive on a  $\rho(\mathbb{B}) < 1$ , c'est à dire que pour tout élément propre  $(\lambda, \mathbf{u})$  de  $\mathbb{B}$  alors  $|\lambda| < 1$ . Soit  $(\lambda, \mathbf{u}) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$  un élément propre de  $\mathbb{B}$ . On a  $\mathbb{B}\mathbf{u} = \lambda\mathbf{u}$ . En prenant  $\mathbf{x} = \mathbf{u}$  dans Q.2, on a  $\mathbf{y} = \mathbb{B}\mathbf{u} = \lambda\mathbf{u}$  et donc  $\mathbf{x} - \mathbf{y} = (1 - \lambda)\mathbf{u}$ . De (3.3) on obtient

$$\langle \mathbf{u}, \mathbb{A}\mathbf{u} \rangle - \langle \lambda\mathbf{u}, \lambda\mathbb{A}\mathbf{u} \rangle = \langle (1 - \lambda)\mathbf{u}, (\mathbb{M}^* + \mathbb{N})((1 - \lambda)\mathbf{u}) \rangle$$

c'est à dire

$$(1 - |\lambda|^2) \langle \mathbf{u}, \mathbb{A}\mathbf{u} \rangle = |1 - \lambda|^2 \langle \mathbf{u}, (\mathbb{M}^* + \mathbb{N})\mathbf{u} \rangle. \quad (\text{R2.6})$$

On va montrer que  $|1 - \lambda| > 0$ , c'est à dire  $\lambda \neq 1$ . Pour celà on effectue une démonstration par l'absurde.

**Par l'absurde** on suppose  $\lambda = 1$ . Dans ce cas, comme  $\mathbb{B}\mathbf{u} = \mathbf{u}$ , on a  $\mathbf{x} = \mathbf{y}$ . De (3.2), on déduit alors  $\mathbb{M}^{-1}\mathbb{A}\mathbf{x} = \mathbf{0}$  et comme  $\mathbb{A}$  et  $\mathbb{M}^{-1}$  sont inversibles  $\mathbf{x} = \mathbf{0}$ . Or  $\mathbf{x} = \mathbf{u}$  est un vecteur propre de  $\mathbb{B}$ , il ne peut donc être nul! On a une contradiction et donc, l'hypothèse de départ est fausse: on a démontré que  $\lambda \neq 1$ .

Comme par hypothèse, la matrice  $\mathbb{M}^* + \mathbb{N}$  (hermitienne) est définie positive et  $\mathbf{u} \neq 0$ , on obtient

$$|1 - \lambda|^2 \langle \mathbf{u}, (\mathbb{M}^* + \mathbb{N})\mathbf{u} \rangle > 0.$$

On déduit alors de (R2.6) que

$$(1 - |\lambda|^2) \langle \mathbf{u}, \mathbb{A}\mathbf{u} \rangle > 0.$$

Comme  $A$  est hermitienne définie positive et  $\mathbf{u} \neq 0$ , on a

$$\langle \mathbf{u}, A\mathbf{u} \rangle > 0$$

et donc  $1 - |\lambda|^2 > 0$ , c'est à dire  $|\lambda| < 1$ .

**R. 4**

Pour une matrice quelconque  $\langle \mathbf{x}, A\mathbf{x} \rangle \in \mathbb{C}$ , or, comme  $A$  est hermitienne, on a  $\langle \mathbf{x}, A\mathbf{x} \rangle \in \mathbb{R}$ . En effet, par propriété du produit scalaire on a

$$\langle \mathbf{x}, A\mathbf{x} \rangle = \langle A^* \mathbf{x}, \mathbf{x} \rangle = \langle A\mathbf{x}, \mathbf{x} \rangle = \overline{\langle \mathbf{x}, A\mathbf{x} \rangle}.$$

On veut démontrer par l'absurde que, sous les hypothèses  $M^* + N$  (hermitienne) définie positive,  $A$  hermitienne inversible et  $\rho(B) < 1$ , on a  $A$  définie positive.

On suppose que  $A$  hermitienne n'est pas définie positive, c'est à dire qu'il existe  $\mathbf{x}^{[0]} \in \mathbb{C}^n \setminus \{0\}$  tel que  $\alpha_0 \stackrel{\text{def}}{=} \langle \mathbf{x}^{[0]}, A\mathbf{x}^{[0]} \rangle \leq 0$ .

On définit alors les suites

$$\forall k \in \mathbb{N}^*, \mathbf{x}^{[k]} = B\mathbf{x}^{[k-1]} \quad \text{et} \quad \alpha_k = \langle \mathbf{x}^{[k]}, A\mathbf{x}^{[k]} \rangle.$$

On a alors

$$\mathbf{x}^{[k]} = B^k \mathbf{x}^{[0]}, \quad \forall k \in \mathbb{N}.$$

D'après le théorème du cours (Théorème B.72 page 195),

$$\rho(B) < 1 \iff \lim_{k \rightarrow +\infty} B^k \mathbf{v} = 0, \quad \forall \mathbf{v} \in \mathbb{C}^n.$$

On a donc

$$\lim_{k \rightarrow +\infty} \mathbf{x}^{[k]} = 0 \quad \text{et} \quad \lim_{k \rightarrow +\infty} \alpha_k = 0$$

On utilise maintenant l'égalité (3.3) avec  $\mathbf{x} = \mathbf{x}^{[k-1]}$  et  $\mathbf{y} = B\mathbf{x} = \mathbf{x}^{[k]}$

$$\langle \mathbf{x}^{[k-1]}, A\mathbf{x}^{[k-1]} \rangle - \langle \mathbf{x}^{[k]}, A\mathbf{x}^{[k]} \rangle = \langle (\mathbf{x}^{[k-1]} - \mathbf{x}^{[k]}), (M^* + N)(\mathbf{x}^{[k-1]} - \mathbf{x}^{[k]}) \rangle$$

Montrons, par récurrence sur  $k \in \mathbb{N}^*$ , que

$$\mathbf{x}^{[k]} \neq \mathbf{0}, \quad \mathbf{x}^{[k]} - \mathbf{x}^{[k-1]} \neq \mathbf{0}, \quad \text{et} \quad 0 \geq \alpha_{k-1} > \alpha_k$$

- **Initialisation :** On a  $\mathbf{x}^{[0]} \neq 0$  et  $\mathbf{x}^{[1]} = B\mathbf{x}^{[0]}$ .  
On montre par l'absurde que  $\mathbf{x}^{[1]} \neq 0$ . Supposons  $\mathbf{x}^{[1]} = 0$ , alors  $\alpha_1 = 0$  et  $\mathbf{x}^{[0]} - \mathbf{x}^{[1]} = \mathbf{x}^{[0]} \neq 0$ . Comme  $M^* + N$  est hermitienne définie positive on obtient

$$\alpha_0 - \alpha_1 = \langle (\mathbf{x}^{[0]} - \mathbf{x}^{[1]}), (M^* + N)(\mathbf{x}^{[0]} - \mathbf{x}^{[1]}) \rangle > 0$$

et contradiction avec  $\alpha_0 \leq 0$ .

On montre ensuite par l'absurde que  $\mathbf{x}^{[0]} \neq \mathbf{x}^{[1]}$ . Supposons  $\mathbf{x}^{[1]} = \mathbf{x}^{[0]}$ . Par construction  $\mathbf{x}^{[1]} = B\mathbf{x}^{[0]}$ , et dans ce cas, comme  $\mathbf{x}^{[0]} \neq 0$ ,  $(1, \mathbf{x}^{[0]})$  serait un élément propre de  $B$ : contradiction avec  $\rho(B) < 1$ .

Comme  $\mathbf{x}^{[0]} - \mathbf{x}^{[1]} \neq 0$ , on a

$$\alpha_0 - \alpha_1 = \langle (\mathbf{x}^{[0]} - \mathbf{x}^{[1]}), (M^* + N)(\mathbf{x}^{[0]} - \mathbf{x}^{[1]}) \rangle > 0$$

et donc  $0 \geq \alpha_0 > \alpha_1$ .

- **Hérédité :** On suppose la propriété vraie jusqu'au rang  $k$ . On a alors  $\mathbf{x}^{[k]} \neq 0$ ,  $\mathbf{x}^{[k+1]} = B\mathbf{x}^{[k]}$  et  $\alpha_k \leq 0$ .  
On montre par l'absurde que  $\mathbf{x}^{[k+1]} \neq 0$ . Supposons  $\mathbf{x}^{[k+1]} = 0$ , alors  $\alpha_{k+1} = 0$  et  $\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]} = \mathbf{x}^{[k]} \neq 0$ . Comme  $M^* + N$  est hermitienne définie positive on obtient

$$\alpha_k - \alpha_{k+1} = \langle (\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]}), (M^* + N)(\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]}) \rangle > 0$$

et contradiction avec  $\alpha_k \leq 0$ .

On montre ensuite par l'absurde que  $\mathbf{x}^{[k]} \neq \mathbf{x}^{[k+1]}$ . Supposons  $\mathbf{x}^{[k+1]} = \mathbf{x}^{[k]}$ . Par construction  $\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]}$ , et dans ce cas, comme  $\mathbf{x}^{[k]} \neq 0$ ,  $(1, \mathbf{x}^{[k]})$  serait un élément propre de  $\mathbb{B}$ : contradiction avec  $\rho(\mathbb{B}) < 1$ .

Comme  $\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]} \neq 0$ , on a

$$\alpha_k - \alpha_{k+1} = \langle (\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]}), (\mathbb{M}^* + \mathbb{N})(\mathbf{x}^{[k]} - \mathbf{x}^{[k+1]}) \rangle > 0$$

et donc  $0 \geq \alpha_k > \alpha_{k+1}$ .

- **Conclusion** : la proposition est vérifiée pour tout  $k \in \mathbb{N}^*$ .

La suite  $(\alpha_k)_{k \in \mathbb{N}}$  est donc strictement décroissante de premier terme  $\alpha_0 \leq 0$ : elle ne peut converger vers 0. On a donc une contradiction

~~~~~ Fin correction ~~~~~

**Théorème 3.4.4.** Soit  $A$  une matrice hermitienne définie positive, alors la méthode S.O.R. converge si et seulement si  $w \in ]0, 2[$ .

*Preuve.* voir [7], vol.2, Corollaire 24, page 351. □

### 3.4.4 Algorithmes

Nous allons écrire des algorithmes pour chacune des méthodes itératives proposées. L'objectif n'est pas d'écrire des algorithmes "optimisés" mais de voir la méthodologie permettant la construction d'algorithmes simples et fonctionnels.

#### Principe de base

Les méthodes itératives pour la résolution d'un système linéaire  $A\mathbf{x} = \mathbf{b}$  s'écrivent sous la forme

$$\mathbf{x}^{[0]} \in \mathbb{K}^n \quad \text{et} \quad \mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

La matrice d'itération  $\mathbb{B}$  et le vecteur  $\mathbf{c}$  sont construits de telle sorte que si la suite  $(\mathbf{x}^{[k]})_{k \in \mathbb{N}}$  converge vers  $\bar{\mathbf{x}}$  alors  $\bar{\mathbf{x}}$  est aussi solution du système linéaire  $A\mathbf{x} = \mathbf{b}$ .

Comme pour les méthodes de point fixe, La convergence de ces méthodes n'est pas assurée et si il y a convergence le nombre d'itération nécessaire n'est (à priori) pas connu. C'est pourquoi algorithmiquement on utilise une boucle **Tantque**. On définit alors un **nombre maximum d'itérations** au delà du quel les calculs itératifs sont stoppés et une valeur  $\varepsilon > 0$  permettant d'arrêter les calculs lorsque  $\mathbf{x}^{[k]}$  est suffisamment proche de  $\bar{\mathbf{x}}$ . Pour être plus précis, on note  $\mathbf{r}^{[k]} = \mathbf{b} - A\mathbf{x}^{[k]}$  le résidu. On a alors

$$\mathbf{r}^{[k]} = A\bar{\mathbf{x}} - A\mathbf{x}^{[k]} = A\mathbf{e}^{[k]}$$

Si on prend comme critère d'arrêt

$$\frac{\|\mathbf{r}^{[k]}\|}{\|\mathbf{b}\|} \leq \varepsilon$$

alors

$$\|\mathbf{e}^{[k]}\| = \|A^{-1}\mathbf{r}^{[k]}\| \leq \|A^{-1}\| \|\mathbf{r}^{[k]}\| \leq \varepsilon \|A^{-1}\| \|\mathbf{b}\| = \varepsilon \|A^{-1}\| \|A\bar{\mathbf{x}}\| \leq \varepsilon \|A^{-1}\| \|A\| \|\bar{\mathbf{x}}\| = \varepsilon \text{cond}(A) \|\bar{\mathbf{x}}\|$$

et donc

$$\frac{\|\mathbf{e}^{[k]}\|}{\|\bar{\mathbf{x}}\|} \leq \varepsilon \text{cond}(A)$$

Pour éviter des soucis lorsque  $\|\mathbf{b}\|$  est proche de zéro, on peut utiliser comme critère d'arrêt de convergence

$$\frac{\|\mathbf{r}^{[k]}\|}{\|\mathbf{b}\| + 1} \leq \varepsilon.$$

Comme vu avec les méthodes itératives de type point fixe (voir section 2.2.3, page 30), il n'est pas forcément utile de stocker l'intégralité des termes de la suite  $\mathbf{x}^{[k]}$  puisque seul le "dernier" nous intéresse. On a alors l'Algorithme 3.19 correspondant à un algorithme itératif générique sans stockage des valeurs intermédiaires.

**Algorithm 3.19** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

**Données :**

- $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,
- $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,
- $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,
- $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,
- kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

- $\mathbf{x}^{\text{tol}}$  : un vecteur de  $\mathbb{K}^n$  si convergence, sinon  $\emptyset$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$  ▷  $\mathbf{p}$  contient le vecteur précédent
7:    $\mathbf{x} \leftarrow$  calcul de l'itérée suivante en fonction de  $\mathbf{p}, \mathbb{A}, \mathbf{b}, \dots$ 
8:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
9: Fin
10: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors ▷ Convergence
11:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
12: Fin

```

### Méthode de Jacobi

Pour Jacobi, la suite des itérées est définie par

$$x_i^{[k+1]} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} x_j^{[k]} \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

Cette formule donne explicitement les composantes du vecteur  $\mathbf{x}^{[k+1]}$  en fonction de la matrice  $\mathbb{A}$ , et des vecteurs  $\mathbf{b}$  et  $\mathbf{x}^{[k]}$ . A partir de l'Algorithme 3.19, on va construire par raffinements successifs la `RSLJacobi` donnée dans l'Algorithme 3.20.

**Algorithme 3.20**  $\mathcal{R}_0$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:
8:    $\mathbf{x} \leftarrow$  calcul par Jacobi
9:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
10: Fin
11: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors ▷ Convergence
12:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13: Fin

```

**Algorithme 3.20**  $\mathcal{R}_1$

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} p_j \right)$ 
9:   Fin
10:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
11: Fin
12: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors ▷ Convergence
13:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
14: Fin

```

Algorithme 3.20  $\mathcal{R}_1$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:     
$$x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} p_j \right)$$

9:   Fin
10:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
11: Fin
12: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
13:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
14: Fin

```

Algorithme 3.20  $\mathcal{R}_2$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $S \leftarrow 0$ 
9:     Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
10:       $S \leftarrow S + A_{i,j} p_j$ 
11:     Fin
12:      $x_i \leftarrow \frac{1}{A_{ii}} (b_i - S)$ 
13:   Fin
14:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
15: Fin
16: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
17:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
18: Fin

```

On peut alors écrire la fonction `RSLJacobi` :

---

**Algorithm 3.20** Méthode itérative de Jacobi pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,  
 $\text{kmax}$  : nombre maximum d'itérations,  $\text{kmax} \in \mathbb{N}^*$

**Résultat :**

$\mathbf{X}$  : un vecteur de  $\mathbb{K}^n$

```

1: Fonction  $\mathbf{X} \leftarrow \text{RSLJacobi}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
4:    $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5:   Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{p} \leftarrow \mathbf{x}$ 
8:     Pour  $i \leftarrow 1$  à  $n$  faire
9:        $S \leftarrow 0$ 
10:      Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
11:         $S \leftarrow S + A(i, j) * p(j)$ 
12:      Fin
13:       $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
14:    Fin
15:     $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
16:  Fin
17:  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
18:     $\mathbf{X} \leftarrow \mathbf{x}$ 
19:  Fin
20: Fin

```

---

## Méthode de Gauss-Seidel

Pour Gauss-Seidel, la suite des itérées est définie par

$$x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) \quad \forall i \in \llbracket 1, n \rrbracket$$

Cette formule donne explicitement la composante  $i$  du vecteur  $\mathbf{x}^{[k+1]}$  en fonction de la matrice  $\mathbb{A}$ , et des vecteurs  $\mathbf{b}$  et  $\mathbf{x}^{[k]}$ , mais aussi des  $i - 1$  premières composantes de  $\mathbf{x}^{[k+1]}$ . Contrairement à la méthode de Jacobi, il est impératif de calculer successivement  $x_1^{[k+1]}$ ,  $x_2^{[k+1]}$ , ... A partir de l'Algorithme 3.19, on va construire par raffinements successifs la **RSLGaussSeidel** donnée dans l'Algorithme 3.21.

Algorithme 3.21  $\mathcal{R}_0$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:
8:    $\mathbf{x} \leftarrow$  calcul par Gauss-Seidel
9:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
10: Fin
11: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13: Fin

```

Algorithme 3.21  $\mathcal{R}_1$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:
8:   Pour  $i \leftarrow 1$  à  $n$  faire
9:      $x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} p_j \right)$ 
10:   Fin
11:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
12: Fin
13: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
14:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
15: Fin

```

Algorithme 3.21  $\mathcal{R}_1$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $x_i \leftarrow \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} p_j \right)$ 
9:   Fin
10:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
11: Fin
12: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
13:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
14: Fin

```

Algorithme 3.21  $\mathcal{R}_2$ 

```

1:  $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
2:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
3:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
4: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
5:    $k \leftarrow k + 1$ 
6:    $\mathbf{p} \leftarrow \mathbf{x}$ 
7:   Pour  $i \leftarrow 1$  à  $n$  faire
8:      $S \leftarrow 0$ 
9:     Pour  $j \leftarrow 1$  à  $i - 1$  faire
10:       $S \leftarrow S + A_{i,j} x_j$ 
11:     Fin
12:     Pour  $j \leftarrow i + 1$  à  $n$  faire
13:       $S \leftarrow S + A_{i,j} p_j$ 
14:     Fin
15:      $x_i \leftarrow \frac{1}{A_{ii}} (b_i - S)$ 
16:   Fin
17:    $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x}$ ,
18: Fin
19: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
20:    $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
21: Fin

```

On peut alors écrire la fonction **RSLGaussSeidel** :

---

**Algorithm 3.21** Méthode itérative de Gauss-Seidel pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**

$\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$  ,  
 $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$  ,  
 $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$  ,  
kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

$\mathbf{X}$  : un vecteur de  $\mathbb{K}^n$

```

1: Fonction  $\mathbf{X} \leftarrow \text{RSLGaussSeidel}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:  $k \leftarrow 0, \mathbf{X} \leftarrow \emptyset$ 
3:  $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
4:  $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5: Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:    $k \leftarrow k + 1$ 
7:    $\mathbf{p} \leftarrow \mathbf{x}$ 
8:   Pour  $i \leftarrow 1$  à  $n$  faire
9:      $S \leftarrow 0$ 
10:    Pour  $j \leftarrow 1$  à  $i - 1$  faire
11:       $S \leftarrow S + A(i, j) * x(j)$ 
12:    Fin
13:    Pour  $j \leftarrow i + 1$  à  $n$  faire
14:       $S \leftarrow S + A(i, j) * p(j)$ 
15:    Fin
16:     $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
17:  Fin
18:   $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A} * \mathbf{x}$ ,
19: Fin
20: Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
21:    $\mathbf{X} \leftarrow \mathbf{x}$ 
22: Fin
23: Fin

```

---

### Jeux algorithmiques

On a bien sûr noté que les fonctions `RSLJacobi` et `RSLGaussSeidel` ont la même ossature puisque toutes deux basées sur l'Algorithme 3.19 générique. En effet seule la transcription de la ligne 7 de l'Algorithme 3.19 diffère :

```

Fonction  $X \leftarrow \text{RSLJacobi}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
  Pour  $i \leftarrow 1$  à  $n$  faire
     $S \leftarrow 0$ 
    Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
       $S \leftarrow S + A(i, j) * p(j)$ 
    Fin Pour
     $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
  Fin Pour
   $\mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
Fin
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin
Fin

```

```

Fonction  $X \leftarrow \text{RSLGaussSeidel}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
  Pour  $i \leftarrow 1$  à  $n$  faire
     $S \leftarrow 0$ 
    Pour  $j \leftarrow 1$  à  $i - 1$  faire
       $S \leftarrow S + A(i, j) * x(j)$ 
    Fin Pour
    Pour  $j \leftarrow i + 1$  à  $n$  faire
       $S \leftarrow S + A(i, j) * p(j)$ 
    Fin Pour
     $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
  Fin Pour
   $\mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
Fin
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin
Fin

```

On va écrire les fonctions algorithmiques `IterJacobi` et `IterGaussSeidel` permettant le calcul d'une itérée respectivement pour les méthodes de Jacobi et Gauss-Seidel (voir Algorithmes 3.22 et 3.23).

**Algorithm 3.22** Itération de Jacobi : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij} y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Données :**

$A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,  
 $\mathbf{y}$  : vecteur de  $\mathbb{K}^n$ ,

**Résultat :**

$\mathbf{x}$  : un vecteur de  $\mathbb{K}^n$

```

1: Fonction  $\mathbf{x} \leftarrow \text{IterJacobi}(A, \mathbf{b}, \mathbf{y})$ 
2: Pour  $i \leftarrow 1$  à  $n$  faire
3:    $S \leftarrow 0$ 
4:   Pour  $j \leftarrow 1$  à  $n$  ( $j \neq i$ ) faire
5:      $S \leftarrow S + A(i, j) * y(j)$ 
6:   Fin
7:    $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
8: Fin
9: Fin

```

**Algorithm 3.23** Itération de Gauss-Seidel : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j - \sum_{j=i+1}^n A_{ij} y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Données :**

$A$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,  
 $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,  
 $\mathbf{y}$  : vecteur de  $\mathbb{K}^n$ ,

**Résultat :**

$\mathbf{x}$  : un vecteur de  $\mathbb{K}^n$

```

1: Fonction  $\mathbf{x} \leftarrow \text{IterGaussSeidel}(A, \mathbf{b}, \mathbf{y})$ 
2: Pour  $i \leftarrow 1$  à  $n$  faire
3:    $S \leftarrow 0$ 
4:   Pour  $j \leftarrow 1$  à  $i - 1$  faire
5:      $S \leftarrow S + A(i, j) * x(j)$ 
6:   Fin
7:   Pour  $j \leftarrow i + 1$  à  $n$  faire
8:      $S \leftarrow S + A(i, j) * y(j)$ 
9:   Fin
10:   $x(i) \leftarrow (b(i) - S) / A(i, i)$ 
11: Fin
12: Fin

```

En utilisant ces deux fonctions, les algorithmes de résolutions de systèmes linéaires par les méthodes de Jacobi et Gauss-Seidel peuvent se réécrire sous la forme suivante :

```

Fonction  $X \leftarrow \text{RSLJacobiV2}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
   $\mathbf{x} \leftarrow \text{IterJacobi}(A, \mathbf{b}, \mathbf{p})$ 
   $\mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
Fin
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin
Fin

```

```

Fonction  $X \leftarrow \text{RSLGaussSeidelV2}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
 $k \leftarrow 0, X \leftarrow \emptyset$ 
 $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
 $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
   $k \leftarrow k + 1$ 
   $\mathbf{p} \leftarrow \mathbf{x}$ 
   $\mathbf{x} \leftarrow \text{IterGaussSeidel}(A, \mathbf{b}, \mathbf{p})$ 
   $\mathbf{r} \leftarrow \mathbf{b} - A * \mathbf{x},$ 
Fin
Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
   $X \leftarrow \mathbf{x}$ 
Fin
Fin

```

En programmation, dès que l'on commence à faire des copier/coller<sup>1</sup> il faut se poser la question : est-il possible de faire sans?

La plupart du temps la réponse est oui, et cela permet souvent de simplifier, clarifier et raccourcir le code ce qui simplifie grandement sa maintenance. Dans notre cas, on écrit l'Algorithme générique 3.19 sous forme d'une fonction à laquelle on ajoute aux paramètres d'entrées une fonction formelle `IterFonc` permettant le calcul d'une itérée :

$$\mathbf{x} \leftarrow \text{IterFonc}(\mathbb{A}, \mathbf{b}, \mathbf{y}).$$

Cette fonction est donc une donnée du problème. On a alors

---

**Algorithm 3.24** Méthode itérative pour la résolution d'un système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$

---

**Données :**

- $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,
- $\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ ,
- `IterFonc` : fonction de paramètres une matrice d'ordre  $n$ ,  
: et deux vecteurs de  $\mathbb{K}^n$ . retourne un vecteur de  $\mathbb{K}^n$ .
- $\mathbf{x}^0$  : vecteur initial de  $\mathbb{K}^n$ ,
- $\varepsilon$  : la tolérance,  $\varepsilon \in \mathbb{R}^+$ ,
- kmax : nombre maximum d'itérations, kmax  $\in \mathbb{N}^*$

**Résultat :**

- $\mathbf{x}^{\text{tol}}$  : un vecteur de  $\mathbb{K}^n$  si convergence, sinon  $\emptyset$

```

1: Fonction  $\mathbf{X} \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterFonc}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
2:    $k \leftarrow 0, \mathbf{x}^{\text{tol}} \leftarrow \emptyset$ 
3:    $\mathbf{x} \leftarrow \mathbf{x}^0, \mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
4:    $\text{tol} \leftarrow \varepsilon(\|\mathbf{b}\| + 1)$ 
5:   Tantque  $\|\mathbf{r}\| > \text{tol}$  et  $k \leq \text{kmax}$  faire
6:      $k \leftarrow k + 1$ 
7:      $\mathbf{p} \leftarrow \mathbf{x}$ 
8:      $\mathbf{x} \leftarrow \text{IterFonc}(\mathbb{A}, \mathbf{b}, \mathbf{p})$ 
9:      $\mathbf{r} \leftarrow \mathbf{b} - \mathbb{A}\mathbf{x},$ 
10:  Fin
11:  Si  $\|\mathbf{r}\| \leq \text{tol}$  alors
12:     $\mathbf{x}^{\text{tol}} \leftarrow \mathbf{x}$ 
13:  Fin
14: Fin

```

---

En utilisant cette fonction, les algorithmes de résolutions de systèmes linéaires par les méthodes de Jacobi et Gauss-Seidel peuvent se réécrire sous la forme suivante :

```

Fonction  $\mathbf{X} \leftarrow \text{RSLJacobiV3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
   $\mathbf{X} \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterJacobi}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
Fin

```

```

Fonction  $\mathbf{X} \leftarrow \text{RSLGaussSeidelV3}(\mathbb{A}, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
   $\mathbf{X} \leftarrow \text{RSLMethIter}(\mathbb{A}, \mathbf{b}, \text{IterGaussSeidel}, \mathbf{x}^0, \varepsilon, \text{kmax})$ 
Fin

```

### Méthode S.O.R.

Pour la méthode de relaxation utilisant Gauss-Seidel, avec  $w \in \mathbb{R}^*$ , la suite des itérées est définie par

$$x_i^{[k+1]} = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{[k+1]} - \sum_{j=i+1}^n A_{ij} x_j^{[k]} \right) + (1-w)x_i^{[k]} \quad \forall i \in \llbracket 1, n \rrbracket$$

---

<sup>1</sup>Opération qui pour un bon programmeur est une chose très fatigante!

**Algorithm 3.25** Itération S.O.R. : calcul de  $\mathbf{x}$  tel que

$$x_i = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}y_j \right) + (1-w)y_i$$

**Données :**

- A : matrice de  $\mathcal{M}_n(\mathbb{K})$ ,
- b : vecteur de  $\mathbb{K}^n$ ,
- y : vecteur de  $\mathbb{K}^n$ ,
- w : réel non nul.

**Résultat :**

- x : un vecteur de  $\mathbb{K}^n$

```

1: Fonction x ← IterSOR( A, b, y, w )
2: Pour i ← 1 à n faire
3:   S ← 0
4:   Pour j ← 1 à i - 1 faire
5:     S ← S + A(i, j) * x(j)
6:   Fin
7:   Pour j ← i + 1 à n faire
8:     S ← S + A(i, j) * y(j)
9:   Fin
10:  x(i) ← w * (b(i) - S) / A(i, i) + (1 - w) * y(i)
11: Fin
12: Fin

```

```

Fonction X ← RLSORV3( A, b, w, x0, ε, kmax )
  IterFun ← ((M, r, s) ↦ IterSOR(M, r, s, w))
  X ← RSLMethIter(A, b, IterFun, x0, ε, kmax)
Fin

```

### Une ch'tite remarque

Les méthodes itératives que l'on a étudiées sont basées sur l'expression

$$\mathbf{x}^{[k+1]} = \mathbb{B}\mathbf{x}^{[k]} + \mathbf{c}$$

Si l'on pose

$$\Phi : \mathbf{x} \mapsto \mathbb{B}\mathbf{x} + \mathbf{c}$$

alors

$$\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]}).$$

Celà vous rappelle-t'il quelque chose?

## 3.4.5 Algorithmes matriciels

Les méthodes de Jacobi, Gauss-Seidel et S.O.R. peuvent s'écrire sous la forme

$$\mathbb{M}\mathbf{x}^{[k+1]} = \mathbb{N}\mathbf{x}^{[k]} + \mathbf{b}$$

avec  $\mathbb{A} = \mathbb{M} - \mathbb{N}$  et, dans ce cas, la matrice d'itération est  $\mathbb{B} = \mathbb{M}^{-1}\mathbb{N}$ .

- Jacobi :  $\mathbb{M} = \mathbb{D}$  et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$ ,
- Gauss-Seidel :  $\mathbb{M} = \mathbb{D} - \mathbb{E}$  et  $\mathbb{N} = \mathbb{F}$ ,
- S.O.R. :  $\mathbb{M} = \frac{\mathbb{D}}{w} - \mathbb{E}$  et  $\mathbb{N} = \frac{1-w}{w}\mathbb{D} + \mathbb{F}$ .

En posant,  $\Phi(\mathbf{x}) = \mathbb{M}^{-1}(\mathbb{N}\mathbf{x} + \mathbf{b})$ , on a

$$\mathbf{x}^{[k+1]} = \Phi(\mathbf{x}^{[k]}).$$

On va donc pouvoir utiliser l'algorithme vectoriel du point fixe 2.16.

Pour celà, on note que

$$\mathbf{y} = \Phi(\mathbf{x}) \iff \mathbb{M}\mathbf{y} = \mathbb{N}\mathbf{x} + \mathbf{b}.$$

Pour calculer  $\Phi(\mathbf{x})$  on va donc résoudre un système linéaire,

- Jacobi :  $\mathbb{M} = \mathbb{D}$  (diagonale) et  $\mathbb{N} = \mathbb{E} + \mathbb{F}$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLMatDiag}(\mathbb{M}, \mathbb{N}\mathbf{x} + \mathbf{b}) \right)$$

**Fonction**  $X \leftarrow \text{RSLJacobiPF}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $[M, N] \leftarrow \text{JacobiMN}(A)$   
 $\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLMatDiag}(M, N\mathbf{x} + \mathbf{b}) \right)$   
 $X \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x}^0, \text{tol}, \text{kmax})$   
**Fin**

- Gauss-Seidel :  $M = D - E$  (tri. inf.) et  $N = F$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(M, N\mathbf{x} + \mathbf{b}) \right)$$

**Fonction**  $X \leftarrow \text{RSLGaussSeidelPF}(A, \mathbf{b}, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $[M, N] \leftarrow \text{GaussSeidelMN}(A)$   
 $\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(M, N\mathbf{x} + \mathbf{b}) \right)$   
 $X \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x}^0, \text{tol}, \text{kmax})$   
**Fin**

- S.O.R. :  $M = \frac{D}{w} - E$  (tri. inf.) et  $N = \frac{1-w}{w}D + F$ ,

$$\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(M, N\mathbf{x} + \mathbf{b}) \right).$$

**Fonction**  $X \leftarrow \text{RSLSORPF}(A, \mathbf{b}, w, \mathbf{x}^0, \varepsilon, \text{kmax})$   
 $[M, N] \leftarrow \text{SORmatMN}(A, w)$   
 $\Phi \leftarrow \left( \mathbf{x} \mapsto \text{RSLTriInf}(M, N\mathbf{x} + \mathbf{b}) \right)$   
 $X \leftarrow \text{PtFixeVec}(\Phi, \mathbf{x}^0, \text{tol}, \text{kmax})$   
**Fin**

### 3.4.6 Exercices

#### EXERCICE 3.4.3

**Q. 1** Etudier la convergence des méthodes de Jacobi et Gauss-Seidel pour la matrice

$$A_1 = \begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}$$

**Q. 2** Etudier la convergence des méthodes de Jacobi et Gauss-Seidel pour la matrice

$$A_2 = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}$$

#### Correction

**R. 1**

La matrice  $A_1$  s'écrit sous la forme  $A_1 = D_1 - E_1 - F_1$  où  $D_1 = \text{diag}(A_1)$ ,  $E_1$  est triangulaire inférieure et d'éléments nuls sur la diagonale et  $F_1$  est triangulaire supérieure et d'éléments nuls sur la diagonale.

On a donc  $E_1 = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & -2 & 0 \end{pmatrix}$ ,  $F_1 = \begin{pmatrix} 0 & -2 & 2 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}$  et  $D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ . On note que  $D_1$  est inversible.

- Pour la méthode de Jacobi, de matrice d'itération  $\mathbb{J}_1 = \mathbb{D}_1^{-1}(\mathbb{E}_1 + \mathbb{F}_1)$ , on a convergence si et seulement si  $\rho(\mathbb{J}_1) < 1$ , c'est à dire si les modules de valeurs propres de  $\mathbb{J}_1$  sont strictement plus petit que 1.

On a

$$\mathbb{J}_1 = \begin{pmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{pmatrix}$$

Les valeurs propres de  $\mathbb{J}_1$  sont les racines du polynôme caractéristique donné par

$$\begin{aligned} P(\lambda) &= \det(\lambda \mathbb{I} - \mathbb{J}_1) \\ &= \det \begin{pmatrix} \lambda & 2 & -2 \\ 1 & \lambda & 1 \\ 2 & 2 & \lambda \end{pmatrix} \\ &= \lambda^3 \end{aligned}$$

Ses racines étant  $(0, 0, 0)$ , on a  $\rho(\mathbb{J}_1) = 0$  et donc la méthode de Jacobi converge.

- Pour la méthode de Gauss-Seidel, de matrice d'itération  $\mathbb{B}_1 = (\mathbb{D}_1 - \mathbb{E}_1)^{-1}\mathbb{F}_1$ , on a convergence si et seulement si  $\rho(\mathbb{B}_1) < 1$ , c'est à dire si les modules des valeurs propres de  $\mathbb{B}_1$  sont strictement plus petit que 1.

On a

$$\mathbb{B}_1 = \begin{pmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{pmatrix}$$

Les valeurs propres de  $\mathbb{B}_1$  sont les racines du polynôme caractéristique donné par

$$\begin{aligned} P(\lambda) &= \det(\lambda \mathbb{I} - \mathbb{B}_1) \\ &= \det \begin{pmatrix} \lambda & 2 & -2 \\ 0 & \lambda - 2 & 3 \\ 0 & 0 & \lambda - 2 \end{pmatrix} \\ &= \lambda^3 - 4\lambda^2 + 4\lambda \end{aligned}$$

Ses racines étant  $(2, 2, 0)$ , on a  $\rho(\mathbb{B}_1) = 2$  et donc la méthode de Gauss-Seidel diverge.

### R. 2

La matrice  $\mathbb{A}_2$  s'écrit sous la forme  $\mathbb{A}_2 = \mathbb{D}_2 - \mathbb{E}_2 - \mathbb{F}_2$  où  $\mathbb{D}_2 = \text{diag}(\mathbb{A}_2)$ ,  $\mathbb{E}_2$  est triangulaire inférieure et d'éléments nuls sur la diagonale et  $\mathbb{F}_2$  est triangulaire supérieure et d'éléments nuls sur la diagonale.

On a donc  $\mathbb{E}_2 = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ ,  $\mathbb{F}_2 = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$  et  $\mathbb{D}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ . On note que  $\mathbb{D}_2$  est inversible.

- Pour la méthode de Jacobi, de matrice d'itération  $\mathbb{J}_2 = \mathbb{D}_2^{-1}(\mathbb{E}_2 + \mathbb{F}_2)$ , on a convergence si et seulement si  $\rho(\mathbb{J}_2) < 1$ , c'est à dire si les modules des valeurs propres de  $\mathbb{J}_2$  sont strictement plus petit que 1.

On a

$$\mathbb{J}_2 = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & 0 & -1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

Les valeurs propres de  $\mathbb{J}_2$  sont les racines du polynôme caractéristique donné par

$$\begin{aligned} P(\lambda) &= \det(\lambda \mathbb{I} - \mathbb{J}_2) \\ &= \det \begin{pmatrix} \lambda & -\frac{1}{2} & \frac{1}{2} \\ 1 & \lambda & 1 \\ -\frac{1}{2} & -\frac{1}{2} & \lambda \end{pmatrix} \\ &= t^3 + \frac{5}{4}t \end{aligned}$$

Ses racines étant  $(-\frac{1}{2}i\sqrt{5}, \frac{1}{2}i\sqrt{5}, 0)$ , on a  $\rho(\mathbb{J}_2) = \frac{1}{2}\sqrt{5}$  et donc la méthode de Jacobi diverge.

- Pour la méthode de Gauss-Seidel, de matrice d'itération  $\mathbb{B}_2 = (\mathbb{D}_2 - \mathbb{E}_2)^{-1}\mathbb{F}_2$ , on a convergence si et seulement si  $\rho(\mathbb{B}_2) < 1$ , c'est à dire si les modules des valeurs propres de  $\mathbb{B}_2$  sont strictement plus petit que 1.

On a

$$\mathbb{B}_2 = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

Les valeurs propres de  $\mathbb{B}_2$  sont les racines du polynôme caractéristique donné par

$$\begin{aligned} P(\lambda) &= \det(\lambda\mathbb{I} - \mathbb{B}_2) \\ &= \det \begin{pmatrix} \lambda & -\frac{1}{2} & \frac{1}{2} \\ 0 & \lambda + \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \lambda + \frac{1}{2} \end{pmatrix} \\ &= \lambda^3 + \lambda^2 + \frac{1}{4}\lambda \end{aligned}$$

Ses racines étant  $(-\frac{1}{2}, -\frac{1}{2}, 0)$ , on a  $\rho(\mathbb{B}_2) = \frac{1}{2}$  et donc la méthode de Gauss-Seidel converge.

~~~~~ Fin correction ~~~~~

#### EXERCICE 3.4.4

On note  $\mathbb{T} \in \mathcal{M}_n(\mathbb{C})$  la matrice tridiagonale

$$\mathbb{T} = \begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_2 & a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & 0 & b_n & a_n \end{pmatrix}. \quad (3.1)$$

Q. 1

Soit  $\mu \in \mathbb{C}^*$ . On note  $\mathbb{Q}(\mu) \in \mathcal{M}_n(\mathbb{C})$  la matrice diagonale de diagonale  $(\mu, \mu^2, \dots, \mu^n)$ .

- Expliciter la matrice  $\mathbb{T}(\mu) \stackrel{\text{def}}{=} \mathbb{Q}(\mu)\mathbb{T}\mathbb{Q}^{-1}(\mu)$  en fonction des coefficients tridiagonaux de la matrice  $\mathbb{T}$  et de  $\mu$ .
- Déterminer  $\det(\mathbb{T}(\mu))$  en fonction de  $\det(\mathbb{T})$ .

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  une matrice inversible dont les éléments diagonaux sont non nuls. On note  $A_{i,j}$  la composante  $(i, j)$  de la matrice  $\mathbb{A}$ . On décompose la matrice  $\mathbb{A}$  sous la forme  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$ , où  $\mathbb{D}$  représente la diagonale de  $\mathbb{A}$ ,  $-\mathbb{E}$  la partie triangulaire inférieure stricte et  $-\mathbb{F}$  la partie triangulaire supérieure stricte.

On note respectivement  $\mathbb{J} \stackrel{\text{def}}{=} \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})$  et  $\mathcal{L}_1 \stackrel{\text{def}}{=} (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}$  les matrices d'itérations des méthodes de Jacobi et de Gauss-Seidel.

Soit  $\mathbf{y} \in \mathbb{R}^n$ . On souhaite résoudre le système  $\mathbb{A}\mathbf{x} = \mathbf{y}$  par la méthode de Gauss-Seidel ou par la méthode de Jacobi.

On suppose dans la suite que la matrice inversible  $\mathbb{A} \in \mathcal{M}_n(\mathbb{R})$  s'écrit sous la forme

$$\mathbb{A} = \begin{pmatrix} \alpha_1 & \nu_1 & 0 & \dots & 0 \\ \beta_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \nu_{n-1} \\ 0 & \dots & 0 & \beta_n & \alpha_n \end{pmatrix} \quad (3.2)$$

et que ses éléments diagonaux sont non nuls.

Q. 2

a. Montrer que les valeurs propres de  $\mathbb{J}$  sont les racines du polynôme

$$q_{\mathbb{J}}(\lambda) \stackrel{\text{def}}{=} \det(\lambda \mathbb{D} - \mathbb{E} - \mathbb{F}).$$

b. En utilisant la question 1, montrer que  $q_{\mathbb{J}}(\lambda) = \det(\lambda \mathbb{D} - \lambda \mathbb{E} - \frac{1}{\lambda} \mathbb{F})$ .

c. En déduire que si  $\lambda \in \mathbb{C}$  est valeur propre de  $\mathbb{J}$  alors  $-\lambda$  l'est aussi.

Q. 3

a. Montrer que les valeurs propres de  $\mathcal{L}_1$  sont les racines du polynôme

$$q_{\mathcal{L}_1}(\lambda) \stackrel{\text{def}}{=} \det(\lambda \mathbb{D} - \lambda \mathbb{E} - \mathbb{F}).$$

b. En déduire que

$$\forall \lambda \in \mathbb{C}^*, \quad q_{\mathcal{L}_1}(\lambda^2) = \lambda^n q_{\mathbb{J}}(\lambda). \quad (3.3)$$

Q. 4

a. Comparer les valeurs propres de  $\mathbb{J}$  à celles de  $\mathcal{L}_1$ .

b. Une des deux méthodes est-elle à privilégier dans ce cas?

## Correction

R. 1

a. On peut noter que la matrice  $\mathbb{Q}(\mu)$  est inversible car elle est diagonale et  $\mu \in \mathbb{C}^*$ . Son inverse est la matrice diagonale de diagonale  $(\mu^{-1}, \mu^{-2}, \dots, \mu^{-n})$ .

**1ère démonstration.** On a

$$\begin{aligned} \mathbb{T}(\mu) &= \begin{pmatrix} \mu & 0 & \dots & 0 \\ 0 & \mu^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mu^n \end{pmatrix} \mathbb{T} \begin{pmatrix} \frac{1}{\mu} & 0 & \dots & 0 \\ 0 & \frac{1}{\mu^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{\mu^n} \end{pmatrix} \\ &= \begin{pmatrix} \mu a_1 & \mu c_1 & 0 & \dots & 0 \\ \mu^2 b_2 & \mu^2 a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \mu^{n-1} c_{n-1} \\ 0 & \dots & 0 & \mu^n b_n & \mu^n a_n \end{pmatrix} \begin{pmatrix} \frac{1}{\mu} & 0 & \dots & 0 \\ 0 & \frac{1}{\mu^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{\mu^n} \end{pmatrix} \\ &= \begin{pmatrix} a_1 & \mu^{-1} c_1 & 0 & \dots & 0 \\ \mu b_2 & a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \mu^{-1} c_{n-1} \\ 0 & \dots & 0 & \mu b_n & a_n \end{pmatrix} \end{aligned}$$

**2ème démonstration.** Pour simplifier l'écriture, on pose  $\mathbb{Q} \stackrel{\text{def}}{=} \mathbb{Q}(\mu)$ . Soit  $(i, j) \in \llbracket 1, n \rrbracket^2$ , on a

$$(\mathbb{T}(\mu))_{i,j} = (\mathbb{Q} \mathbb{T} \mathbb{Q}^{-1})_{i,j} = \sum_{k=1}^n (\mathbb{Q}^i)_{i,k} (\mathbb{Q}^{-1})_{k,j}$$

Or  $\mathbb{Q}^{-1}$  est diagonale, donc  $(\mathbb{Q}^{-1})_{k,j} = 0$ , si  $k \neq j$ . Ceci donne

$$\begin{aligned} (\mathbb{T}(\mu))_{i,j} &= (\mathbb{Q} \mathbb{T})_{i,j} (\mathbb{Q}^{-1})_{j,j} = \mu^{-j} (\mathbb{Q} \mathbb{T})_{i,j} \\ &= \mu^{-j} \sum_{k=1}^n \mathbb{Q}_{i,k} \mathbb{T}_{k,j}. \end{aligned}$$

De même,  $\mathbb{Q}$  est diagonale, donc  $\mathbb{Q}_{i,k} = 0$ , si  $k \neq i$ . Ceci donne

$$(\mathbb{T}(\mu))_{i,j} = \mu^{-j} \mathbb{Q}_{i,i} \mathbb{T}_{i,j} = \mu^{i-j} \mathbb{T}_{i,j}.$$

La matrice  $\mathbb{T}$  étant tridiagonale,  $\mathbb{T}(\mu)$  l'est aussi et on a

$$\begin{aligned} (\mathbb{T}(\mu))_{i,i} &= \mathbb{T}_{i,i} = a_i, & \forall i \in \llbracket 1, n \rrbracket & \quad \text{(diagonale)} \\ (\mathbb{T}(\mu))_{i,i+1} &= \mu^{-1} \mathbb{T}_{i,i+1} = \mu^{-1} c_i, & \forall i \in \llbracket 1, n-1 \rrbracket & \quad \text{(sur-diagonale)} \\ (\mathbb{T}(\mu))_{i-1,i} &= \mu \mathbb{T}_{i-1,i} = \mu^{-1} b_i, & \forall i \in \llbracket 2, n \rrbracket & \quad \text{(sous-diagonale)} \end{aligned}$$

b. On a

$$\det(\mathbb{T}(\mu)) = \det(\mathbb{Q}(\mu) \mathbb{T} \mathbb{Q}^{-1}(\mu)) = \det(\mathbb{Q}(\mu)) \det(\mathbb{T}) \det(\mathbb{Q}^{-1}(\mu)) = \det(\mathbb{T}),$$

car  $\det(\mathbb{Q}(\mu)) \det(\mathbb{Q}^{-1}(\mu)) = 1$ .

### R. 2

a. Les valeurs propres de  $\mathbb{J}$  sont les racines de son polynôme caractéristique

$$P_{\mathbb{J}}(\lambda) \stackrel{\text{def}}{=} \det(\lambda \mathbb{I} - \mathbb{J}).$$

Or on a

$$\begin{aligned} P_{\mathbb{J}}(\lambda) &= \det(\lambda \mathbb{I} - \mathbb{D}^{-1}(\mathbb{E} + \mathbb{F})) \\ &= \det(\mathbb{D}^{-1}(\lambda \mathbb{D} - (\mathbb{E} + \mathbb{F}))) \\ &= \det(\mathbb{D}^{-1}) \det(\lambda \mathbb{D} - (\mathbb{E} + \mathbb{F})) \\ &= \det(\mathbb{D}^{-1}) q_{\mathbb{J}}(\lambda). \end{aligned}$$

Comme  $\det(\mathbb{D}^{-1}) \neq 0$ , les valeurs propres de  $\mathbb{J}$  sont aussi les racines de  $q_{\mathbb{J}}(\lambda)$ .

b. En reprenant les notations de la question 1, et en notant  $\mathbb{T}$  la matrice

$$\mathbb{T} = \lambda \mathbb{D} - \mathbb{E} - \mathbb{F} = \begin{pmatrix} \lambda \alpha_1 & \nu_1 & 0 & \dots & 0 \\ \beta_2 & \lambda \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \nu_{n-1} \\ 0 & \dots & 0 & \beta_n & \lambda \alpha_n \end{pmatrix}$$

la matrice  $\mathbb{T}(\lambda) \stackrel{\text{def}}{=} \mathbb{Q}(\lambda) \mathbb{T} \mathbb{Q}^{-1}(\lambda)$  correspond alors à

$$\mathbb{T}(\lambda) = \begin{pmatrix} \lambda \alpha_1 & \frac{1}{\lambda} \nu_1 & 0 & \dots & 0 \\ \lambda \beta_2 & \lambda \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{\lambda} \nu_{n-1} \\ 0 & \dots & 0 & \lambda \beta_n & \lambda \alpha_n \end{pmatrix} = \lambda \mathbb{D} - \lambda \mathbb{E} - \frac{1}{\lambda} \mathbb{F}.$$

D'après la question 1, on a  $\det(\mathbb{T}(\lambda)) = \det(\mathbb{T})$  ce qui donne

$$\det(\lambda \mathbb{D} - \lambda \mathbb{E} - \frac{1}{\lambda} \mathbb{F}) = \det(\lambda \mathbb{D} - \mathbb{E} - \mathbb{F}) = q_{\mathbb{J}}(\lambda).$$

c. Soit  $\lambda \in \mathbb{C}$  une valeur propre de  $\mathbb{J}$ . On a donc  $P_{\mathbb{J}}(\lambda) = 0$ . Or on a

$$P_{\mathbb{J}}(\lambda) = \det(\mathbb{D}^{-1}) q_{\mathbb{J}}(\lambda) = \det(\mathbb{D}^{-1}) \det(\lambda \mathbb{D} - \lambda \mathbb{E} - \frac{1}{\lambda} \mathbb{F})$$

et donc

$$\begin{aligned} P_{\mathbb{J}}(-\lambda) &= \det(\mathbb{D}^{-1}) \det(-\lambda \mathbb{D} + \lambda \mathbb{E} + \frac{1}{\lambda} \mathbb{F}) \\ &= (-1)^n \det(\mathbb{D}^{-1}) \det(\lambda \mathbb{D} - \lambda \mathbb{E} - \frac{1}{\lambda} \mathbb{F}) \\ &= (-1)^n P_{\mathbb{J}}(\lambda) \\ &= 0 \end{aligned}$$

c'est à dire  $-\lambda$  est aussi une valeur propre de  $\mathbb{J}$ .

R. 3

- a. Les valeurs propres de  $\mathcal{L}_1$  sont les racines de son polynôme caractéristique

$$P_{\mathcal{L}_1}(\lambda) \stackrel{\text{def}}{=} \det(\lambda \mathbb{I} - \mathcal{L}_1).$$

Or on a

$$\begin{aligned} P_{\mathcal{L}_1}(\lambda) &= \det(\lambda \mathbb{I} - (\mathbb{D} - \mathbb{E})^{-1} \mathbb{F}) \\ &= \det((\mathbb{D} - \mathbb{E})^{-1} (\lambda(\mathbb{D} - \mathbb{E}) - \mathbb{F})) \\ &= \det((\mathbb{D} - \mathbb{E})^{-1}) \det(\lambda(\mathbb{D} - \mathbb{E}) - \mathbb{F}) \\ &= \det((\mathbb{D} - \mathbb{E})^{-1}) q_{\mathcal{L}_1}(\lambda). \end{aligned}$$

Comme  $\det((\mathbb{D} - \mathbb{E})^{-1}) = \det(\mathbb{D}^{-1}) \neq 0$ , les valeurs propres de  $\mathcal{L}_1$  sont aussi les racines de  $q_{\mathcal{L}_1}(\lambda)$ .

- b. Soit  $\lambda \in \mathbb{C}^*$ . On a

$$\begin{aligned} q_{\mathcal{L}_1}(\lambda^2) &= \det(\lambda^2(\mathbb{D} - \mathbb{E}) - \mathbb{F}) \\ &= \det\left(\lambda\left(\lambda\mathbb{D} - \lambda\mathbb{E} - \frac{1}{\lambda}\mathbb{F}\right)\right) \\ &= \lambda^n \det\left(\lambda\mathbb{D} - \lambda\mathbb{E} - \frac{1}{\lambda}\mathbb{F}\right). \end{aligned}$$

Et donc on obtient bien (3.3).

R. 4

- a. Si  $\lambda$  est une valeur propre de  $\mathbb{J}$  alors  $\lambda^2$  est une valeur propre de  $\mathcal{L}_1$ . Si  $\mu \neq 0$  est une valeur propre de  $\mathcal{L}_1$  alors ses racines carrées complexes  $\sqrt{\mu}$  et  $-\sqrt{\mu}$  sont valeurs propres de  $\mathbb{J}$ .
- b. On a  $\rho(\mathcal{L}_1) = \rho(\mathbb{J})^2$ , et donc  $\rho(\mathcal{L}_1) < 1 \Leftrightarrow \rho(\mathbb{J}) < 1$ . Les deux méthodes convergent donc simultanément. Toutefois, lorsqu'il y a convergence, on a

$$\rho(\mathcal{L}_1) = \rho(\mathbb{J})^2 < \rho(\mathbb{J}) < 1$$

et donc, Il faut privilégier la méthode de Gauss-Seidel car une méthode itérative converge d'autant plus vite que le rayon spectral de sa matrice d'itération est petit.

~~~~~ Fin correction ~~~~~

### EXERCICE 3.4.5

Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne définie positive décomposée (par points) sous la forme  $A = \mathbb{D} - \mathbb{E} - \mathbb{F}$  où  $\mathbb{D} = \text{diag}(A)$ ,  $\mathbb{E}$  est triangulaire inférieure et d'éléments nuls sur la diagonale et  $\mathbb{F}$  est triangulaire supérieure et d'éléments nuls sur la diagonale.

On étudie une méthode itérative de résolution du système linéaire  $A\mathbf{x} = \mathbf{b}$ . Soit  $\mathbf{x}_0 \in \mathbb{C}^n$ , on définit la suite  $(\mathbf{x}_k)_{k \in \mathbb{N}}$  par

$$(\mathbb{D} - \mathbb{E})\mathbf{x}_{k+1/2} = \mathbb{F}\mathbf{x}_k + \mathbf{b} \quad (3.1)$$

$$(\mathbb{D} - \mathbb{F})\mathbf{x}_{k+1} = \mathbb{E}\mathbf{x}_{k+1/2} + \mathbf{b} \quad (3.2)$$

Q. 1

Ecrire le vecteur  $\mathbf{x}_{k+1}$  sous la forme

$$\mathbf{x}_{k+1} = \mathbb{B}\mathbf{x}_k + \mathbf{c} \quad (3.3)$$

en explicitant la matrice  $\mathbb{B}$  et le vecteur  $\mathbf{c}$ .

Q. 2

a. Montrer que

$$\mathbb{D}^{-1} = (\mathbb{D} - \mathbb{E})^{-1} - \mathbb{D}^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}. \quad (3.4)$$

b. Soit  $(\lambda, \mathbf{p})$  un élément propre de la matrice  $\mathbb{B}$ . Montrer que

$$\lambda \mathbb{A}\mathbf{p} + (\lambda - 1)\mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} = 0. \quad (3.5)$$

Q. 3

En déduire la convergence de cette méthode vers la solution  $\underline{\mathbf{x}}$  de  $\mathbb{A}\mathbf{x} = \mathbf{b}$ .

Q. 4

Etendre ces résultats au cas d'une décomposition  $\mathbb{A} = \mathbb{D} - \mathbb{E} - \mathbb{F}$  par blocs.

## Correction

R. 1

La matrice  $\mathbb{D}$  est inversible. En effet, pour tout  $i \in \llbracket 1, n \rrbracket$ ,  $d_{i,i} = a_{i,i} = \langle \mathbb{A}\mathbf{e}_i, \mathbf{e}_i \rangle > 0$  car  $\mathbb{A}$  définie positive et  $\mathbf{e}_i$ ,  $i$ -ème vecteur de la base canonique est non nul.

On en déduit que les matrices  $\mathbb{D} - \mathbb{E}$  (triangulaire inférieure de diagonale la diagonale de  $\mathbb{D}$ ) et  $\mathbb{D} - \mathbb{F}$  (triangulaire supérieure de diagonale la diagonale de  $\mathbb{D}$ ) sont inversibles.

De (3.1), on obtient en multipliant à gauche par  $(\mathbb{D} - \mathbb{E})^{-1}$

$$\mathbf{x}_{k+1/2} = (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}_k + (\mathbb{D} - \mathbb{E})^{-1}\mathbf{b}.$$

En remplaçant cette expression de  $\mathbf{x}_{k+1/2}$  dans (3.2), on a

$$\begin{aligned} (\mathbb{D} - \mathbb{F})\mathbf{x}_{k+1} &= \mathbb{E} \left( (\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}_k + (\mathbb{D} - \mathbb{E})^{-1}\mathbf{b} \right) + \mathbf{b} \\ &= \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}_k + \left( \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1} + \mathbb{I} \right) \mathbf{b} \end{aligned}$$

En multipliant à gauche cette équation par  $(\mathbb{D} - \mathbb{F})^{-1}$ , on abouti a

$$\mathbf{x}_{k+1} = (\mathbb{D} - \mathbb{F})^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{x}_k + (\mathbb{D} - \mathbb{F})^{-1} \left( \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1} + \mathbb{I} \right) \mathbf{b}$$

En posant

$$\begin{aligned} \mathbb{B} &= (\mathbb{D} - \mathbb{F})^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F} \\ \mathbf{c} &= (\mathbb{D} - \mathbb{F})^{-1} \left( \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1} + \mathbb{I} \right) \mathbf{b} \end{aligned}$$

on obtient (3.3).

R. 2

a. L'expression à démontrer est bien définie car  $\mathbb{D}$  et  $\mathbb{D} - \mathbb{E}$  inversibles. De plus on a

$$\mathbb{I} = (\mathbb{D} - \mathbb{E})(\mathbb{D} - \mathbb{E})^{-1}$$

En multipliant à gauche cette équation par  $\mathbb{D}^{-1}$  on obtient

$$\begin{aligned} \mathbb{D}^{-1} &= \mathbb{D}^{-1}(\mathbb{D} - \mathbb{E})(\mathbb{D} - \mathbb{E})^{-1} \\ &= (\mathbb{D} - \mathbb{E})^{-1} - \mathbb{D}^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}. \end{aligned}$$

b. Soit  $(\lambda, \mathbf{p})$  un élément propre de la matrice  $\mathbb{B}$ . on a alors

$$\begin{aligned} \mathbb{B}\mathbf{p} = \lambda\mathbf{p} &\Leftrightarrow (\mathbb{D} - \mathbb{F})^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{p} = \lambda\mathbf{p} \\ &\Leftrightarrow \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{p} = \lambda(\mathbb{D} - \mathbb{F})\mathbf{p} \\ &\Leftrightarrow \mathbb{D}^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}\mathbf{p} = \lambda\mathbb{D}^{-1}(\mathbb{D} - \mathbb{F})\mathbf{p} \end{aligned}$$

De (3.4), on a

$$\mathbb{D}^{-1}\mathbb{E}(\mathbb{D} - \mathbb{E})^{-1} = (\mathbb{D} - \mathbb{E})^{-1} - \mathbb{D}^{-1}$$

et donc

$$\begin{aligned}
 \mathbb{B}\mathbf{p} = \lambda\mathbf{p} &\Leftrightarrow \left( (\mathbb{D} - \mathbb{E})^{-1} - \mathbb{D}^{-1} \right) \mathbb{F}\mathbf{p} = \lambda\mathbb{D}^{-1}(\mathbb{D} - \mathbb{F})\mathbf{p} \\
 &\Leftrightarrow (\mathbb{D} - \mathbb{E}) \left( (\mathbb{D} - \mathbb{E})^{-1} - \mathbb{D}^{-1} \right) \mathbb{F}\mathbf{p} = \lambda(\mathbb{D} - \mathbb{E})\mathbb{D}^{-1}(\mathbb{D} - \mathbb{F})\mathbf{p} \\
 &\Leftrightarrow (\mathbb{I} - \mathbb{I} + \mathbb{E}\mathbb{D}^{-1}) \mathbb{F}\mathbf{p} = \lambda(\mathbb{I} - \mathbb{E}\mathbb{D}^{-1})(\mathbb{D} - \mathbb{F})\mathbf{p} \\
 &\Leftrightarrow \mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} = \lambda(\mathbb{D} - \mathbb{E} - \mathbb{F} + \mathbb{E}\mathbb{D}^{-1}\mathbb{F})\mathbf{p} \\
 &\Leftrightarrow \mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} = \lambda(\mathbb{A} + \mathbb{E}\mathbb{D}^{-1}\mathbb{F})\mathbf{p}
 \end{aligned}$$

On en déduit alors

$$\lambda\mathbb{A}\mathbf{p} + (\lambda - 1)\mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} = 0.$$

**R. 3**

La matrice  $\mathbb{A}$  est inversible car elle est définie positive et donc  $\underline{\mathbf{x}}$  est bien définie. De l'équation (3.1), on déduit

$$(\mathbb{D} - \mathbb{E})\mathbf{x}_{k+1/2} = \mathbb{F}\mathbf{x}_k + \mathbb{A}\underline{\mathbf{x}} = \mathbb{F}\mathbf{x}_k + (\mathbb{D} - \mathbb{E} - \mathbb{F})\underline{\mathbf{x}}$$

et donc

$$(\mathbb{D} - \mathbb{E})(\mathbf{x}_{k+1/2} - \underline{\mathbf{x}}) = \mathbb{F}(\mathbf{x}_k - \underline{\mathbf{x}}) \quad (\text{R5.7})$$

De la même manière à partir de l'équation (3.2), on déduit

$$(\mathbb{D} - \mathbb{F})(\mathbf{x}_{k+1} - \underline{\mathbf{x}}) = \mathbb{E}(\mathbf{x}_{k+1/2} - \underline{\mathbf{x}}) \quad (\text{R5.8})$$

En utilisant (R5.7), l'équation (R5.9) devient

$$(\mathbb{D} - \mathbb{F})(\mathbf{x}_{k+1} - \underline{\mathbf{x}}) = \mathbb{E}(\mathbb{D} - \mathbb{E})^{-1}\mathbb{F}(\mathbf{x}_k - \underline{\mathbf{x}})$$

c'est à dire

$$\mathbf{x}_{k+1} - \underline{\mathbf{x}} = \mathbb{B}(\mathbf{x}_k - \underline{\mathbf{x}}).$$

En posant  $\mathbf{e}_k = \mathbf{x}_k - \underline{\mathbf{x}}$  on a alors

$$\mathbf{e}_k = \mathbb{B}^k \mathbf{e}_0, \quad \forall k \geq 0.$$

Or la suite  $\mathbf{x}_k$  converge vers  $\underline{\mathbf{x}}$  si et seulement si la suite  $\mathbf{e}_k$  converge vers  $\mathbf{0}$ . Pour cela, d'après le Théorème 3.2.5, page 106, il est nécessaire et suffisant d'avoir  $\rho(\mathbb{B}) < 1$ .

Soit  $(\lambda, \mathbf{p})$  un élément propre de  $\mathbb{B}$ . Montrons que  $|\lambda| < 1$ .

On déduit de l'équation (3.5)

$$\begin{aligned}
 0 &= \langle \mathbf{p}, \lambda\mathbb{A}\mathbf{p} + (\lambda - 1)\mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle \\
 &= \lambda \langle \mathbf{p}, \mathbb{A}\mathbf{p} \rangle + (\lambda - 1) \langle \mathbf{p}, \mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle
 \end{aligned} \quad (\text{R5.9})$$

Comme la matrice  $\mathbb{A}$  est définie positive on a  $\langle \mathbb{A}\mathbf{p}, \mathbf{p} \rangle > 0$  car  $\mathbf{p} \neq \mathbf{0}$  (vecteur propre) et donc

$$\langle \mathbf{p}, \mathbb{A}\mathbf{p} \rangle = \overline{\langle \mathbb{A}\mathbf{p}, \mathbf{p} \rangle} = \langle \mathbb{A}\mathbf{p}, \mathbf{p} \rangle > 0.$$

De plus on a

$$\langle \mathbf{p}, \mathbb{E}\mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle = \langle \mathbb{E}^*\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle.$$

La matrice  $\mathbb{A}$  étant hermitienne, on a  $\mathbb{E}^* = \mathbb{F}$ . La matrice  $\mathbb{A}$  étant définie positive, la matrice diagonale  $\mathbb{D}$  est définie positive car  $d_{i,i} > 0, \forall i \in \llbracket 1, n \rrbracket$ , et donc  $\mathbb{D}^{-1}$  aussi. Comme  $\mathbb{F}\mathbf{p}$  n'est pas nécessairement non nul, on a

$$\langle \mathbb{D}^{-1}\mathbb{F}\mathbf{p}, \mathbb{F}\mathbf{p} \rangle \in \mathbb{R}^+.$$

On en déduit

$$\langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle = \overline{\langle \mathbb{D}^{-1}\mathbb{F}\mathbf{p}, \mathbb{F}\mathbf{p} \rangle} = \langle \mathbb{D}^{-1}\mathbb{F}\mathbf{p}, \mathbb{F}\mathbf{p} \rangle \geq 0.$$

De l'équation (R5.9), on obtient

$$\lambda(\langle \mathbf{p}, \mathbb{A}\mathbf{p} \rangle + \langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle) = \langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle$$

or

$$\langle \mathbf{p}, \mathbb{A}\mathbf{p} \rangle + \langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle \neq 0$$

ce qui donne

$$\lambda = \frac{\langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle}{\langle \mathbf{p}, \mathbb{A}\mathbf{p} \rangle + \langle \mathbb{F}\mathbf{p}, \mathbb{D}^{-1}\mathbb{F}\mathbf{p} \rangle}.$$

On a alors  $\lambda \in [0, 1[$ .

R. 4

Comme la matrice  $A$  est hermitienne définie positive, chaque bloc diagonal l'est aussi. Donc la matrice diagonale bloc  $D$  est aussi hermitienne définie positive ainsi que son inverse. Les résultats précédents sont donc toujours valables.

~~~~~ Fin correction ~~~~~

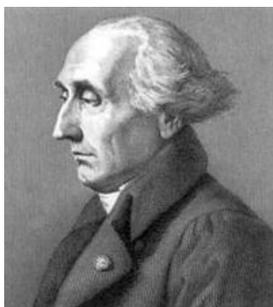
# Chapitre 4

## Interpolation

L'interpolation est un outil mathématique permettant de construire des fonctions à partir de la donnée d'un nombre fini de valeurs.

**A développer...**

Les protagonistes de cette histoire



(a) *Joseph-Louis Lagrange* 1736-1813, mathématicien italien puis français



(b) *Pafnouti Lvovitch Tchebychev* 1821-1894, mathématicien russe puis français



(c) *Charles Hermite* 1822-1901, mathématicien français



(d) *Henri-Léon Lebesgue* 1875-1941, mathématicien français

## 4.1 Polynôme d'interpolation de Lagrange

### EXERCICE 4.1.1

Soient  $n \in \mathbb{N}^*$  et  $(n+1)$  couples de  $\mathbb{R}^2$ ,  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$ , tels que les  $x_i$  sont distincts deux à deux. On note

Q. 1

a. Soit  $i \in \llbracket 0, n \rrbracket$ . Montrer qu'il existe un unique polynôme  $L_i$  de degré  $n$  vérifiant

$$L_i(x_j) = \delta_{ij}, \quad \forall j \in \llbracket 0, n \rrbracket. \quad (4.1)$$

b. Montrer que les  $(L_i)_{i \in \llbracket 0, n \rrbracket}$  forment une base de  $\mathbb{R}_n[X]$  (espace vectoriel des polynômes à coefficients réels de degré inférieur ou égal à  $n$ ).

On définit le polynôme  $P_n$  par

$$P_n(x) = \sum_{i=0}^n y_i L_i(x). \quad (4.2)$$

Q. 2

Montrer que polynôme  $P_n$  est l'unique polynôme de degré au plus  $n$  vérifiant  $P_n(x_i) = y_i$ ,  $\forall i \in \llbracket 0, n \rrbracket$ .

### Correction

R. 1

a. De (4.1), on déduit que les  $n$  points distincts  $x_j$  pour  $j \in \llbracket 0, n \rrbracket \setminus \{i\}$  sont les  $n$  zéros du polynôme  $L_i$  de degré  $n$  : il s'écrit donc sous la forme

$$L_i(x) = C \prod_{\substack{j=0 \\ j \neq i}}^n (x - x_j) \quad \text{avec } C \in \mathbb{R}$$

Pour déterminer la constante  $C$ , on utilise (4.1) avec  $j = i$

$$L_i(x_i) = 1 = C \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)$$

Les points  $x_i$  sont distincts deux à deux, on a  $\prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j) \neq 0$  et donc

$$C = \frac{1}{\prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)}$$

d'où

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}, \quad \forall i \in \llbracket 0, n \rrbracket. \quad (\text{R1.10})$$

Il reste à démontrer l'unicité. On suppose qu'il existe  $L_i$  et  $U_i$  deux polynômes de  $\mathbb{R}_n[X]$  vérifiant (4.1). Alors  $Q_i = L_i - U_i$  est polynôme de degré  $n$  (au plus) admettant  $n+1$  zéros distincts, c'est donc le polynôme nul et on a nécessairement  $L_i = U_i$ .

b. On sait que  $\dim \mathbb{R}_n[X] = n+1$ . Pour que les  $\{L_i\}_{i \in \llbracket 0, n \rrbracket}$  forment une base de  $\mathbb{R}_n[X]$  il suffit de démontrer qu'ils sont linéairement indépendants.

Soit  $\lambda_0, \dots, \lambda_n$   $n+1$  scalaires. Montrons pour cela que

$$\sum_{i=0}^n \lambda_i L_i = 0 \implies \lambda_i = 0, \quad \forall i \in \llbracket 0, n \rrbracket$$

Noter que la première égalité est dans l'espace vectoriel  $\mathbb{R}_n[X]$  et donc le 0 est pris au sens polynôme nul.

On a

$$\sum_{i=1}^n \lambda_i L_i = 0 \iff \sum_{i=1}^n \lambda_i L_i(x) = 0, \forall x \in \mathbb{R}$$

Soit  $k \in \llbracket 0, n \rrbracket$ . En choisissant  $x = x_k$ , on a par (4.1)  $\sum_{i=1}^n \lambda_i L_i(x_k) = \lambda_k$  et donc

$$\sum_{i=1}^n \lambda_i L_i = 0 \implies \sum_{i=1}^n \lambda_i L_i(x_k) = 0, \forall k \in \llbracket 0, n \rrbracket \iff \lambda_k = 0, \forall k \in \llbracket 0, n \rrbracket.$$

Les  $\{L_i\}_{i \in \llbracket 0, n \rrbracket}$  sont donc linéairement indépendants.

**R. 2**

Par construction  $P_n \in \mathbb{R}_n[X]$  et on a,  $\forall j \in \llbracket 0, n \rrbracket^a$ ,

$$\begin{aligned} P_n(x_j) &\stackrel{\text{def}}{=} \sum_{i=0}^n y_i L_i(x_j) \\ &= \sum_{i=0}^n y_i \delta_{i,j} \text{ par (4.1)} \\ &= y_j. \end{aligned}$$

Pour démontrer l'unicité, on propose ici deux méthodes

- On note  $P_a$  et  $P_b$  deux polynômes de  $\mathbb{R}_n[X]$  vérifiant (4.1). Le polynôme  $Q = P_a - P_b$  appartient aussi à  $\mathbb{R}_n[X]$  et il vérifie,  $\forall i \in \llbracket 0, n \rrbracket$ ,

$$Q(x_i) = P_a(x_i) - P_b(x_i) = 0.$$

Les  $n+1$  points  $x_i$  étant distincts, ce sont donc  $n+1$  racines distinctes du polynôme  $Q$ . Or tout polynôme de degré  $n$  admet au plus  $n$  racines distinctes<sup>b</sup>. On en déduit que le seul polynôme de degré au plus  $n$  admettant  $n+1$  racines distinctes est le polynôme nulle et donc  $P_a = P_b$ .

- c'est l'unique polynôme de degré au plus  $n$  vérifiant (4.2) car la décomposition dans la base  $\{L_i\}_{i \in \llbracket 0, n \rrbracket}$  est unique.

<sup>a</sup>A noter le choix de l'indice  $j$ . Que doit-on faire dans ce qui suit si l'on choisit  $i$  comme indice?

<sup>b</sup>Le théorème de d'Alembert-Gauss affirme que tout polynôme à coefficients complexes de degré  $n$  admet  $n$  racines complexes qui ne sont pas nécessairement distinctes

~~~~~ Fin correction ~~~~~

**Définition 4.1.1.** Soient  $n \in \mathbb{N}^*$  et  $(n+1)$  couples de  $\mathbb{R}^2$ ,  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$ , tels que les  $x_i$  sont distincts deux à deux. Le **polynôme d'interpolation de Lagrange** de  $\mathbb{R}_n[X]$ , noté  $P_n$ , et vérifiant

$$\forall i \in \llbracket 0, n \rrbracket, P_n(x_i) = y_i \quad (4.1)$$

est donné par

$$\forall x \in \mathbb{R}, P_n(x) = \sum_{i=0}^n y_i L_i(x), \quad (4.2)$$

où les  $L_i \in \mathbb{R}_n[X]$  sont les **polynômes de base de Lagrange** donnés par

$$\forall i \in \llbracket 0, n \rrbracket, \forall x \in \mathbb{R}, L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}. \quad (4.3)$$

**Théorème 4.1.1.** Le *polynôme d'interpolation de Lagrange*,  $\mathcal{P}_n$ , associé aux  $(n+1)$  couples  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$ , est l'unique polynôme de degré au plus  $n$ , vérifiant

$$\mathcal{P}_n(x_i) = y_i, \quad \forall i \in \llbracket 0, n \rrbracket. \quad (4.4)$$

**Remarque 4.1.1.** Il est aussi possible d'obtenir l'existence et l'unicité du polynôme d'interpolation de Lagrange sans passer par sa construction. Pour cela on définit  $\Phi : \mathbb{R}_n[X] \rightarrow \mathbb{R}^{n+1}$  par

$$\forall P \in \mathbb{R}_n[X], \quad \Phi(P) = (P(x_0), \dots, P(x_n)).$$

L'existence et l'unicité du polynôme  $P_n$  est équivalente à la bijectivité de l'application  $\Phi$ . Or celle-ci est une application linéaire entre deux espaces de dimension  $n+1$ . Elle est donc bijective si et seulement si elle est injective (ou surjective). Pour vérifier l'injectivité de  $\Phi$  il est nécessaire et suffisant de vérifier que son noyau est réduit au polynôme nul.

Soit  $P \in \ker \Phi$ . On a alors  $\Phi(P) = \mathbf{0}_{n+1}$  et donc  $(x_0, \dots, x_n)$  sont  $n+1$  racines distinctes de  $P$ . Or le seul polynôme de  $\mathbb{R}_n[X]$  ayant  $n+1$  racines distinctes est le polynôme nul et donc  $P = 0$ .

A titre d'exemple, on représente, En figure 4.2, le polynôme d'interpolation de Lagrange associé à 7 points donnés.

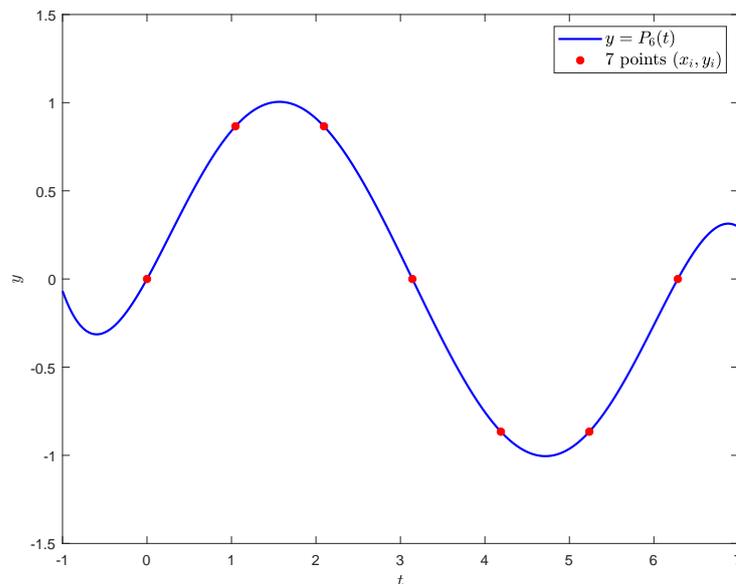


Figure 4.2: Polynôme d'interpolation de Lagrange avec 7 points donnés)

#### EXERCICE 4.1.2

Ecrire la fonction `Lagrange` permettant de calculer  $\mathcal{P}_n$  (polynôme d'interpolation de Lagrange associé aux  $(n+1)$  couples  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$ ) en  $t \in \mathbb{R}$ .

**But :** Calculer le polynôme  $\mathcal{P}_n(t)$  défini par (4.2)  
**Données :**  $\mathbf{X}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $X(i) = x_{i-1} \quad \forall i \in \llbracket 1, n+1 \rrbracket$  et  $X(i) \neq X(j)$  pour  $i \neq j$ ,  
**Correction**  $\mathbf{Y}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $Y(i) = y_{i-1} \quad \forall i \in \llbracket 1, n+1 \rrbracket$ ,  
 $t$  : un réel.  
**Résultat :**  $y$  : le réel  $y = \mathcal{P}_n(t)$ .

**Algorithme 4.1**  $\mathcal{R}_0$

1: Calcul de  $y = \mathcal{P}_n(t) = \sum_{i=1}^{n+1} Y(i)L_{i-1}(t)$

**Algorithme 4.1**  $\mathcal{R}_1$

- 1:  $y \leftarrow 0$
- 2: **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 3:      $y \leftarrow y + Y(i) * L_{i-1}(t)$
- 4: **Fin**

**Algorithme 4.1**  $\mathcal{R}_1$

- 1:  $y \leftarrow 0$
- 2: **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 3:      $y \leftarrow y + Y(i) * L_{i-1}(t)$
- 4: **Fin**

**Algorithme 4.1**  $\mathcal{R}_2$

- 1:  $y \leftarrow 0$
- 2: **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 3:      $L \leftarrow \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{t - X(j)}{X(i) - X(j)}$
- 4:      $y \leftarrow y + Y(i) * L$
- 5: **Fin**

**Algorithme 4.1**  $\mathcal{R}_2$

- 1:  $y \leftarrow 0$
- 2: **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 3:      $L \leftarrow \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{t - X(j)}{X(i) - X(j)}$
- 4:      $y \leftarrow y + Y(i) * L$
- 5: **Fin**

**Algorithme 4.1**  $\mathcal{R}_3$

- 1:  $y \leftarrow 0$
- 2: **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 3:      $L \leftarrow 1$
- 4:     **Pour**  $j \leftarrow 1$  à  $n + 1$ , ( $j \sim = i$ ) **faire**
- 5:          $L \leftarrow L * (t - X(j)) / (X(i) - X(j))$
- 6:     **Fin**
- 7:      $y \leftarrow y + Y(i) * L$
- 8: **Fin**

On obtient alors l'algorithme final

**Algorithme 4.1** Fonction **Lagrange** permettant de calculer le polynôme d'interpolation de Lagrange  $\mathcal{P}_n(x)$  défini par (4.2)

**Données :**  $\mathbf{X}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $\mathbf{X}(i) = x_{i-1} \forall i \in \llbracket 1, n + 1 \rrbracket$  et  $\mathbf{X}(i) \neq \mathbf{X}(j)$  pour  $i \neq j$ ,  
 $\mathbf{Y}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $\mathbf{Y}(i) = y_{i-1} \forall i \in \llbracket 1, n + 1 \rrbracket$ ,  
 $t$  : un réel.

**Résultat :**  $y$  : le réel  $y = \mathcal{P}_n(t)$ .

- 1: **Fonction**  $y \leftarrow \text{Lagrange}(t, \mathbf{X}, \mathbf{Y})$
- 2:      $y \leftarrow 0$
- 3:     **Pour**  $i \leftarrow 1$  à  $n + 1$  **faire**
- 4:          $L \leftarrow 1$
- 5:         **Pour**  $j \leftarrow 1$  à  $n + 1$ , ( $j \sim = i$ ) **faire**
- 6:              $L \leftarrow L * (t - \mathbf{X}(j)) / (\mathbf{X}(i) - \mathbf{X}(j))$
- 7:         **Fin**
- 8:          $y \leftarrow y + \mathbf{Y}(i) * L$
- 9:     **Fin**
- 10:     **return**  $y$
- 11: **Fin**

~~~~~ Fin correction ~~~~~

## 4.1.1 Erreur de l'interpolation

Soit une fonction  $f : [a, b] \rightarrow \mathbb{R}$ . On suppose que les  $y_i$  sont donnés par

$$y_i = f(x_i), \quad \forall i \in \llbracket 0, n \rrbracket. \quad (4.5)$$

On cherche à évaluer l'erreur  $E_n(t) = f(t) - \mathcal{P}_n(t)$ ,  $\forall t \in [a, b]$ .

On propose en figures 4.3 à 4.5 d'étudier deux exemples. Le premier (figure de gauche) correspond à l'interpolation de la fonction  $\sin(t)$  par le polynôme d'interpolation de Lagrange aux points équidistants  $t_i = a + ih$  avec  $a = 0$  et  $h = 2\pi/n$ . Le second (figure de droite) correspond à l'interpolation de la fonction  $\frac{1}{1+t^2}$  par le polynôme d'interpolation de Lagrange aux points  $x_i = a + ih$  avec  $a = -5$  et  $h = 10/n$ .

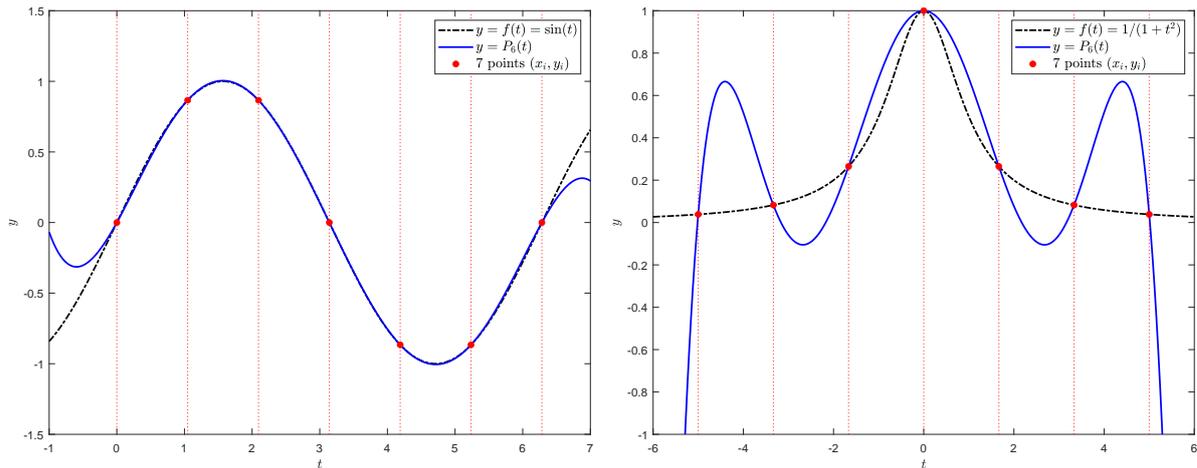


Figure 4.3: Polynômes d'interpolation de Lagrange avec  $n = 6$  (7 points) uniformément répartis. À gauche pour la fonction  $f : t \rightarrow \sin(t)$  avec  $x_0 = 0$ ,  $x_6 = 2\pi$  et à droite pour la fonction  $f : t \rightarrow 1/(1+t^2)$  avec  $x_0 = -5$ ,  $x_6 = 5$ .

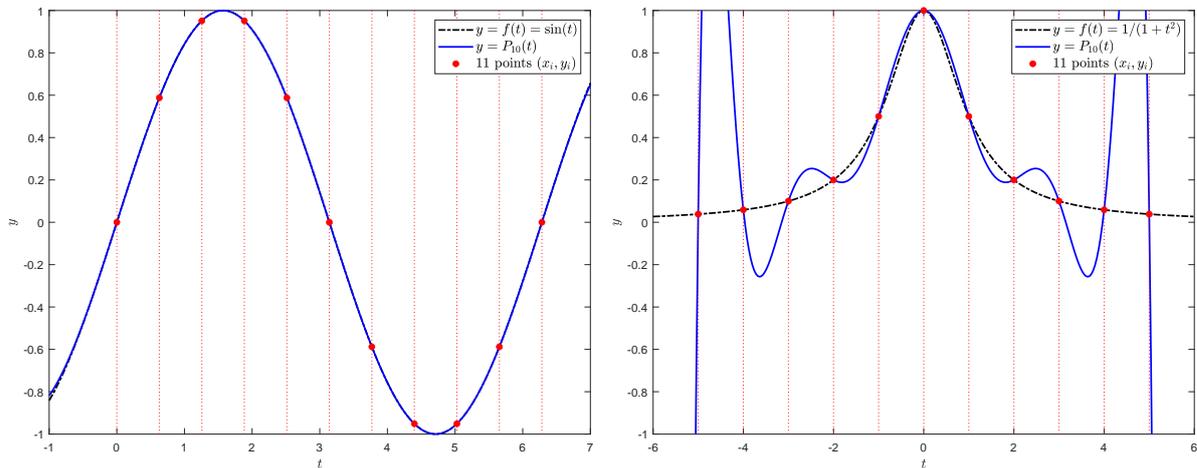


Figure 4.4: Polynômes d'interpolation de Lagrange avec  $n = 10$  (11 points) uniformément répartis. À gauche pour la fonction  $f : t \rightarrow \sin(t)$  avec  $x_0 = 0$ ,  $x_{10} = 2\pi$  et à droite pour la fonction  $f : tx \rightarrow 1/(1+t^2)$  avec  $x_0 = -5$ ,  $x_{10} = 5$ .

**EXERCICE 4.1.3**

Soit  $f \in \mathcal{C}^{n+1}([a, b]; \mathbb{R})$ . Soient  $n \in \mathbb{N}^*$  et  $n + 1$  couples de  $\mathbb{R}^2$ ,  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$ , tels que les  $x_i$  sont distincts deux à deux et  $y_i = f(x_i)$ .

On note par  $\mathcal{P}_n$  le polynôme d'interpolation de Lagrange associé aux points  $(x_i, y_i)_{i \in \llbracket 0, n \rrbracket}$  et  $\pi_n$  le

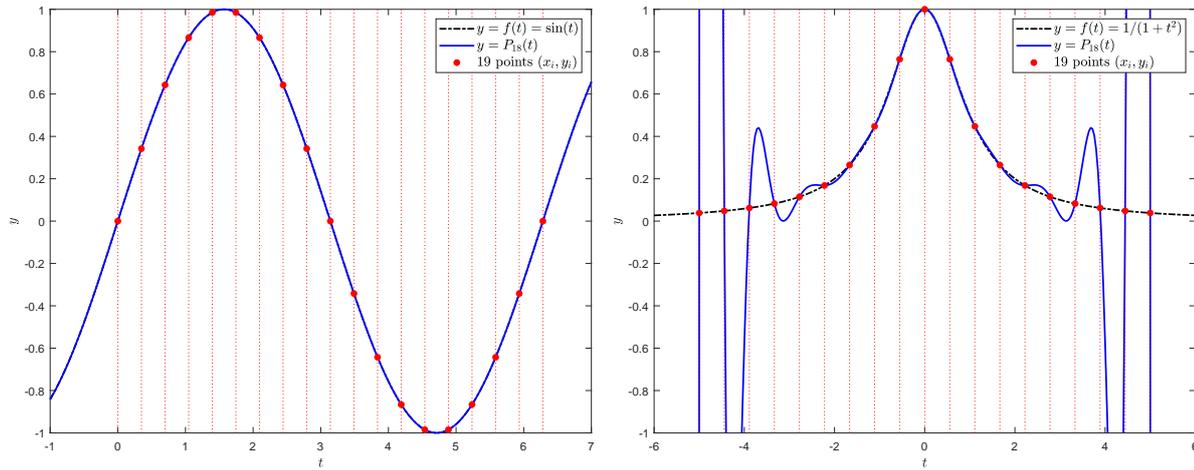


Figure 4.5: Polynômes d'interpolation de Lagrange avec  $n = 18$  (19 points) uniformément répartis. A gauche pour la fonction  $f : t \rightarrow \sin(t)$  avec  $x_0 = 0, x_{18} = 2\pi$  et à droite pour la fonction  $f : t \rightarrow 1/(1 + t^2)$  avec  $x_0 = -5, x_{18} = 5$ .

polynôme de degré  $n + 1$  défini par

$$\pi_n(x) = \prod_{i=0}^n (x - x_i). \tag{4.1}$$

**Q. 1**

Montrer que,  $\forall x \in [a; b]$ , il existe  $\xi_x$  appartenant au plus petit intervalle fermé contenant  $x, x_0, \dots, x_n$  tel que

$$f(x) - P_n(x) = \frac{\pi_n(x)}{(n + 1)!} f^{(n+1)}(\xi_x). \tag{4.2}$$

**Indication :** Etudier les zéros de la fonction  $F(t) = f(t) - P_n(t) - \frac{f(x) - P_n(x)}{\pi_n(x)} \pi_n(t)$ .

**Correction**

**R. 1**

S'il existe  $i \in \llbracket 0, n \rrbracket$  tel que  $x = x_i$  alors l'équation (4.2) est immédiatement vérifiée. Soit  $x \in [a, b]$  distinct de tous les  $x_i$ . Comme  $f \in \mathcal{C}^{n+1}([a; b]; \mathbb{R})$ ,  $P_n \in \mathbb{R}_n[X]$  et  $\pi_n \in \mathbb{R}_{n+1}[X]$ , on en déduit que la fonction  $F$  est dans  $\mathcal{C}^{n+1}([a; b]; \mathbb{R})$ . La fonction  $F$  admet aussi  $n + 2$  zéros :  $x, x_0, \dots, x_n$ . On note  $\xi_{x,1}^{[0]}, \dots, \xi_{x,n+2}^{[0]}$  ces  $n + 2$  zéros ordonnés  $\xi_{x,1}^{[0]} < \dots < \xi_{x,n+2}^{[0]}$ . La fonction  $F$  étant continue sur  $[a, b]$  et dérivable sur  $]a, b[$ , le théorème de Rolle dit qu'entre deux zéros consécutifs de  $F$ , il existe au moins un zéro de  $F' = F^{(1)}$ . Plus précisément on a

$$\forall i \in \llbracket 1, n + 1 \rrbracket, \exists \xi_{x,i}^{[1]} \in ]\xi_{x,i}^{[0]}, \xi_{x,i+1}^{[0]}[ \text{ tels que } F^{(1)}(\xi_{x,i}^{[1]}) = 0$$

et on en déduit que la fonction  $F^{(1)}$  admet  $n + 1$  zéros  $\xi_{x,1}^{[1]}, \dots, \xi_{x,n+1}^{[1]}$  et l'on a  $\xi_{x,1}^{[0]} < \xi_{x,1}^{[1]} < \dots < \xi_{x,n+1}^{[1]} < \xi_{x,n+2}^{[0]}$ . Il faut noter la dépendance en  $x$  des zéros de  $F'$  d'où la notation un peu "lourde". Montrons par récurrence finie que  $(\mathcal{P}_k)$  est vraie pour tout  $k \in \llbracket 1, n + 1 \rrbracket$

$$(\mathcal{P}_k) : \exists \xi_{x,i}^{[k]}, i \in \llbracket 1, n + 2 - k \rrbracket, \xi_{x,1}^{[0]} < \xi_{x,1}^{[k]} < \dots < \xi_{x,n+2-k}^{[k]} < \xi_{x,n+2}^{[0]} \text{ tels que } F^{(k)}(\xi_{x,i}^{[k]}) = 0$$

**Initialisation :** Pour  $k = 1$ , la preuve a déjà été faite.

**Hérédité :** Soit  $1 < k - 1 < n + 1$ , on suppose  $(\mathcal{P}_{k-1})$  vérifiée. La fonction  $F^{(k-1)}$  étant continue sur  $[a, b]$  et dérivable sur  $]a, b[$ , le théorème de Rolle dit qu'entre deux zéros consécutifs de  $F^{(k-1)}$ , il existe au moins un zéro de  $F^{(k)}$ . Par hypothèse  $F^{(k-1)}$  admet  $n + 2 - (k - 1)$  zéros vérifiant

$$\xi_{x,1}^{[0]} < \xi_{x,1}^{[k-1]} < \dots < \xi_{x,n+2-(k-1)}^{[k-1]} < \xi_{x,n+2}^{[0]}$$

La fonction  $F^{(k-1)}$  est continue sur  $[a, b]$  et dérivable sur  $]a, b[$  puisque  $F \in \mathcal{C}^{n+1}([a; b]; \mathbb{R})$ . Par application du théorème de Rolle, entre deux zéros de  $F^{(k-1)}$ , il existe au moins un zéro de  $F^{(k)}$ . Plus précisément pour tout  $i \in \llbracket 1, n+2-k \rrbracket$  on a

$$\exists \xi_{x,i}^{[k]} \in ]\xi_{x,i}^{[k-1]}, \xi_{x,i+1}^{[k-1}[ , \quad F^{(k)}(\xi_{x,i}^{[k]}) = 0$$

De plus, par construction,  $\xi_{x,1}^{[0]} < \xi_{x,1}^{[k]} < \dots < \xi_{x,n+2-k}^{[k]} < \xi_{x,n+2}^{[0]}$  et donc  $(\mathcal{P}_k)$  est vraie.

Avec  $k = n+1$  on obtient

$$(\mathcal{P}_{n+1}) : \exists \xi_{x,1}^{[n+1]} \in ]\xi_{x,1}^{[0]}, \xi_{x,n+2}^{[0}[ \text{ tel que } F^{(n+1)}(\xi_{x,1}^{[n+1]}) = 0$$

et donc

$$0 = F^{(n+1)}(\xi_{x,1}^{[n+1]}) = f^{(n+1)}(\xi_{x,1}^{[n+1]}) - P_n^{(n+1)}(\xi_{x,1}^{[n+1]}) - \frac{f(x) - P_n(x)}{\pi_n(x)} \pi_n^{(n+1)}(\xi_{x,1}^{[n+1]})$$

Comme  $P_n \in \mathbb{R}_n[X]$ , on a  $P_n^{(n+1)} = 0$ . De plus  $\pi_n \in \mathbb{R}_{n+1}[X]$ , et comme  $\pi_n(x) = x^{n+1} + Q(x)$  avec  $Q \in \mathbb{R}_n[X]$  (i.e. son monôme de puissance  $n+1$  à pour coefficient 1) on obtient  $\pi_n^{(n+1)}(x) = (n+1)!$  On a alors

$$f^{(n+1)}(\xi_{x,1}^{[n+1]}) = \frac{f(x) - P_n(x)}{\pi_n(x)} (n+1)!$$

~~~~~ Fin correction ~~~~~

Le résultat suivant est du à Cauchy (1840)

**Théorème 4.1.2.** Soient  $n \in \mathbb{N}^*$  et  $(x_i)_{i=0}^n$ ,  $(n+1)$  points distincts de l'intervalle  $[a, b]$ . Soient  $f \in \mathcal{C}^{n+1}([a; b]; \mathbb{R})$  et  $\mathcal{P}_n$  le polynôme d'interpolation de Lagrange de degré  $n$  passant par  $(x_i, f(x_i))$ ,  $\forall i \in \llbracket 0, n \rrbracket$ . Alors,  
 $\forall x \in [a, b]$ ,  $\exists \xi_x \in (\min(x_i, x), \max(x_i, x))$ ,

$$f(x) - \mathcal{P}_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i) \quad (4.6)$$

#### 4.1.2 Points de Chebyshev

Pour minimiser l'erreur commise lors de l'interpolation d'une fonction  $f$  par un polynôme d'interpolation de Lagrange, on peut, pour un  $n$  donné, "jouer" sur le choix des points  $x_i$  :

Trouver  $(\bar{x}_i)_{i=0}^n$ ,  $\bar{x}_i \in [a, b]$ , distincts deux à deux, tels que

$$\max_{t \in [a, b]} \prod_{i=0}^n |t - \bar{x}_i| \leq \max_{t \in [a, b]} \prod_{i=0}^n |t - x_i|, \quad \forall (x_i)_{i=0}^n, \quad x_i \in [a, b], \text{ distincts 2 à 2} \quad (4.7)$$

On a alors le résultat suivant

**Théorème 4.1.3.** Les points réalisant (4.7) sont les points de Chebyshev donnés par

$$\bar{x}_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2i+1)\pi}{2n+2}\right), \quad \forall i \in \llbracket 0, n \rrbracket. \quad (4.8)$$

#### 4.1.3 Stabilité

Inspiré du polycopié de G. Barles : ici

On suppose que l'on commet des erreurs lors du calcul des  $f(x_i)$  et l'on note  $f_i \approx f(x_i)$  les valeurs numériques obtenues. Dans ce cadre, on a deux polynômes d'interpolation de Lagrange l'un "exact" avec

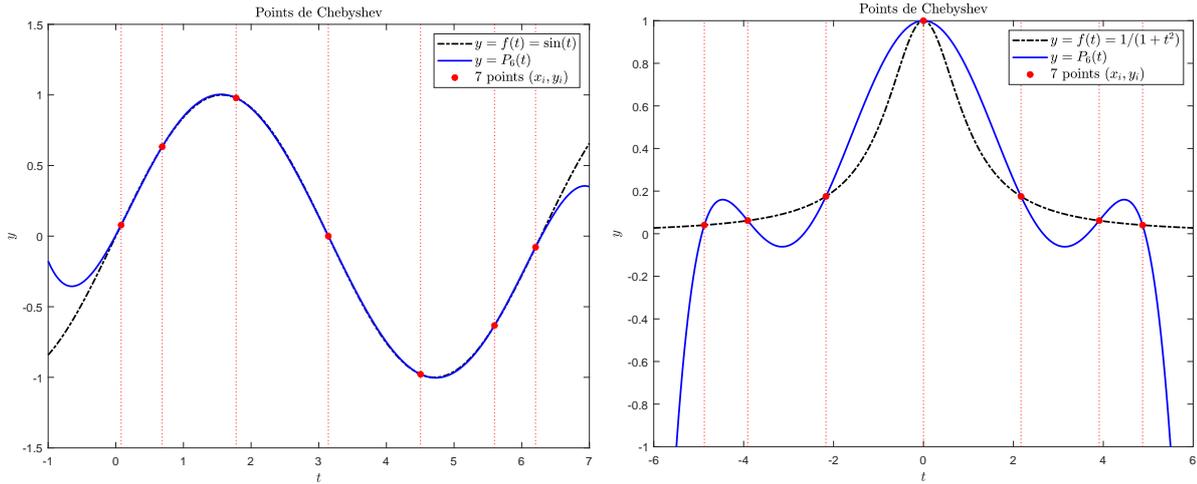


Figure 4.6: Erreurs d'interpolation avec  $n = 6$

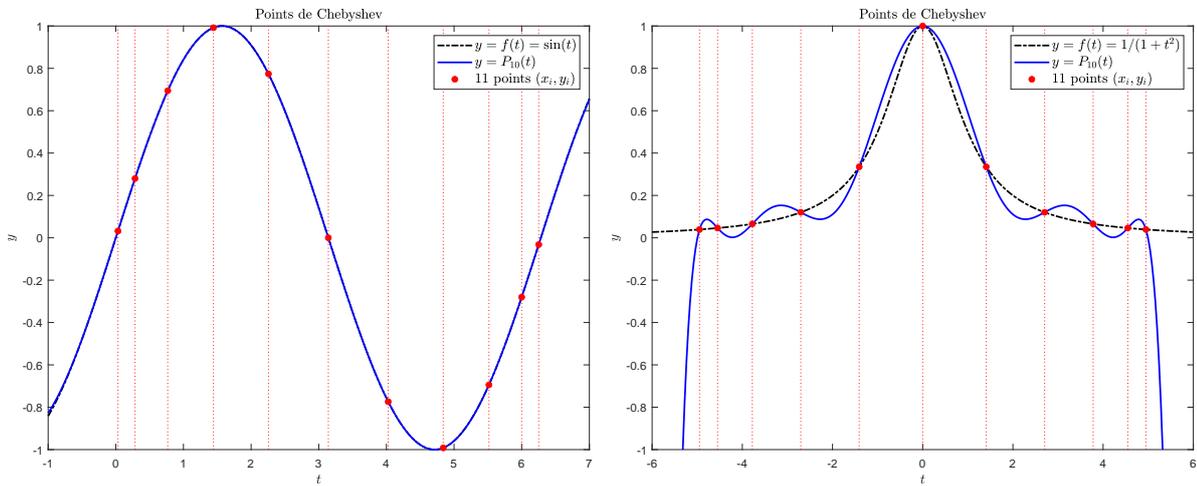


Figure 4.7: Erreurs d'interpolation avec  $n = 10$

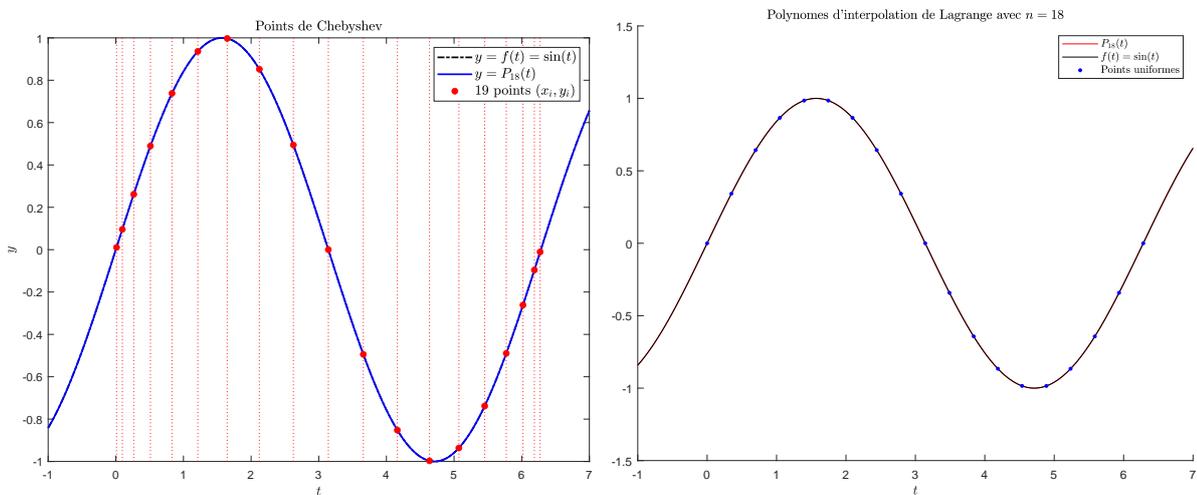


Figure 4.8: Erreurs d'interpolation avec  $n = 18$

les couples de points  $(x_i, y_i) = (x_i, f(x_i))$  noté  $P_n$ , et l'autre "approché" avec les couples de points  $(x_i, f_i)$  noté  $\hat{P}_n$ . On a donc

$$P_n(x) = \sum_{i=0}^n f(x_i)L_i(x) \quad \text{et} \quad \hat{P}_n(x) = \sum_{i=0}^n f_i L_i(x)$$

L'erreur commise est alors majorée par :

$$\begin{aligned} |\hat{P}_n(x) - P_n(x)| &= \left| \sum_{i=0}^n (f_i - f(x_i))L_i(x) \right| \\ &\leq \sum_{i=0}^n |f_i - f(x_i)| |L_i(x)| \\ &\leq \max_{i \in \llbracket 0, n \rrbracket} |f_i - f(x_i)| \sum_{i=0}^n |L_i(x)|. \end{aligned}$$

On note  $\Lambda_n = \max_{x \in [a, b]} \sum_{i=0}^n |L_i(x)|$ , dites *Constante de Lebesgue*. Cette constante est bien définie car l'application  $x \mapsto \sum_{i=0}^n |L_i(x)|$  est continue sur  $[a, b]$  intervalle fermé borné de  $\mathbb{R}$  et donc le maximum est bien atteint : il est donc fini.

On obtient alors

$$\|\hat{P}_n - P_n\|_{\infty} \leq \Lambda_n \max_{i \in \llbracket 0, n \rrbracket} |f_i - f(x_i)|. \quad (4.9)$$

**Proposition 4.1.1.** Soient  $n \in \mathbb{N}^*$  et  $x_0, \dots, x_n$  des points distincts de  $[a, b]$ . L'application  $\mathcal{L}_n : \mathcal{C}^0([a, b]; \mathbb{R}) \rightarrow \mathbb{R}_n[X]$  qui à toute fonction  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$  donne le polynôme d'interpolation de Lagrange  $P_n$  associés aux couples de  $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$  est bien définie et linéaire. De plus, en munissant  $\mathcal{C}^0([a, b]; \mathbb{R})$  et  $\mathbb{R}_n[X]$  de la norme  $\|\cdot\|_{\infty}$ , on a

$$\|\mathcal{L}_n(f)\|_{\infty} \leq \Lambda_n \|f\|_{\infty}, \quad (4.10)$$

ce qui assure la continuité de  $\mathcal{L}_n$ , et

$$\|\mathcal{L}_n\| \stackrel{\text{def}}{=} \sup_{\substack{f \in \mathcal{C}^0([a, b]; \mathbb{R}) \\ f \neq 0}} \frac{\|\mathcal{L}_n(f)\|_{\infty}}{\|f\|_{\infty}} = \Lambda_n. \quad (4.11)$$

*Preuve.* • Montrons que  $\mathcal{L}_n$  est bien définie.

Comme  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ , on a,  $\forall x \in [a, b]$ ,  $f(x)$  défini. <sup>1</sup> On a donc,  $\forall i \in \llbracket 0, n \rrbracket$ ,  $f(x_i)$  défini. D'après le Théorème 4.1.1, les points  $(x_i)_{i=0}^n$  étant distincts deux à deux, le polynôme d'interpolation de Lagrange  $\mathcal{L}_n(f)$  associés aux couples de  $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$  est unique dans  $\mathbb{R}_n[X]$ . L'application  $\mathcal{L}_n$  est donc bien définie.

• Montrons que  $\mathcal{L}_n$  est linéaire.

Soient  $(\lambda, \mu) \in \mathbb{R}^2$ , et,  $f$  et  $g$  deux fonctions de  $\mathcal{C}^0([a, b]; \mathbb{R})$ . il nous faut démontrer, par exemple, que

$$\mathcal{L}_n(\lambda f + \mu g) = \lambda \mathcal{L}_n(f) + \mu \mathcal{L}_n(g).$$

Cette dernière équation est équivalente à

$$\forall x \in [a, b], \quad \mathcal{L}_n(\lambda f + \mu g)(x) = \lambda \mathcal{L}_n(f)(x) + \mu \mathcal{L}_n(g)(x).$$

Soit  $x \in [a, b]$ . On a

$$\begin{aligned} \mathcal{L}_n(\lambda f + \mu g)(x) &= \sum_{i=0}^n (\lambda f + \mu g)(x_i) L_i(x) \\ &= \sum_{i=0}^n (\lambda f(x_i) + \mu g(x_i)) L_i(x) \\ &= \lambda \sum_{i=0}^n f(x_i) L_i(x) + \mu \sum_{i=0}^n g(x_i) L_i(x) \\ &= \lambda \mathcal{L}_n(f)(x) + \mu \mathcal{L}_n(g)(x). \end{aligned}$$

<sup>1</sup>Ceci n'aurait pas été le cas si  $f \in L^2([a, b])$  puisque  $f(x)$  aurait alors été défini pour presque tout  $x \in [a, b]$ .

Ce qui démontre la linéarité de  $\mathcal{L}_n$ .

- Montrons que  $\mathcal{L}_n$  est continue. Comme  $\mathcal{L}_n$  est linéaire, il suffit de démontrer que

$$\exists C > 0 \text{ t.q. } \forall f \in \mathcal{C}^0([a, b]; \mathbb{R}), \|\mathcal{L}_n(f)\|_\infty \leq C \|f\|_\infty.$$

En effet, on a pour tout  $x \in [a, b]$  et pour tout  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ ,

$$\begin{aligned} |\mathcal{L}_n(f)(x)| &= \left| \sum_{i=0}^n f(x_i) L_i(x) \right| \\ &\leq \sum_{i=0}^n |f(x_i) L_i(x)| \leq \|f\|_\infty \sum_{i=0}^n |L_i(x)| \\ &\leq \Lambda_n \|f\|_\infty. \end{aligned}$$

On obtient alors

$$\|\mathcal{L}_n(f)\|_\infty \leq \Lambda_n \|f\|_\infty. \quad (4.10)$$

En prenant  $C = \Lambda_n$ , qui est bien indépendant de  $f$ , on obtient la continuité de  $\mathcal{L}_n$ . Nous allons démontrer (4.11) en deux étapes.

- Montrons que  $\|\mathcal{L}_n\| \leq \Lambda_n$ .  
Soit  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ ,  $f \neq 0$ . (rappel:  $f = 0 \Leftrightarrow \|f\|_\infty = 0$ ) De (4.10), on a

$$\frac{\|\mathcal{L}_n(f)\|_\infty}{\|f\|_\infty} \leq \Lambda_n.$$

En prenant le sup sur toutes les fonctions non nulles, on obtient

$$\|\mathcal{L}_n\| \stackrel{\text{def}}{=} \sup_{\substack{f \in \mathcal{C}^0([a, b]; \mathbb{R}) \\ f \neq 0}} \frac{\|\mathcal{L}_n(f)\|_\infty}{\|f\|_\infty} \leq \Lambda_n.$$

- Nous allons construire une fonction vérifiant l'égalité (4.11).  
Pour cela nous allons reprendre les calculs précédents et regarder si l'on peut trouver  $\bar{x} \in [a, b]$  et  $\bar{f} \in \mathcal{C}^0([a, b]; \mathbb{R})$  vérifiant

$$|\mathcal{L}_n(\bar{f})(\bar{x})| = \Lambda_n \|\bar{f}\|_\infty.$$

On détermine  $\bar{x}$  pour que la dernière majoration, correspondant à  $\sum_{i=0}^n |L_i(x)| \leq \Lambda_n$ , devienne une égalité. L'application  $x \mapsto \sum_{i=0}^n |L_i(x)|$  étant continue sur le fermé borné  $[a, b]$ , il existe alors  $\bar{x} \in [a, b]$  tel que

$$\Lambda_n = \max_{x \in [a, b]} \sum_{i=0}^n |L_i(x)| = \sum_{i=0}^n |L_i(\bar{x})|.$$

On va maintenant regarder s'il est possible de construire une fonction  $\bar{f} \in \mathcal{C}^0([a, b]; \mathbb{R})$  telle que

$$\left| \sum_{i=0}^n \bar{f}(x_i) L_i(\bar{x}) \right| = \sum_{i=0}^n |\bar{f}(x_i) L_i(\bar{x})| = \|\bar{f}\|_\infty \sum_{i=0}^n |L_i(\bar{x})|.$$

Si c'est le cas, on aura bien le résultat souhaité.

Sans déroger à la généralité, on peut supposer les points  $x_i$  ordonnés :  $x_0 < x_1 < \dots < x_n$ . Pour avoir la première égalité, il faut que tous les  $\bar{f}(x_i) L_i(\bar{x})$  soient de même signe. On choisit le signe positif, et on impose par exemple les valeurs de  $\bar{f}(x_i)$ ,  $\forall i \in \llbracket 0, n \rrbracket$ ,

$$\begin{cases} \bar{f}(x_i) = 1, & \text{si } L_i(\bar{x}) \geq 0, \\ \bar{f}(x_i) = -1, & \text{si } L_i(\bar{x}) < 0. \end{cases}$$

Il existe une multitude de fonctions de  $\mathcal{C}^0([a, b]; \mathbb{R})$  mais il faut contrôler son maximum pour qu'il vaille 1 et avoir ainsi  $|\bar{f}(x_i)| = \|\bar{f}\|_\infty$ ,  $\forall i \in \llbracket 0, n \rrbracket$ . On peut alors choisir  $\bar{f}$  affine sur chacun des intervalles  $[x_k, x_{k+1}]$ ,  $\forall k \in \llbracket 0, n-1 \rrbracket$ , puisque l'on connaît les valeurs aux extrémités de chaque

intervalle (+1 ou -1). En dehors de ces intervalles, on prend par exemple  $\bar{f}(x) = \bar{f}(x_0)$ ,  $\forall x \in [a, x_0]$ , et  $\bar{f}(x) = \bar{f}(x_n)$ ,  $\forall x \in [x_n, b]$ . Cette fonction est par construction continue sur  $[a, b]$  et vérifie

$$\begin{aligned} |\mathcal{L}_n(\bar{f})(\bar{x})| &= \left| \sum_{i=0}^n \bar{f}(x_i) L_i(\bar{x}) \right| \\ &= \sum_{i=0}^n |\bar{f}(x_i) L_i(\bar{x})| = \|\bar{f}\|_\infty \sum_{i=0}^n |L_i(\bar{x})| \\ &= \Lambda_n \|\bar{f}\|_\infty. \end{aligned}$$

Or on a

$$\|\mathcal{L}_n(\bar{f})\|_\infty = \sup_{x \in [a, b]} |\mathcal{L}_n(\bar{f})(x)| \geq |\mathcal{L}_n(\bar{f})(\bar{x})| = \Lambda_n \|\bar{f}\|_\infty.$$

ce qui donne

$$\sup_{\substack{f \in \mathcal{C}^0([a, b]; \mathbb{R}) \\ f \neq 0}} \frac{\|\mathcal{L}_n(f)\|_\infty}{\|f\|_\infty} \geq \Lambda_n.$$

□

**Théorème 4.1.4.** Pour toute fonction  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ , on a

$$\|f - \mathcal{L}_n(f)\|_\infty \leq (1 + \Lambda_n) \inf_{Q \in \mathbb{R}_n[X]} \|f - Q\|_\infty \quad (4.12)$$

*Preuve.* Soit  $Q \in \mathbb{R}_n[X]$ . Par unicité du théorème d'interpolation on a  $\mathcal{L}_n(Q) = Q$  et alors

$$\begin{aligned} \|f - \mathcal{L}_n(f)\|_\infty &= \|f - Q + \mathcal{L}_n(Q) - \mathcal{L}_n(f)\|_\infty \\ &\leq \|f - Q\|_\infty + \|\mathcal{L}_n(Q - f)\|_\infty \quad \text{par linéarité de } \mathcal{L}_n \\ &\leq \|f - Q\|_\infty + \Lambda_n \|f - Q\|_\infty \quad \text{par continuité de } \mathcal{L}_n \end{aligned}$$

d'où le résultat. □

- Pour les points équi-distants  $x_i = a + ih$ ,  $i \in \llbracket 0, n \rrbracket$  et  $h = (b - a)/n$ , on a la minoration suivante (voir [3] p. 49)

$$\Lambda_n \geq \frac{2^n}{4n^2} \quad (4.13)$$

et le comportement asymptotique

$$\Lambda_n \approx \frac{2^{n+1}}{e \cdot n \ln(n)} \quad \text{quand } n \rightarrow +\infty \quad (4.14)$$

- Pour les points de Tchebychev, on a la majoration suivante : il existe  $C > 0$  tel que

$$\Lambda_n \leq C \ln(n) \quad (4.15)$$

et le comportement asymptotique

$$\Lambda_n \approx \frac{2}{\pi} \ln(n) \quad \text{quand } n \rightarrow +\infty \quad (4.16)$$

**Proposition 4.1.2** (admis). Pour toute famille de points d'interpolation, il existe une fonction  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$  telle que la suite des polynômes d'interpolation de Lagrange associés ne converge pas uniformément.

**Proposition 4.1.3** (admis). Soit  $f$  une fonction lipschitzienne sur  $[a, b]$  à valeurs réelles, i.e. il existe une constante  $K \geq 0$  telle que  $\forall (x, y) \in [a, b]^2$ , on ait  $|f(x) - f(y)| \leq K|x - y|$ . Soient  $n \in \mathbb{N}^*$  et  $x_0, \dots, x_n$  les points de Tchebychev  $[a, b]$ . On note  $\mathcal{L}_n(f)$  le polynôme d'interpolation

de Lagrange associés aux couples de  $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$ .  
Alors la suite  $(\mathcal{L}_n(f))_{n \geq 1}$  des polynômes d'interpolation converge uniformément vers  $f$  sur  $[a, b]$ .

Pour conclure, l'interpolation de Lagrange en des points équidistants n'est à utiliser qu'avec un nombre de points assez faible : des phénomènes d'instabilités pouvant apparaître.

## 4.2 Polynôme d'interpolation de Lagrange-Hermite

### EXERCICE 4.2.1

Soient  $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$   $n + 1$  triplets de  $\mathbb{R}^3$ , où les  $x_i$  sont des points distincts deux à deux de l'intervalle  $[a, b]$ . Le polynôme d'interpolation de **Lagrange-Hermite**, noté  $H_n$ , associé aux  $n + 1$  triplets  $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$ , est défini par

$$H_n(x_i) = y_i \text{ et } H'_n(x_i) = z_i, \quad \forall i \in \llbracket 0, n \rrbracket \quad (4.1)$$

Q. 1

Quel est a priori le degré de  $H_n$  ?

On définit le polynôme  $P_n$  par

$$P_n(x) = \sum_{i=0}^n y_i A_i(x) + \sum_{i=0}^n z_i B_i(x) \quad (4.2)$$

avec, pour  $i \in \llbracket 0, n \rrbracket$ ,  $A_i$  et  $B_i$  polynômes de degré au plus  $2n + 1$  indépendants des valeurs  $y_i$  et  $z_i$ .

Q. 2

a. Déterminer des conditions suffisantes sur  $A_i$  et  $B_i$  pour que  $P_n$  vérifie (4.1).

b. En déduire les expressions de  $A_i$  et  $B_i$  en fonction de  $L_i$  et de  $L'_i(x_i)$  où

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}.$$

Q. 3

Démontrer qu'il existe un unique polynôme d'interpolation de Lagrange-Hermite de degré au plus  $2n + 1$  défini par (4.1).

### Correction

R. 1

On a  $2n + 2$  équations, donc a priori  $H_n$  est de degré  $2n + 1$ .

R. 2

a. D'après (4.2) on a pour tout  $j \in \llbracket 0, n \rrbracket$

$$P_n(x_j) = \sum_{i=0}^n y_i A_i(x_j) + \sum_{i=0}^n z_i B_i(x_j)$$

Pour avoir  $P_n(x_j) = y_j$  il suffit d'avoir

$$A_i(x_j) = \delta_{i,j} \text{ et } B_i(x_j) = 0, \quad \forall i \in \llbracket 0, n \rrbracket. \quad (R1.11)$$

De même, on a

$$P'_n(x_j) = \sum_{i=0}^n y_i A'_i(x_j) + \sum_{i=0}^n z_i B'_i(x_j)$$

et donc pour avoir  $P'_n(x_j) = z_j$  il suffit d'avoir

$$A'_i(x_j) = 0 \text{ et } B'_i(x_j) = \delta_{i,j}, \quad \forall i \in \llbracket 0, n \rrbracket. \quad (R1.12)$$

b. Soit  $i \in \llbracket 0, n \rrbracket$ . On commence par déterminer le polynôme  $A_i \in \mathbb{R}_{2n+1}[X]$  vérifiant

$$A_i(x_j) = \delta_{i,j} \text{ et } A_i'(x_j) = 0, \quad \forall j \in \llbracket 0, n \rrbracket.$$

Les points  $(x_j)_{j \in \llbracket 0, n \rrbracket \setminus \{i\}}$  sont racines doubles de  $A_i$ . Le polynôme  $L_i \in \mathbb{R}_n[X]$  admet les mêmes racines (simples) que  $A_i$  et donc  $L_i^2 \in \mathbb{R}_{2n}[X]$  admet les mêmes racines doubles que  $A_i$ . On peut alors écrire

$$A_i(x) = \alpha_i(x)L_i^2(x) \text{ avec } \alpha_i(x) \in \mathbb{R}_1[X].$$

Il reste à déterminer le polynôme  $\alpha_i$ . Or on a

$$A_i(x_i) = 1 \text{ et } A_i'(x_i) = 0.$$

Comme  $L_i(x_i) = 1$ , on obtient

$$A_i(x_i) = \alpha_i(x_i)L_i^2(x_i) = \alpha_i(x_i) = 1$$

et

$$A_i'(x_i) = \alpha_i'(x_i)L_i^2(x_i) + 2\alpha_i(x_i)L_i'(x_i)L_i(x_i) = \alpha_i'(x_i) + 2\alpha_i(x_i)L_i'(x_i) = 0$$

c'est à dire

$$\alpha_i(x_i) = 1 \text{ et } \alpha_i'(x_i) = -2L_i'(x_i).$$

Comme  $\alpha_i$  est un polynôme de degré 1 on en déduit

$$\alpha_i(x) = 1 - 2L_i'(x_i)(x - x_i)$$

et donc

$$A_i(x) = (1 - 2L_i'(x_i)(x - x_i))L_i^2(x). \quad (\text{R1.13})$$

On détermine ensuite le polynôme  $B_i \in \mathbb{R}_{2n+1}[X]$  vérifiant

$$B_i(x_j) = 0 \text{ et } B_i'(x_j) = \delta_{i,j}, \quad \forall j \in \llbracket 0, n \rrbracket.$$

Les points  $(x_j)_{j \in \llbracket 0, n \rrbracket \setminus \{i\}}$  sont racines doubles de  $B_i$  et le point  $x_i$  est racine simple. Le polynôme  $L_i^2 \in \mathbb{R}_{2n}[X]$  admet les mêmes racines doubles. On peut alors écrire

$$B_i(x) = C(x - x_i)L_i^2(x) \text{ avec } C \in \mathbb{R}.$$

Il reste à déterminer la constante  $C$ . Or  $L_i(x_i) = 1$  et comme  $B_i'(x_i) = 1$  on obtient

$$B_i'(x_i) = CL_i^2(x_i) + 2C(x_i - x_i)L_i'(x_i)L_i(x_i) = C = 1$$

ce qui donne

$$B_i(x) = (x - x_i)L_i^2(x). \quad (\text{R1.14})$$

On vient de démontrer l'existence en construisant un polynôme de degré  $2n + 1$  vérifiant (4.1).

### R. 3

Deux démonstrations pour l'unicité sont proposées (la deuxième donne aussi l'existence).

**dém. 1:** Soient  $P$  et  $Q$  deux polynômes de  $\mathbb{R}_{2n+1}[X]$  vérifiant (4.1). Le polynôme  $R = P - Q \in \mathbb{R}_{2n+1}[X]$  admet alors  $n + 1$  racines doubles distinctes  $(x_0, \dots, x_n)$ . Or le seul polynôme de  $\mathbb{R}_{2n+1}[X]$  ayant  $n + 1$  racines doubles est le polynôme nul et donc  $R = 0$ , i.e.  $P = Q$ .

**dém. 2:** Soit  $\Phi : \mathbb{R}_{2n+1}[X] \longrightarrow \mathbb{R}^{2n+2}$  définie par

$$\forall P \in \mathbb{R}_{2n+1}[X], \quad \Phi(P) = (P(x_0), \dots, P(x_n), P'(x_0), \dots, P'(x_n)).$$

L'existence et l'unicité du polynôme  $H_n$  est équivalente à la bijectivité de l'application  $\Phi$ . Or celle-ci est une application linéaire entre deux espaces de dimension  $2n + 2$ . Elle est donc bijective si et seulement si elle est injective (ou surjective). Pour vérifier l'injectivité de  $\Phi$  il est nécessaire et suffisant de vérifier que son noyau est réduit au polynôme nul.

Soit  $P \in \ker \Phi$ . On a alors  $\Phi(P) = \mathbf{0}_{2n+2}$  et donc  $(x_0, \dots, x_n)$  sont  $n + 1$  racines doubles distinctes

de  $P$ . Or le seul polynôme de  $\mathbb{R}_{2n+1}[X]$  ayant  $n + 1$  racines doubles est le polynôme nul et donc  $P = 0$ .

~~~~~Fin correction~~~~~

**Définition 4.2.1.** Soient  $n \in \mathbb{N}^*$  et  $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$   $(n + 1)$  triplets de  $\mathbb{R}^3$ , où les  $x_i$  sont des points distincts deux à deux de  $\mathbb{R}$ . Le **polynôme d'interpolation de Lagrange-Hermite**, noté  $H_n$ , associé aux  $(n + 1)$  triplets  $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$ , est défini par

$$H_n(x) = \sum_{i=0}^n y_i A_i(x) + \sum_{i=0}^n z_i B_i(x) \tag{4.17}$$

avec

$$A_i(x) = (1 - 2(x - x_i)L'_i(x_i))L_i^2(x) \quad \text{et} \quad B_i(x) = (x - x_i)L_i^2(x) \tag{4.18}$$

où

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}.$$

**Théorème 4.2.1.** Le **polynôme d'interpolation de Lagrange-Hermite**,  $H_n$ , associé aux  $n + 1$  triplets  $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$ , est l'unique polynôme de degré au plus  $2n + 1$ , vérifiant

$$H_n(x_i) = y_i \quad \text{et} \quad H'_n(x_i) = z_i, \quad \forall i \in \llbracket 0, n \rrbracket \tag{4.19}$$

**EXERCICE 4.2.2**

Soit  $f \in \mathcal{C}^{2n+2}([a, b]; \mathbb{R})$ . On suppose de plus que,  $\forall i \in \llbracket 0, n \rrbracket$ ,  $x_i \in [a, b]$ ,  $y_i = f(x_i)$  et  $z_i = f'(x_i)$ . On note

$$\pi_n^2(x) = \prod_{i=0}^n (x - x_i)^2$$

et  $H_n$  le polynôme d'interpolation de Lagrange-Hermite associé aux triplets  $(x_i, f(x_i), f'(x_i))_{i \in \llbracket 0, n \rrbracket}$ .

**Q. 1** Montrer que

$$|f(x) - H_n(x)| \leq \frac{\|f^{(2n+2)}\|_\infty}{(2n + 2)!} \pi_n^2(x). \tag{4.1}$$

**Indications :** Etudier les zéros de la fonction  $F(y) = f(y) - H_n(y) - \frac{f(x) - H_n(x)}{\pi_n^2(x)} \pi_n^2(y)$  et appliquer le théorème de Rolle.

**Correction**

**R. 1**

Soit  $i \in \llbracket 1, n \rrbracket$ , on a  $f(x_i) - H_n(x_i) = 0$  et l'inégalité (4.1) est donc vérifiée pour  $x = x_i$ . Soit  $x \in [a, b]$  tel que  $x \neq x_i, \forall i \in \llbracket 1, n \rrbracket$ . On a alors  $\pi_n^2(x) \neq 0$ . Comme  $f \in \mathcal{C}^{2n+2}([a; b]; \mathbb{R})$ ,  $H_n \in \mathbb{R}_{2n+1}[X]$  et  $\pi_n \in \mathbb{R}_{n+1}[X]$ , on en déduit que

$$F \in \mathcal{C}^{2n+2}([a; b]; \mathbb{R}).$$

On note que  $\pi_n^2$  admet  $(x_0, \dots, x_n)$  comme racines doubles distinctes. Par construction  $f - H_n$  admet les mêmes racines doubles. On en déduit alors que  $F$  admet aussi  $(x_0, \dots, x_n)$  comme racines doubles. De plus, on a  $F(x) = 0$  (i.e.  $x$  est racine simple) et donc

$F$  admet au moins  $2n + 3$  racines (comptées avec leurs multiplicités).

Les points  $x, x_0, \dots, x_n$  étant distincts, la fonction  $F'$  admet par le théorème de Rolle  $n + 1$  zéros

distincts entre eux et distincts des points  $x, x_0, \dots, x_n$ . De plus les points  $x_0, \dots, x_n$  sont racines de  $F'$  puisque racines doubles de  $F$ . On en déduit alors que

$F'$  admet au moins  $2n + 2$  racines distinctes deux à deux.

Par applications successives du théorème de Rolle, on abouti a :

$F^{(2n+2)}$  admet au moins une racine notée  $\xi_x \in ]a, b[$ .

On a alors

$$F^{(2n+2)}(\xi_x) = 0 = f^{(2n+2)}(\xi_x) - H_n^{(2n+2)}(\xi_x) - \frac{f(x) - H_n(x)}{\pi_n^2(x)} \frac{d^{2n+2}\pi_n^2}{dx^{2n+2}}(\xi_x)$$

Comme  $H_n \in \mathbb{R}_{2n+1}[X]$  on a  $H_n^{(2n+2)} \equiv 0$ . De plus comme  $\pi_n^2(x) = \prod_{i=0}^n (x - x_i)^2 \in \mathbb{R}_{2n+2}[X]$  sa dérivée d'ordre  $2n + 2$  est constante et

$$\frac{d^{2n+2}\pi_n^2}{dx^{2n+2}} = (2n + 2)!$$

On en déduit alors

$$f^{(2n+2)}(\xi_x) = \frac{f(x) - H_n(x)}{\pi_n^2(x)} (2n + 2)!$$

On a donc montrer que  $\forall x \in [a, b] \exists \xi_x \in ]a, b[$  tels que

$$f(x) - H_n(x) = \frac{\pi_n^2(x)}{(2n + 2)!} f^{(2n+2)}(\xi_x).$$

Comme  $\pi_n^2(x) \geq 0$  on obtient bien (4.1).

~~~~~ Fin correction ~~~~~

**Théorème 4.2.2.** Soient  $n \in \mathbb{N}^*$  et  $x_0, \dots, x_n$ ,  $n + 1$  points distincts de l'intervalle  $[a, b]$ . Soient  $f \in \mathcal{C}^{2n+2}([a, b]; \mathbb{R})$  et  $H_n$  le polynôme d'interpolation de Lagrange-Hermite associé aux  $n + 1$  triplets  $(x_i, f(x_i), f'(x_i))_{i \in [0, n]}$ . On a alors  $\forall x \in [a, b], \exists \xi_x \in (\min(x_i, x), \max(x_i, x))$ , tels que

$$f(x) - H_n(x) = \frac{f^{(2n+2)}(\xi_x)}{(2n + 2)!} \prod_{i=0}^n (x - x_i)^2 \quad (4.20)$$

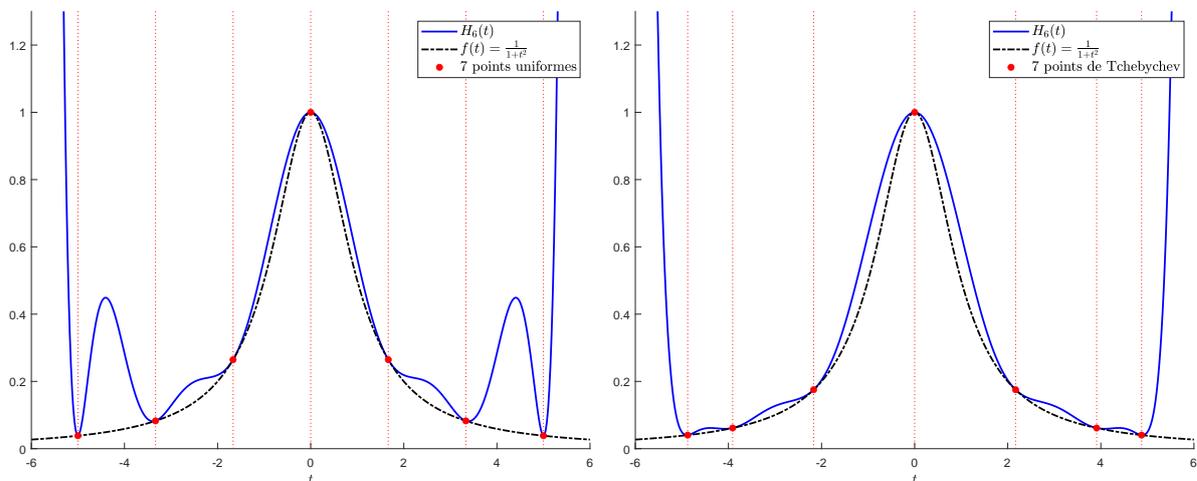


Figure 4.9: Polynôme d'interpolation de lagrange-Hermite avec  $n = 6$  (7 points) pour la fonction  $f : t \rightarrow 1/(1 + t^2)$ . A gauche avec des points uniformément répartis et à droite avec des points de Tchebychev

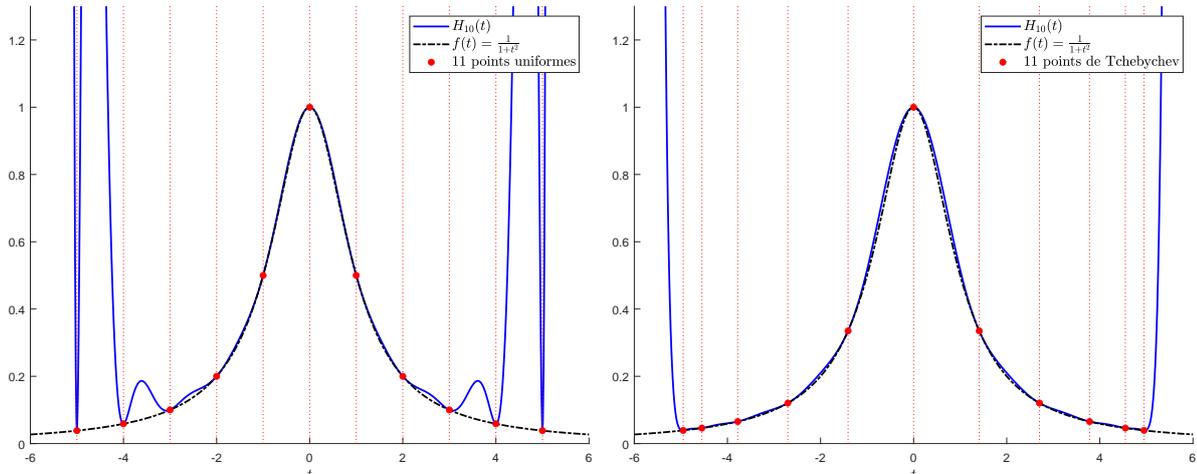


Figure 4.10: Polynôme d'interpolation de lagrange-Hermite avec  $n = 10$  (11 points) pour la fonction  $f : t \rightarrow 1/(1 + t^2)$ . A gauche avec des points uniformément répartis et à droite avec des points de Tchebychev

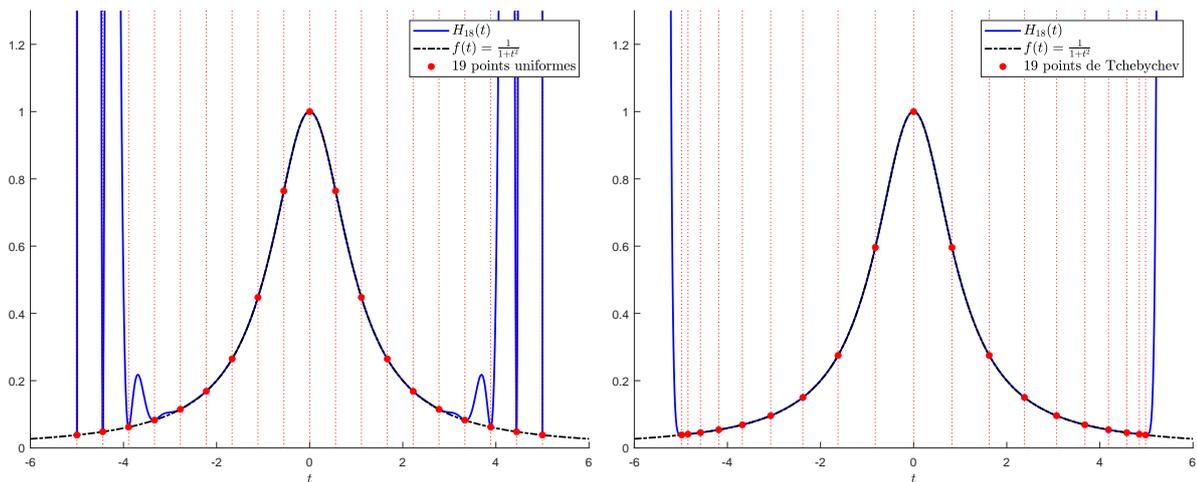


Figure 4.11: Polynôme d'interpolation de lagrange-Hermite avec  $n = 18$  (19 points) pour la fonction  $f : t \rightarrow 1/(1 + t^2)$ . A gauche avec des points uniformément répartis et à droite avec des points de Tchebychev

**EXERCICE 4.2.3**

Ecrire une fonction algorithmique **Hermite** permettant de calculer  $H_n$  (polynôme d'interpolation de Lagrange-Hermite associé aux  $n + 1$  triplets  $(x_i, y_i, z_i)_{i \in [0, n]}$ ) en  $t \in \mathbb{R}$ .

**Correction**

**But :** Calculer le polynôme  $H_n(t)$  défini par l'équation (4.17) de [?]

**Données :**  $\mathbf{X}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $X(i) = x_{i-1} \forall i \in [1, n + 1]$  et  $X(i) \neq X(j)$  pour  $i \neq j$ ,  
 $\mathbf{Y}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $Y(i) = y_{i-1} \forall i \in [1, n + 1]$ ,  
 $\mathbf{Z}$  : vecteur/tableau de  $\mathbb{R}^{n+1}$ ,  $Z(i) = z_{i-1} \forall i \in [1, n + 1]$ ,  
 $t$  : un réel.

**Résultat :** pH : le réel pH =  $H_n(t)$ .

D'après la Définition ?? de [?], on a

$$H_n(t) = \sum_{i=0}^n y_i A_i(t) + \sum_{i=0}^n z_i B_i(t) = \sum_{i=0}^n (y_i A_i(t) + z_i B_i(t))$$

avec

$$A_i(t) = (1 - 2L'_i(x_i)(t - x_i))L_i^2(t) \text{ et } B_i(t) = (t - x_i)L_i^2(t)$$

où

$$L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - x_j}{x_i - x_j}.$$

Pour rendre effectif le calcul de  $H_n(t)$ , il reste à déterminer  $L'_i(x_i)$ . On a

$$L'_i(t) = \sum_{\substack{k=0 \\ k \neq i}}^n \frac{1}{x_i - x_k} \prod_{\substack{j=0 \\ j \neq i \\ j \neq k}}^n \frac{t - x_j}{x_i - x_j}$$

d'où

$$L'_i(x_i) = \sum_{\substack{k=0 \\ k \neq i}}^n \frac{1}{x_i - x_k}. \quad (4.21)$$

La fonction que l'on va écrire use (et certains diront abuse) de fonctions.

---

**Algorithme 4.2** Fonction **Hermite** permettant de calculer le polynôme d'interpolation de Lagrange-Hermite  $H_n(t)$  définit par (4.17)

---

- 1: **Fonction** pH  $\leftarrow$  **Hermite**(  $X, Y, Z, t$  )
  - 2: pH  $\leftarrow$  0
  - 3: **Pour**  $i \leftarrow 0$  à  $n$  **faire**
  - 4: pH  $\leftarrow$  pH + **polyA**( $i, X, t$ ) \*  $Y(i + 1)$  + **polyB**( $i, X, t$ ) \*  $Z(i + 1)$
  - 5: **Fin**
  - 6: **Fin**
- 

Les différentes fonctions utilisées pour la fonction **Hermite** (directement ou indirectement) sont les suivantes :

**polyA** : calcul du polynôme  $A_i$  en  $t$ , (données  $i, X, t$ )

**polyB** : calcul du polynôme  $B_i$  en  $t$ , (données  $i, X, t$ )

**polyL** : calcul du polynôme  $L_i$  en  $t$ , (données  $i, X, t$ )

**polyLp** : calcul de  $L'_i(x_i)$ , (données  $i, X$ )

---

**Algorithme 4.3** Fonction `polyA` permettant de calculer le polynôme  $A_i$  en  $t \in \mathbb{R}$  donné par  $A_i(t) = (1 - 2L'_i(x_i)(t - x_i))L_i^2(t)$

```
1: Fonction  $y \leftarrow \text{polyA}(i, X, t)$ 
2:  $y \leftarrow (1 - 2 * \text{polyLp}(i, X) * (t - X(i + 1))) * (\text{polyL}(i, X, t))^2$ 
3: Fin
```

**Algorithme 4.5** Fonction `polyL` permettant de calculer le polynôme  $L_i$  en  $t \in \mathbb{R}$  donné par  $L_i(t) = \prod_{j=0, j \neq i}^n \frac{t - x_j}{x_i - x_j}$

```
1: Fonction  $y \leftarrow \text{polyL}(i, X, t)$ 
2:  $y \leftarrow 1$ 
3: Pour  $j \leftarrow 0$  à  $n$ , ( $j \sim i$ ) faire
4:    $y \leftarrow y * (t - X(j + 1)) / (X(i + 1) - X(j + 1))$ 
5: Fin
6: Fin
```

**Algorithme 4.4** Fonction `polyB` permettant de calculer le polynôme  $B_i$  en  $t \in \mathbb{R}$  donné par  $B_i(t) = (t - x_i)L_i^2(t)$

```
1: Fonction  $y \leftarrow \text{polyB}(i, X, t)$ 
2:  $y \leftarrow (t - X(i + 1)) * (\text{polyL}(i, X, t))^2$ 
3: Fin
```

**Algorithme 4.6** Fonction `polyLp` permettant de calculer  $L'_i(x_i) = \sum_{k=0, k \neq i}^n \frac{1}{x_i - x_k}$

```
1: Fonction  $y \leftarrow \text{polyLp}(i, X)$ 
2:  $y \leftarrow 0$ 
3: Pour  $k \leftarrow 0$  à  $n$ , ( $k \sim i$ ) faire
4:    $y \leftarrow y + 1 / (X(i + 1) - X(k + 1))$ 
5: Fin
6: Fin
```

Bien évidemment une telle écriture est loin d'être optimale mais elle a l'avantage d'être facile à programmer et facile à lire car elle "colle" aux formules mathématiques.

On laisse le soin au lecteur d'écrire des fonctions plus performantes...

~~~~~ Fin correction ~~~~~

## 4.3 Exercices

### EXERCICE 4.3.1

Q. 1

- Ecrire explicitement un polynôme  $P$  de degré 2 passant par les points  $A = (1; 2)$ ,  $B = (2; 6)$  et  $C = (3; 12)$ .
- Démontrer que le polynôme  $P$  est l'unique polynôme de degré 2 passant par les points  $A$ ,  $B$  et  $C$ .

Q. 2

Ecrire explicitement un polynôme  $Q$  de degré 3 tel que

$$\begin{aligned} Q(1) &= 4, & Q(2) &= 5, \\ Q'(1) &= 3, & Q'(2) &= 2. \end{aligned}$$

### EXERCICE 4.3.2

Q. 1

- Construire les polynômes  $h_{00}$ ,  $h_{10}$ ,  $h_{01}$  et  $h_{11}$  de degré 3 vérifiant

$$h_{00}(0) = 1, h'_{00}(0) = h_{00}(1) = h'_{00}(1) = 0, \quad (4.1)$$

$$h_{10}(1) = 1, h_{10}(0) = h'_{10}(0) = h'_{10}(1) = 0, \quad (4.2)$$

$$h'_{01}(0) = 1, h_{01}(0) = h_{01}(1) = h'_{01}(1) = 0, \quad (4.3)$$

$$h'_{11}(1) = 1, h_{11}(0) = h'_{11}(0) = h_{11}(1) = 0; \quad (4.4)$$

- Démontrer que  $\{h_{00}, h_{10}, h_{01}, h_{11}\}$  est une base de  $\mathbb{R}_3[X]$ .

On pose

$$P(x) = \alpha h_{00}(x) + \beta h_{10}(x) + \gamma h_{01}(x) + \delta h_{11}(x). \quad (4.5)$$

Q. 2

Quelles sont les particularités de  $P$  ?

Soient  $a$  et  $b$  deux réels,  $a < b$  et  $Q$  un polynôme de degré  $\mathbb{R}_3[X]$  vérifiant

$$Q(a) = u_a, \quad Q'(a) = v_a, \quad Q(b) = u_b \quad \text{et} \quad Q'(b) = v_b. \quad (4.6)$$

On note  $x = \varphi(t) = \frac{t-a}{b-a}$  le changement de variable de  $[a; b]$  dans  $[0; 1]$  et on définit

$$H_{a0} \stackrel{\text{def}}{=} h_{00} \circ \varphi, \quad H_{b0} \stackrel{\text{def}}{=} h_{10} \circ \varphi, \quad H_{a1} \stackrel{\text{def}}{=} h_{01} \circ \varphi, \quad \text{et} \quad H_{b1} \stackrel{\text{def}}{=} h_{11} \circ \varphi.$$

Q. 3

- a. Déterminer les 4 équations vérifiées par chacune des fonctions  $H_{a0}$ ,  $H_{a1}$ ,  $H_{b0}$  et  $H_{b1}$ ? (i.e. 4 équations par fonction)
- b. Exprimer le polynôme  $Q$  avec les fonctions  $H_{a0}$ ,  $H_{b0}$ ,  $H_{a1}$  et  $H_{b1}$ .
- c. Le polynôme  $Q$  est-il l'unique polynôme de  $\mathbb{R}_3[X]$  vérifiant (4.6)? Justifier.

## EXERCICE 4.3.3

Soit  $t_0 < t_1$  deux nombres réels et soit  $\varepsilon$  un réel tel que  $0 < \varepsilon < \frac{t_1 - t_0}{2}$ .

Q. 1

Expliciter un polynôme  $P_\varepsilon$  de degré 3 tel que

$$P_\varepsilon(t_0) = P_\varepsilon(t_0 + \varepsilon) = 1, \quad (4.1)$$

$$P_\varepsilon(t_1) = P_\varepsilon(t_1 + \varepsilon) = 0. \quad (4.2)$$

On note  $\Phi_0(t) = \lim_{\varepsilon \rightarrow 0} P_\varepsilon(t)$ .

Q. 2

- a. Montrer que  $\Phi_0(t_0) = 1$ ,  $\Phi_0'(t_0) = 0$ ,  $\Phi_0(t_1) = 0$  et  $\Phi_0'(t_1) = 0$  (i.e.  $\Phi_0$  est une fonction de base des polynômes de degré 3 pour l'interpolation de Hermite).
- b. Peut-on obtenir toutes les fonctions de base de Hermite par des procédés analogues. Si oui, expliquer comment!

## EXERCICE 4.3.4

Soient  $(x_i)_{i \in \mathbb{N}}$  une suite de points distincts de l'intervalle  $[a, b]$  et  $f$  une fonction définie sur  $[a, b]$  à valeurs réelles.

On désigne par  $f[\ ]$  les **différences divisées** de la fonction  $f$  définie par

$$f[x_i] = f(x_i), \quad \forall i \in \mathbb{N}, \quad (\text{ordre } 0) \quad (4.1)$$

et

$$f[x_k, \dots, x_{k+r}] = \frac{f[x_{k+1}, \dots, x_{k+r}] - f[x_k, \dots, x_{k+r-1}]}{x_{k+r} - x_k}, \quad \forall k \in \mathbb{N}, \quad \forall r \in \mathbb{N}^*, \quad (\text{ordre } r) \quad (4.2)$$

Q. 1

Montrer que

$$f[x_k, \dots, x_{k+r}] = \sum_{i=k}^{k+r} \frac{f(x_i)}{\prod_{\substack{j=k \\ j \neq i}}^{k+r} (x_i - x_j)}, \quad \forall k \in \mathbb{N}, \quad \forall r \in \mathbb{N}^*. \quad (4.3)$$

Q. 2

Soit  $\sigma$  une permutation des entiers  $\{k, \dots, k+r\}$ . Montrer que

$$f[x_k, \dots, x_{k+r}] = f[x_{\sigma(1)}, \dots, x_{\sigma(r+1)}]. \quad (4.4)$$

On note  $Q_{k,r}$  le polynôme d'interpolation associé aux  $r+1$  couples  $(x_{k+i}, f(x_{k+i}))_{i \in [0, r]}$ .

Q. 3

- a. Exprimer le polynôme  $Q_{k,1}$  en fonction de  $Q_{k,0}$ .
- b. Exprimer le polynôme  $Q_{k,1}$  en fonction de  $Q_{k,0}$  et  $Q_{k+1,0}$ .
- c. En déduire que

$$Q_{k,1}(x) = Q_{k,0}(x) + f[x_k, x_{k+1}](x - x_k). \quad (4.5)$$

Q. 4

- a. Exprimer le polynôme  $Q_{k,2}$  en fonction de  $Q_{k,1}$ .
- b. Exprimer le polynôme  $Q_{k,2}$  en fonction de  $Q_{k,1}$  et  $Q_{k+1,1}$ .
- c. En déduire que

$$Q_{k,2}(x) = Q_{k,1}(x) + f[x_k, x_{k+1}, x_{k+2}](x - x_k)(x - x_{k+1}). \quad (4.6)$$

Q. 5

- a. Montrer que

$$Q_{k,r}(x) = Q_{k,r-1}(x) + f[x_k, \dots, x_{k+r}] \prod_{j=0}^{r-1} (x - x_{k+j}) \quad (4.7)$$

**Indication :** Effectuer une démonstration par récurrence en écrivant le polynôme  $Q_{k,r}$  sous deux formes : l'une en fonction de  $Q_{k,r-1}$  et l'autre en fonction de  $Q_{k,r-1}$  et  $Q_{k+1,r-1}$ .

- b. En déduire

$$Q_{k,r}(x) = f[x_k] + \sum_{i=1}^r f[x_k, \dots, x_{k+i}] \prod_{j=0}^{i-1} (x - x_{k+j}) \quad (4.8)$$

Q. 6

On suppose que  $f \in C^r([a, b]; \mathbb{R})$ . Montrer qu'il existe  $\xi \in ]\min_{i \in \llbracket 0, r \rrbracket} x_{k+i}; \max_{i \in \llbracket 0, r \rrbracket} x_{k+i}[$  tel que

$$f[x_k, \dots, x_{k+r}] = \frac{f^{(r)}(\xi)}{r!}. \quad (4.9)$$

Q. 7

On suppose  $f \in C^{r+1}([a, b]; \mathbb{R})$ . Montrer que,  $\forall x \in [a, b]$ , il existe  $\xi_x$  appartenant au plus petit intervalle fermé contenant  $x, x_k, \dots, x_{k+r}$  tel que

$$f(x) - Q_{k,r}(x) = \frac{\prod_{j=0}^r (x - x_{k+j})}{(r+1)!} f^{(r+1)}(\xi_x). \quad (4.10)$$

### EXERCICE 4.3.5

Soit  $\mathbf{X} = (x_i)_{i \in \llbracket 0, n \rrbracket}$   $n+1$  points deux à deux distincts de l'intervalle  $[a, b]$ . On note  $s$  le changement de variables  $s : t \rightarrow a + (b-a)t$  de  $[0, 1]$  à valeurs dans  $[a, b]$ . Pour tout  $i \in \llbracket 0, n \rrbracket$ , on note  $t_i = s^{-1}(x_i) = \frac{x_i - a}{b - a}$  et  $\mathbf{T} = (t_i)_{i \in \llbracket 0, n \rrbracket}$ .

Les polynômes d'interpolation de Lagrange  $\mathcal{L}_n(f)$  et  $\mathcal{L}_n(g)$  associés respectivement aux points  $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$  et  $(t_i, g(t_i))_{i \in \llbracket 0, n \rrbracket}$  sont définis par

$$\begin{aligned} \mathcal{L}_n(f)(x) &= \sum_{i=0}^n L_i^{\mathbf{X}}(x) f(x_i) \quad \text{avec} \quad L_i^{\mathbf{X}}(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \\ \mathcal{L}_n(g)(t) &= \sum_{i=0}^n L_i^{\mathbf{T}}(t) g(t_i) \quad \text{avec} \quad L_i^{\mathbf{T}}(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - t_j}{t_i - t_j} \end{aligned}$$

Q. 1

Montrer que  $L_i^{\mathbf{X}} \circ s = L_i^{\mathbf{T}}$ .

Q. 2

En déduire que  $\mathcal{L}_n(f \circ s) = \mathcal{L}_n(f) \circ s = \mathcal{L}_n(g)$ .

### Correction

R. 1

$$\begin{aligned}
 L_i^{\mathbf{X}} \circ s(t) &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{s(t) - x_j}{x_i - x_j} = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{s(t) - s(t_j)}{s(t_i) - s(t_j)} \\
 &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{a + t(b-a) - (a + t_j(b-a))}{a + t_i(b-a) - (a + t_j(b-a))} \\
 &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - t_j}{t_i - t_j} = L_i^{\mathbf{T}}(t)
 \end{aligned}$$

R. 2

On a

$$\begin{aligned}
 \mathcal{L}_n(f \circ s)(t) &\stackrel{\text{def}}{=} \sum_{i=0}^n L_i^{\mathbf{T}}(t) f \circ s(t_i) \\
 &= \sum_{i=0}^n L_i^{\mathbf{T}}(t) g(t_i) = \mathcal{L}_n(g)
 \end{aligned}$$

et

$$\begin{aligned}
 \mathcal{L}_n(f) \circ s(t) &\stackrel{\text{def}}{=} \sum_{i=0}^n L_i^{\mathbf{X}} \circ s(t) f(x_i) \\
 &= \sum_{i=0}^n L_i^{\mathbf{T}}(t) f \circ s(t_i) = \mathcal{L}_n(g)
 \end{aligned}$$

~~~~~Fin correction~~~~~

**EXERCICE 4.3.6 Spline cubique**

Soient  $n \geq 3$  un entier et  $a = x_0 < x_1 \dots < x_{n-1} < x_n = b$  une discrétisation régulière de l'intervalle  $[a, b]$ . On note  $h = x_{k+1} - x_k$ . Une fonction  $s$  définie sur  $[a, b]$  à valeurs réelles s'appelle **spline cubique** si elle est deux fois continûment différentiable et si, sur chaque intervalle  $[x_{k-1}, x_k]$ , elle est polynomiale de degré inférieur ou égal à 3.

Soit  $f \in \mathcal{C}^2([a, b]; \mathbb{R})$  et  $s$  une spline cubique vérifiant

$$s(x_i) = f(x_i) = f_i, \quad \forall i \in \llbracket 0, n \rrbracket. \quad (4.1)$$

Q. 1

Montrer que si

$$s''(b)(f'(b) - s'(b)) = s''(a)(f'(a) - s'(a)) \quad (4.2)$$

alors

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (f''(x))^2 dx. \quad (4.3)$$

**Indications :** Poser  $r = f - s$  et montrer par intégrations par parties que  $\int_a^b s''(x)r''(x)dx = 0$ .

Soient  $k \in \llbracket 1, n \rrbracket$  et  $S_k$  un polynôme de degré inférieur ou égal à 3 vérifiant

$$\begin{cases} S_k(x_{k-1}) &= f_{k-1} & (4.4a) \end{cases}$$

$$\begin{cases} S_k(x_k) &= f_k & (4.4b) \end{cases}$$

$$\begin{cases} S_k''(x_{k-1}) &= m_{k-1} & (4.4c) \end{cases}$$

$$\begin{cases} S_k''(x_k) &= m_k. & (4.4d) \end{cases}$$

Q. 2

a. Montrer l'existence et l'unicité du polynôme  $S_k$ .

b. Montrer que polynôme  $S_k$  peut s'écrire sous la forme

$$S_k(x) = a_k(x_k - x)^3 + b_k(x - x_{k-1})^3 + \alpha_k(x_k - x) + \beta_k(x - x_{k-1}) \quad (4.5)$$

en explicitant les coefficients  $(a_k, b_k, \alpha_k, \beta_k)$  en fonction de  $(f_{k-1}, f_k, m_{k-1}, m_k)$  et  $h$ .

On note  $g$  la fonction dont la restriction à chaque intervalle  $[x_{k-1}; x_k]$ ,  $k \in \llbracket 1, n \rrbracket$ , est  $S_k$ .

Q. 3

a. Vérifier que  $g$  est bien définie sur  $[a; b]$ .

b. Montrer que  $g$  est une spline cubique si et seulement si,  $\forall k \in \llbracket 1, n-1 \rrbracket$ ,

$$m_{k+1} + 4m_k + m_{k-1} = \frac{6}{h^2}(f_{k+1} - 2f_k + f_{k-1}). \quad (4.6)$$

Q. 4

a. Montrer qu'une condition nécessaire et suffisante pour que  $g$  soit une spline cubique et vérifie  $g''(a) = 0$ ,  $g''(b) = 0$ , est que le vecteur  $\mathbf{M} \in \mathbb{R}^{n+1} = (m_0, m_1, \dots, m_n)^t$  soit solution d'un système linéaire de la forme

$$\mathbb{A}\mathbf{M} = \mathbf{b} \quad (4.7)$$

que l'on précisera.

b. Montrer que la matrice  $\mathbb{A}$  est inversible.

## Correction

R. 1

**Théorème 4.3.1** (Intégration par parties). Soient  $u \in \mathcal{C}^1([a; b]; \mathbb{R})$  et  $v \in \mathcal{C}^1([a; b]; \mathbb{R})$  alors

$$\int_a^b u'(x)v(x)dx = [u(x)v(x)]_a^b - \int_a^b u(x)v'(x)dx.$$

On pose  $r = f - s$ . On a alors  $r \in \mathcal{C}^2([a; b]; \mathbb{R})$  et,

$$\forall i \in \llbracket 0, n \rrbracket, r(x_i) = 0. \quad (R6.15)$$

De plus

$$\begin{aligned} \int_a^b (f''(x))^2 dx &= \int_a^b (s''(x) + r''(x))^2 dx \\ &= \int_a^b (s''(x))^2 dx + 2 \int_a^b s''(x)r''(x)dx + \int_a^b (r''(x))^2 dx \end{aligned} \quad (R6.16)$$

Montrons que  $\int_a^b s''(x)r''(x)dx = 0$ .

On ne peut effectuer une intégration par partie pour  $\int_a^b s''(x)r''(x)dx$  car  $r''$  et  $s''$  ne sont pas dérivables. Par contre, on a

$$\int_a^b s''(x)r''(x)dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s''(x)r''(x)dx$$

et, sur chaque intervalle  $[x_{i-1}, x_i]$ ,  $s''$  est un polynôme de degré au plus 1. On a donc  $s'' \in \mathcal{C}^1([x_{i-1}, x_i]; \mathbb{R})$  et  $r' \in \mathcal{C}^1([x_{i-1}, x_i]; \mathbb{R})$  et il est alors possible de faire une intégration par partie avec  $u = r'$  et  $v = s''$  sur  $[x_{i-1}, x_i]$  :

$$\int_{x_{i-1}}^{x_i} s''(x)r''(x)dx = [s''(x)r'(x)]_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} s'''(x)r'(x)dx. \quad (R6.17)$$

Or, sur  $[x_{i-1}, x_i]$ ,  $s \in \mathbb{R}^3[X]$  et donc  $s'''$  est constante, ce qui donne, en utilisant (R6.15),

$$\int_{x_{i-1}}^{x_i} s'''(x)r'(x)dx = s'''(x_i) \int_{x_{i-1}}^{x_i} r'(x)dx = s'''(x_i)(r(x_i) - r(x_{i-1})) = 0.$$

De (R6.17), On a alors,  $\forall i \in \llbracket 1, n \rrbracket$ ,

$$\int_{x_{i-1}}^{x_i} s'''(x)r'(x)dx = s''(x_i)h'(x_i) - s''(x_{i-1})h'(x_{i-1}).$$

En sommant, on abouti a

$$\int_a^b s''(x)r''(x)dx = s''(x_n)h'(x_n) - s''(x_0)h'(x_0) = s''(b)h'(b) - s''(a)h'(a).$$

Sous l'hypothèse (4.2) on a bien  $\int_a^b s''(x)r''(x)dx = 0$ . L'équation (R6.16) devient alors

$$\int_a^b (f''(x))^2 dx = \int_a^b (s''(x))^2 dx + \int_a^b (r''(x))^2 dx.$$

D'où

$$\int_a^b (f''(x))^2 dx \geq \int_a^b (s''(x))^2 dx.$$

**R. 2**

a. Soit  $\Phi_k : \mathbb{R}^3[X] \rightarrow \mathbb{R}^4$  définie par

$$\Phi_k(P) = (P(x_{k-1}), P(x_k), P''(x_{k-1}), P''(x_k))^t.$$

L'existence et l'unicité du polynôme  $S_k$  est équivalente à la bijectivité de  $\Phi_k$ . Cette dernière étant une application entre deux espaces vectoriels de même dimension finie 4, elle est bijective si et seulement si elle est injective. Pour établir l'injectivité de  $\Phi_k$  il faut montrer que son noyau est réduit au polynôme nul.

Soit  $P \in \ker \Phi_k$  alors  $\Phi_k(P) = 0_{\mathbb{R}^4}$ . On en déduit que  $x_{k-1}$  et  $x_k$  sont racines de  $P$ , et  $P$  s'écrit alors sous la forme

$$P(x) = (x - x_{k-1})(x - x_k)Q(x)$$

avec  $Q(x) = \alpha x + \beta$  polynôme de degré 1.

On a

$$P'(x) = (x - x_{k-1})Q'(x) + (x - x_k)Q'(x) + \alpha(x - x_{k-1})(x - x_k)$$

et

$$P''(x) = 2(Q'(x) + \alpha(x - x_{k-1}) + \alpha(x - x_k)).$$

Comme  $P''(x_{k-1}) = P''(x_k) = 0$ , on obtient

$$\begin{cases} Q(x_{k-1}) + \alpha(x_{k-1} - x_k) = 0, \\ Q(x_k) + \alpha(x_k - x_{k-1}) = 0, \end{cases} \iff \begin{cases} \alpha(x_{k-1} - h) + \beta = 0, \\ \alpha(x_k + h) + \beta = 0, \end{cases}$$

En soustrayant la première équation à la deuxième, on obtient  $3\alpha h = 0$ . Comme  $h \neq 0$ , on obtient  $\alpha = \beta = 0$ . D'où  $Q \equiv 0$  et donc  $P \equiv 0$ .

b. On a  $S_k''(x) = 6a_k(x_k - x) + 6b_k(x - x_{k-1})$ . Pour déterminer  $a_k$  et  $b_k$ , on utilise les équations (4.4c) et (4.4d) qui deviennent respectivement  $6ha_k = m_{k-1}$  et  $6hb_k = m_k$ . On obtient

$$a_k = \frac{m_{k-1}}{6h} \text{ et } b_k = \frac{m_k}{6h}.$$

Pour déterminer  $\alpha_k$  et  $\beta_k$  on utilise les équations (4.4c) et (4.4d) qui deviennent respectivement

$$a_k h^3 + \alpha_k h = f_{k-1} \text{ et } b_k h^3 + \beta_k h = f_k.$$

En remplaçant  $a_k$  et  $b_k$  par leurs valeurs, on obtient

$$\alpha_k = \frac{f_{k-1}}{h} - \frac{h}{6}m_{k-1} \text{ et } \beta_k = \frac{f_k}{h} - \frac{h}{6}m_k.$$

R. 3

- a. On a par définition  $\forall k \in \llbracket 1, n \rrbracket$ ,  $g(x) = S_k(x)$ ,  $\forall x \in [x_{k-1}, x_k]$ . Le problème de définition de  $g$  provient du fait que  $g$  est définie deux fois en  $x_k$ ,  $k \in \llbracket 1, n-1 \rrbracket$ . En effet, on a

$$g(x_k) = S_k(x_k) \text{ et } g(x_k) = S_{k+1}(x_k).$$

Or par construction des  $S_k$ , on a  $S_k(x_k) = S_{k+1}(x_k) = f_k$  et donc la fonction  $g$  est bien définie sur  $[a, b]$ .

- b. Par construction, sur chaque intervalle  $[x_{k-1}; x_k]$ , la fonction  $g$  est polynomiale de degré inférieur ou égal à 3. Pour quelle soit un spline cubique, il reste à démontrer qu'elle est deux fois continûment différentiable sur  $[a, b]$ . Il suffit pour cela de vérifier qu'en chaque point  $x_k$ ,  $k \in \llbracket 1, n-1 \rrbracket$ , la fonction  $g$  est continue et admet des dérivées premières et secondes. La continuité est immédiate puisque  $g(x_k) = f_k$ . Pour les dérivées premières et secondes, il faut que leurs limites à gauche et à droite soient égales, c'est à dire

$$\forall k \in \llbracket 1, n-1 \rrbracket, S'_k(x_k) = S'_{k+1}(x_k) \text{ et } S''_k(x_k) = S''_{k+1}(x_k).$$

Par construction des  $S_k$ , la seconde équation est immédiate :  $S''_k(x_k) = S''_{k+1}(x_k) = m_k$ . On a  $S'_k(x) = -3a_k(x - x_k)^2 + 3b_k(x - x_{k-1})^2 - \alpha_k + \beta_k$  et donc

$$S'_k(x_k) = 3b_k h^2 - \alpha_k + \beta_k = \frac{h}{2} m_k + \frac{1}{h} (f_k - f_{k-1}) - \frac{h}{6} (m_k - m_{k-1})$$

De même, on obtient

$$S'_{k+1}(x_k) = -3a_{k+1} h^2 - \alpha_{k+1} + \beta_{k+1} = -\frac{h}{2} m_k + \frac{1}{h} (f_{k+1} - f_k) - \frac{h}{6} (m_{k+1} - m_k)$$

Donc  $g$  sera dérivable en  $x_k$  si  $S'_k(x_k) = S'_{k+1}(x_k)$ , c'est à dire si

$$\frac{h}{2} m_k + \frac{1}{h} (f_k - f_{k-1}) - \frac{h}{6} (m_k - m_{k-1}) = -\frac{h}{2} m_k + \frac{1}{h} (f_{k+1} - f_k) - \frac{h}{6} (m_{k+1} - m_k)$$

ce qui s'écrit encore,  $\forall k \in \llbracket 1, n-1 \rrbracket$ ,

$$m_{k+1} + 4m_k + m_{k-1} = \frac{6}{h^2} (f_{k+1} - 2f_k + f_{k-1})$$

R. 4

- a. La condition  $g''(a) = 0$  se traduit par  $S''_1(x_0) = 0$  or par (4.4c) avec  $k = 1$  on a  $S''_1(x_0) = m_0$  d'où  $m_0 = 0$ .

La condition  $g''(b) = 0$  se traduit par  $S''_n(x_n) = 0$  or par (4.4d) avec  $k = n$  on a  $S''_n(x_n) = m_n$  d'où  $m_n = 0$ .

Pour déterminer les  $m_k$ ,  $k \in \llbracket 0, n \rrbracket$ , on a  $n+1$  équations linéaires qui s'écrivent sous la forme matricielle  $\mathbb{A}\mathbf{M} = \mathbf{b}$  avec

$$\mathbb{A} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 4 & 1 \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix} \text{ et } \mathbf{b} = \frac{6}{h^2} \begin{pmatrix} 0 \\ f_0 - 2f_1 + f_2 \\ \vdots \\ f_{n-2} - 2f_{n-1} + f_n \\ 0 \end{pmatrix}$$

- b. La matrice  $\mathbb{A}$  est à diagonale strictement dominante : elle est donc inversible.

~~~~~ Fin correction ~~~~~



# Chapitre 5

## Intégration numérique

Soit  $f$  une fonction définie et intégrable sur un intervalle  $[a, b]$  donné. On propose de chercher des approximations de

$$I = \int_a^b f(x) dx$$

dans le cas où l'on ne connaît pas de primitive de  $f$ .

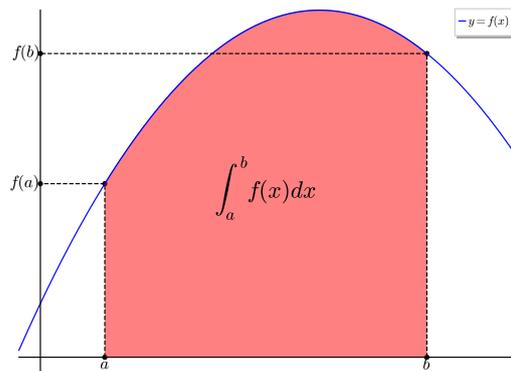


Figure 5.1: Représentation de  $\int_a^b f(x) dx$  (aire de la surface colorée)

**Définition 5.0.1.** Soient  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$  et  $\mathcal{Q}_n(f, a, b)$  la **formule de quadrature élémentaire** donnée par :

$$\mathcal{Q}_n(f, a, b) \stackrel{\text{def}}{=} (b - a) \sum_{j=0}^n w_j f(x_j) \quad (5.1)$$

avec  $\forall j \in \llbracket 0, n \rrbracket$   $w_j \in \mathbb{R}$  et  $x_j \in [a, b]$  distincts deux à deux. L'erreur associée à cette formule de

quadrature, notée  $\mathcal{E}_{a,b}(f)$ , est définie par

$$\mathcal{E}_{a,b}(f) = \int_a^b f(x)dx - \mathcal{Q}_n(f, a, b), \quad \forall f \in \mathcal{C}^0([a, b]; \mathbb{R}) \quad (5.2)$$

**Définition 5.0.2.** On dit qu'une formule d'intégration (ou formule de quadrature) est d'ordre  $p$  ou a pour **degré d'exactitude**  $p$  si elle est exacte pour les polynômes de degré inférieur ou égal à  $p$ .

## 5.1 Méthodes de quadrature élémentaires

On suppose que les points  $x_j$  de la formule de quadrature (5.1) sont deux à deux distincts.

### 5.1.1 Méthodes simplistes

On peut approcher  $f$  par un polynôme constant. Les trois formules usuelles sont

**Méthode du rectangle à gauche :** En figure 5.2, on représente l'approximation de  $\int_a^b f(x)dx$  lorsque  $f$  est approché par le polynôme constant  $P(x) = f(a)$ .

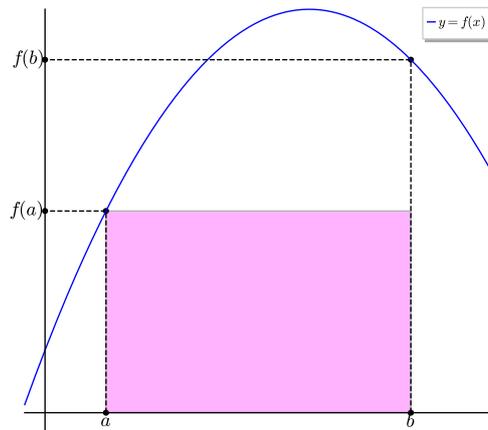


Figure 5.2: Formule du rectangle à gauche :  $\int_a^b f(x)dx \approx (b-a)f(a)$  (aire de la surface colorée)

On a alors

$$\int_a^b f(x)dx \approx \mathcal{Q}_0(f, a, b) = (b-a)f(a), \text{ formule du rectangle (à gauche)}$$

et son degré d'exactitude est 0.

**Méthode du rectangle à droite :** En figure 5.3, on représente l'approximation de  $\int_a^b f(x)dx$  lorsque  $f$  est approché par le polynôme constant  $P(x) = f(b)$ .

On a alors

$$\int_a^b f(x)dx \approx \mathcal{Q}_0(f, a, b) = (b-a)f(b), \text{ formule du rectangle (à droite)}$$

et son degré d'exactitude est 0.

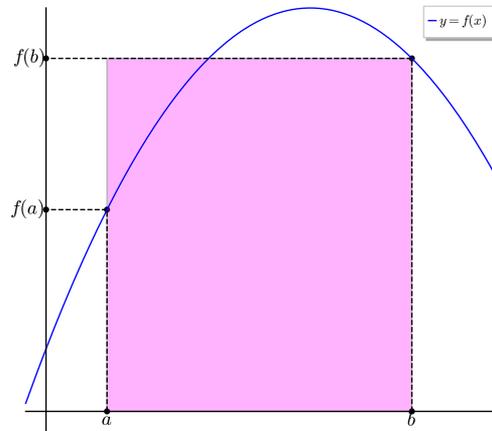


Figure 5.3: Formule du rectangle à droite :  $\int_a^b f(x)dx \approx (b-a)f(b)$  (aire de la surface colorée)

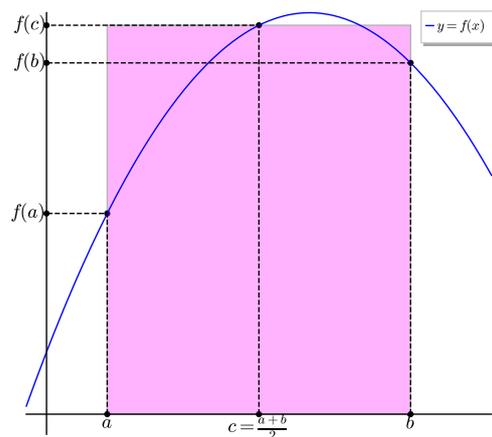


Figure 5.4: Formule du point milieu :  $\int_a^b f(x)dx \approx (b-a)f\left(\frac{a+b}{2}\right)$  (aire de la surface colorée)

**Méthode du point milieu :** En figure 5.4, on représente l'approximation de  $\int_a^b f(x)dx$  lorsque  $f$  est approché par le polynôme constant  $P(x) = f((a+b)/2)$ . On a alors

$$\int_a^b f(x)dx \approx \mathcal{Q}_0(f, a, b) = (b-a)f\left(\frac{a+b}{2}\right), \text{ formule du point milieu}$$

et son degré d'exactitude est 1.

La précision de ces formules est toute relative!

Nous allons maintenant voir comment *généraliser* en approchant la fonction  $f$  par des polynômes d'interpolation de Lagrange de degré plus élevés.

### 5.1.2 Quelques résultats théoriques

Pour obtenir certains résultats associés à une formule de quadrature sur l'intervalle  $[a; b]$ , il serait souvent plus simple de les déterminer sur les intervalles d'intégration  $[0; 1]$  ou  $[-1; 1]$ . La proposition suivante va nous permettre d'affirmer qu'étudier le degré d'exactitude d'une formule de quadrature sur l'intervalle  $[a; b]$  est équivalent à l'étudier sur l'intervalle  $[0; 1]$  ou sur l'intervalle  $[-1; 1]$  moyennant respectivement les changements de variables

$$x = \varphi(t) = a + (b-a)t \quad \text{ou} \quad x = \varphi(t) = \frac{a+b}{2} + \frac{b-a}{2}t$$

**Proposition 5.1.1.** Soit  $\mathcal{Q}_n(f, a, b)$  définie en (5.1), une formule de quadrature élémentaire à  $(n+1)$  points  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  (distincts deux à deux dans  $[a, b]$ ).

On note  $x = \varphi(t) = \alpha + \beta t$ ,  $\beta \in \mathbb{R}^*$ , le changement de variable affine,  $t_i = \varphi^{-1}(x_i)$ ,  $\forall i \in \llbracket 0, n \rrbracket$ , et

$$\mathcal{Q}_n(g, \varphi^{-1}(a), \varphi^{-1}(b)) = (\varphi^{-1}(b) - \varphi^{-1}(a)) \sum_{i=0}^n w_i g(t_i). \quad (5.3)$$

Alors  $\mathcal{Q}_n(f, a, b)$  est de degré d'exactitude  $k$  si et seulement si  $\mathcal{Q}_n(g, \varphi^{-1}(a), \varphi^{-1}(b))$  est de degré d'exactitude  $k$ .

*Preuve.* On a  $\varphi^{-1}(x) = \frac{x-\alpha}{\beta}$ .

$\Rightarrow$  On suppose que  $\mathcal{Q}_n(f, a, b)$  est de degré d'exactitude  $k$ . Soit  $Q \in \mathbb{R}_k[X]$ . On a

$$\int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} Q(t)dt = \frac{1}{\beta} \int_a^b Q \circ \varphi^{-1}(x)dx.$$

Or  $\varphi^{-1}$  est un polynôme de degré 1 et  $Q \circ \varphi^{-1}$  est la composé de deux polynômes: c'est donc un polynôme de degré le produit des degrés des deux polynômes, i.e.  $Q \circ \varphi^{-1} \in \mathbb{R}_k[X]$ . Comme  $\mathcal{Q}_n(f, a, b)$  est de degré d'exactitude  $k$ , on en déduit que

$$\int_a^b Q \circ \varphi^{-1}(x)dx = \mathcal{Q}_n(Q \circ \varphi^{-1}, a, b) = (b-a) \sum_{i=0}^n w_i Q \circ \varphi^{-1}(x_i) = (b-a) \sum_{i=0}^n w_i Q(t_i).$$

On a alors

$$\int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} Q(t)dt = \frac{b-a}{\beta} \sum_{i=0}^n w_i Q(t_i)$$

or

$$\varphi^{-1}(b) - \varphi^{-1}(a) = \frac{b-\alpha}{\beta} - \frac{a-\alpha}{\beta} = \frac{b-a}{\beta}.$$

On en conclut donc que  $\mathcal{Q}_n(g, \varphi^{-1}(a), \varphi^{-1}(b))$  est de degré d'exactitude  $k$ .

$\Leftarrow$  On suppose que  $\mathcal{Q}_n(g, \varphi^{-1}(a), \varphi^{-1}(b))$  est de degré d'exactitude  $k$ . Soit  $P \in \mathbb{R}_k[X]$ . On a

$$\int_a^b P(x)dx = \beta \int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} P \circ \varphi(t)dt.$$

Or  $\varphi$  est un polynôme de degré 1 et  $P \circ \varphi^{-1}$  est la composée de deux polynômes: c'est donc un polynôme de degré le produit des degrés des deux polynômes, i.e.  $P \circ \varphi \in \mathbb{R}_k[X]$ . Comme  $\mathcal{Q}_n(g, \varphi^{-1}(a), \varphi^{-1}(b))$  est de degré d'exactitude  $k$ , on en déduit que

$$\begin{aligned} \int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} P \circ \varphi(t) dt &= \mathcal{Q}_n(P \circ \varphi, \varphi^{-1}(a), \varphi^{-1}(b)) \\ &= (\varphi^{-1}(b) - \varphi^{-1}(a)) \sum_{i=0}^n w_i P \circ \varphi(t_i) \\ &= (\varphi^{-1}(b) - \varphi^{-1}(a)) \sum_{i=0}^n w_i P(x_i). \end{aligned}$$

On a alors

$$\int_a^b P(x) dx = \beta (\varphi^{-1}(b) - \varphi^{-1}(a)) \sum_{i=0}^n w_i P(x_i)$$

et comme  $\varphi^{-1}(b) - \varphi^{-1}(a) = \frac{b-a}{\beta}$ , on en déduit que  $\mathcal{Q}_n(f, a, b)$  est de degré d'exactitude  $k$ . □

**Proposition 5.1.2.** Soit  $\mathcal{Q}_n(f, a, b)$  définie en (5.1), une formule de quadrature élémentaire à  $n + 1$  points (distincts deux à deux dans  $[a, b]$ ). L'application  $f \mapsto \mathcal{Q}_n(f, a, b)$  définie de  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ , muni de la norme infini, à valeurs dans  $\mathbb{R}$  est linéaire continue.

*Preuve.* On commence par démontrer la linéarité. Soient  $f$  et  $g$  dans  $\mathcal{C}^0([a, b]; \mathbb{R})$ , et  $\lambda$  et  $\mu$  deux réels. Alors  $\lambda f + \mu g \in \mathcal{C}^0([a, b]; \mathbb{R})$ , et on a

$$\begin{aligned} \mathcal{Q}_n(\lambda f + \mu g, a, b) &= (b - a) \sum_{j=0}^n w_j (\lambda f + \mu g)(x_j) \\ &= (b - a) \sum_{j=0}^n w_j (\lambda f(x_j) + \mu g(x_j)) \\ &= \lambda (b - a) \sum_{j=0}^n w_j f(x_j) + \mu (b - a) \sum_{j=0}^n w_j g(x_j) \\ &= \lambda \mathcal{Q}_n(f, a, b) + \mu \mathcal{Q}_n(g, a, b). \end{aligned}$$

L'application  $f \mapsto \mathcal{Q}_n(f, a, b)$  est donc linéaire. Pour démontrer qu'elle est continue, il suffit alors de démontrer que

$$\exists C > 0, \text{ tel que } |\mathcal{Q}_n(f, a, b)| \leq C \|f\|_\infty, \quad \forall f \in \mathcal{C}^0([a, b]; \mathbb{R}).$$

Or, on a, pour tout  $f \in \mathcal{C}^0([a, b]; \mathbb{R})$ ,

$$\begin{aligned} |\mathcal{Q}_n(f, a, b)| &= |(b - a) \sum_{j=0}^n w_j f(x_j)| \\ &\leq (b - a) \sum_{j=0}^n |w_j| |f(x_j)| \\ &\leq C \|f\|_\infty, \quad \text{avec } C = (b - a) \sum_{j=0}^n |w_j| \text{ indépendant de } f. \end{aligned}$$

□

**Proposition 5.1.3.** La formule de quadrature élémentaire (5.1) à  $(n + 1)$  points, distincts deux

à deux, est de degré d'exactitude  $k$  (au moins) si et seulement si

$$(b-a) \sum_{i=0}^n w_i x_i^r = \frac{b^{r+1} - a^{r+1}}{r+1}, \quad \forall r \in \llbracket 0, k \rrbracket. \quad (5.4)$$

*Preuve.* •  $\Rightarrow$  Si la formule (5.1) est de degré d'exactitude  $k$ , elle est donc exacte pour tout polynôme de  $\mathbb{R}_k[X]$  et plus particulièrement pour tous les monômes  $1, X, X^2, \dots, X^k$ . Soit  $r \in \llbracket 0, k \rrbracket$ . En prenant  $f(x) = x^r$ , la formule (5.1) étant exacte par hypothèse, on obtient

$$\mathcal{Q}_n(x \mapsto x^r, a, b) = (b-a) \sum_{i=0}^n w_i x_i^r \stackrel{\text{hyp}}{=} \int_a^b x^r dx = \frac{b^{r+1} - a^{r+1}}{r+1}$$

•  $\Leftarrow$  On suppose que l'on a (5.4). Soit  $P \in \mathbb{R}_k[X]$ . On va montrer que la formule de quadrature (5.1) est alors exacte.

Le polynôme  $P$  peut s'écrire comme combinaison linéaire des monômes de  $\{1, X, X^2, \dots, X^k\}$ , base de  $\mathbb{R}_k[X]$ .

**1ère démonstration:** On a donc

$$P(x) = \sum_{j=0}^k \alpha_j x^j, \quad \forall x \in \mathbb{R}.$$

En prenant  $f = P$ , la formule de quadrature (5.1) donne

$$\int_a^b P(x) dx \approx (b-a) \sum_{i=0}^n w_i P(x_i) = (b-a) \sum_{i=0}^n w_i \sum_{j=0}^k \alpha_j x_i^j$$

De plus, par linéarité de l'intégrale, on a

$$\int_a^b P(x) dx = \sum_{j=0}^k \alpha_j \int_a^b x^j dx = \sum_{j=0}^k \alpha_j \frac{b^{j+1} - a^{j+1}}{j+1}$$

et en utilisant (5.4) on obtient

$$\int_a^b P(x) dx = (b-a) \sum_{j=0}^k \alpha_j \sum_{i=0}^n w_i x_i^j$$

Ce qui donne

$$\int_a^b P(x) dx = (b-a) \sum_{i=0}^n w_i P(x_i).$$

La formule de quadrature est donc de degré d'exactitude  $k$ .

**2ème démonstration:** On a donc

$$P = \sum_{j=0}^k \alpha_j X^j$$

et par linéarité de l'application  $f \mapsto \mathcal{Q}_n(f, a, b)$  (voir Proposition 5.1.2) on obtient

$$\begin{aligned} \mathcal{Q}_n(P, a, b) &= \mathcal{Q}_n\left(\sum_{j=0}^k \alpha_j X^j, a, b\right) \\ &= \sum_{j=0}^k \alpha_j \mathcal{Q}_n(X^j, a, b). \end{aligned}$$

Par hypothèse, on a (5.4) et, comme par définition  $X^j$  est le polynôme  $x \mapsto x^j$ , on obtient

$$\forall j \in \llbracket 0, k \rrbracket, \quad \mathcal{Q}_n(X^j, a, b) \stackrel{\text{hyp}}{=} \int_a^b X^j(x) dx = \int_a^b x^j dx = \frac{b^{j+1} - a^{j+1}}{j+1}.$$

De plus, par linéarité de l'intégrale, on a

$$\int_a^b P(x)dx = \sum_{j=0}^k \alpha_j \int_a^b x^j dx = \sum_{j=0}^k \alpha_j \frac{b^{j+1} - a^{j+1}}{j+1}$$

et donc

$$\int_a^b P(x)dx = \sum_{j=0}^k \alpha_j \mathcal{Q}_n(X^j, a, b) = \mathcal{Q}_n(P, a, b).$$

□

En appliquant la Proposition 5.1.3 avec  $k = 0$ , on obtient immédiatement le corollaire suivant.

**Corollaire 5.1.1.** *La formule de quadrature élémentaire (5.1) à  $(n + 1)$  points est de degré d'exactitude 0 au moins si et seulement si*

$$\sum_{i=0}^n w_i = 1.$$

**Proposition 5.1.4.** *Soient  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  des points deux à deux distincts de l'intervalle  $[a, b]$  donnés. Il existe alors une unique formule de quadrature élémentaire (5.1) à  $(n + 1)$  points de degré d'exactitude  $n$  au moins.*

*Preuve.* En fixant les points  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  deux à deux distincts, pour obtenir explicitement la formule de quadrature de type (5.1) il faut déterminer les  $n + 1$  poids  $(w_i)_{i \in \llbracket 0, n \rrbracket}$ . Or, de (5.4), en prenant  $k = n$ , on obtient exactement  $n + 1$  équations linéaires en les  $(w_i)$  s'écrivant matriciellement sous la forme :

$$(b-a) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} b-a \\ \frac{b^2-a^2}{2} \\ \vdots \\ \frac{b^{n+1}-a^{n+1}}{n+1} \end{pmatrix}$$

La matrice intervenant dans le système précédent s'appelle **la matrice de Vandermonde** et elle est inversible (car les  $(x_i)$  sont deux à deux distincts, voir Exercice ??). Ceci établit donc l'existence et l'unicité de poids  $(w_i)_{i \in \llbracket 0, n \rrbracket}$  tels que la formule de quadrature élémentaire (5.1) soit d'ordre (au moins)  $n$ .

Il est aussi possible de démontrer l'unicité classiquement. Supposons qu'il existe  $(w_i)_{i \in \llbracket 0, n \rrbracket}$  et  $(\tilde{w}_i)_{i \in \llbracket 0, n \rrbracket}$  tels que pour tout  $P \in \mathbb{R}_n[X]$ , on ait

$$\int_a^b P(x)dx = (b-a) \sum_{i=0}^n w_i P(x_i) = (b-a) \sum_{i=0}^n \tilde{w}_i P(x_i).$$

On a alors  $\forall P \in \mathbb{R}_n[X]$ ,

$$\sum_{i=0}^n (w_i - \tilde{w}_i) P(x_i) = 0. \quad (5.5)$$

On rappelle que les fonctions de base de Lagrange associées aux  $(n + 1)$  points  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  définies en (4.3), notées  $L_i$ , sont dans  $\mathbb{R}_n[X]$  et vérifient

$$L_i(x_j) = \delta_{i,j}, \quad \forall j \in \llbracket 0, n \rrbracket$$

Soit  $j \in \llbracket 0, n \rrbracket$ . En choisissant  $P = L_j$  dans (5.5), on obtient

$$0 = \sum_{i=0}^n (w_i - \tilde{w}_i) L_j(x_i) = (w_j - \tilde{w}_j)$$

ce qui prouve l'unicité. □

**EXERCICE 5.1.1**

Soient  $(x_i)_{i=0}^n$   $(n + 1)$  points donnés et distincts 2 à 2 d'un intervalle  $[a, b]$  ( $a < b$ ). Ecrire une fonction algorithmique `WeightsFromPoints` permettant de déterminer les poids  $(w_i)_{i=0}^n$  de telle sorte que la formule de quadrature élémentaire associée soit de degré d'exactitude  $n$  au moins en s'inspirant de résultats obtenus dans la démonstration de la Proposition 5.1.4. On pourra utiliser la fonction algorithmique  $\mathbf{x} \leftarrow \text{Solve}(\mathbf{A}, \mathbf{b})$  permettant de résoudre le système linéaire  $\mathbf{A}\mathbf{x} = \mathbf{b}$ .

**Correction** Nous avons vu, dans la Proposition 5.1.4, que pour avoir une formule de quadrature élémentaire de degré d'exactitude  $n$ , il est nécessaire et suffisant que les  $(n + 1)$  poids  $(w_i)_{i=0}^n$  soient solution du système linéaire suivant:

$$(b - a) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} b - a \\ \frac{b^2 - a^2}{2} \\ \vdots \\ \frac{b^{n+1} - a^{n+1}}{n+1} \end{pmatrix}$$

**Algorithme 5.1** Fonction `WeightsFromPoints` retournant le tableau des poids  $\mathbf{w}$  associé à un tableau de points  $\mathbf{x}$  donnés (points 2 à 2 distincts) appartenant à un intervalle  $[a, b]$ .

**Données :**  $\mathbf{x}$  : tableau de  $\mathbb{R}^{n+1}$  contenant  $(n + 1)$  points distincts deux à deux dans un intervalle  $[a, b]$  avec la convention  $\mathbf{x}(i) = x_{i-1}, \forall i \in \llbracket 1, n + 1 \rrbracket$   
 $a, b$  : deux réels,  $a < b$ .

**Résultat :**  $\mathbf{w}$  : vecteur de  $\mathbb{R}^{n+1}$  avec  $\mathbf{w}(i) = w_{i-1}, \forall i \in \llbracket 1, n + 1 \rrbracket$

```

1: Fonction  $\mathbf{w} \leftarrow \text{WeightsFromPoints}(\mathbf{x}, a, b)$ 
2:    $\mathbf{B} \leftarrow \mathbf{O}_{n+1}$ 
3:    $\mathbf{A} \leftarrow \mathbf{O}_{n+1, n+1}$ 
4:   Pour  $i \leftarrow 1$  à  $n + 1$  faire
5:     Pour  $j \leftarrow 1$  à  $n + 1$  faire
6:        $\mathbf{A}(i, j) \leftarrow \mathbf{x}(j)^{(i-1)}$ 
7:     Fin
8:      $\mathbf{B}(i) \leftarrow (b^i - a^i) / (i * (b - a))$ 
9:   Fin
10:   $\mathbf{w} \leftarrow \text{Solve}(\mathbf{A}, \mathbf{B})$ 
11: Fin

```

~~~~~ Fin correction ~~~~~

**Proposition 5.1.5** (symétrie). Soit  $\mathcal{Q}_n(f, a, b)$  définie en (5.1), une formule de quadrature élémentaire à  $(n + 1)$  points (distincts deux à deux). On dit qu'elle est **symétrique** si

$$\forall i \in \llbracket 0, n \rrbracket, \quad \frac{x_i + x_{n-i}}{2} = \frac{a + b}{2} \quad \text{et} \quad w_i = w_{n-i}. \quad (5.6)$$

Dans ce cas si cette formule est exacte pour les polynômes de degré  $2m$  alors elle est nécessairement exacte pour les polynômes de degré  $2m + 1$ .

*Preuve.* Soit  $P \in \mathbb{R}_{2m+1}[X]$ . Il peut alors s'écrire sous la forme

$$P(x) = C \left( x - \frac{a + b}{2} \right)^{2m+1} + R(x)$$

avec  $C$  une constante réelle et  $R \in \mathbb{R}_{2m}[X]$ . On a alors

$$\int_a^b P(x) dx = C \int_a^b \left( x - \frac{a + b}{2} \right)^{2m+1} dx + \int_a^b R(x) dx$$

et en appliquant la formule de quadrature au polynôme  $P$  on obtient

$$\sum_{i=0}^n w_i P(x_i) = C \sum_{i=0}^n w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} + \sum_{i=0}^n w_i R(x_i)$$

On veut donc démontrer que

$$\int_a^b P(x) dx = (b-a) \sum_{i=0}^n w_i P(x_i)$$

c'est à dire

$$C \int_a^b \left(x - \frac{a+b}{2}\right)^{2m+1} dx + \int_a^b R(x) dx = (b-a) C \sum_{i=0}^n w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} + (b-a) \sum_{i=0}^n w_i R(x_i)$$

Comme la formule de quadrature est supposée exacte pour les polynôme de degré  $2m$ , on a

$$\int_a^b R(x) dx = (b-a) \sum_{i=0}^n w_i R(x_i).$$

Il reste donc à démontrer que

$$\int_a^b \left(x - \frac{a+b}{2}\right)^{2m+1} dx = (b-a) \sum_{i=0}^n w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1}.$$

Or en effectuant le changement de variable  $t \mapsto \frac{a+b}{2} + t \frac{b-a}{2}$  on obtient

$$\int_a^b \left(x - \frac{a+b}{2}\right)^{2m+1} dx = \frac{b-a}{2} \int_{-1}^1 t^{2m+1} dt = 0.$$

Des propriétés de symétrie de la formule, on déduit

$$x_i + x_{n-i} = a+b \Leftrightarrow x_i - \frac{a+b}{2} = -\left(x_{n-i} - \frac{a+b}{2}\right)$$

- Si  $n = 2k$ , ( $n$  paire), on a alors un nombre **impair** de points avec nécessairement  $x_k = x_{n-k} = \frac{a+b}{2}$  et

$$\begin{aligned} \sum_{i=0}^n w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} + 0 \times w_k + \sum_{i=k+1}^{2k} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} \\ &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} - \sum_{i=k+1}^{2k} w_{n-i} \left(x_{n-i} - \frac{a+b}{2}\right)^{2m+1} \\ &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} - \sum_{j=0}^{k-1} w_j \left(x_j - \frac{a+b}{2}\right)^{2m+1} \\ &= 0. \end{aligned}$$

- Si  $n = 2k - 1$ , ( $n$  impaire), on a alors un nombre **pair** de points (avec  $x_i \neq \frac{a+b}{2}, \forall i \in \llbracket 0, n \rrbracket$ ) et

$$\begin{aligned} \sum_{i=0}^n w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} + \sum_{i=k}^{2k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} \\ &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} - \sum_{i=k}^{2k-1} w_{n-i} \left(x_{n-i} - \frac{a+b}{2}\right)^{2m+1} \\ &= \sum_{i=0}^{k-1} w_i \left(x_i - \frac{a+b}{2}\right)^{2m+1} - \sum_{j=0}^{k-1} w_j \left(x_j - \frac{a+b}{2}\right)^{2m+1} \\ &= 0. \end{aligned}$$

□

Le résultat suivant fait le lien avec les polynômes d'interpolation de Lagrange pour établir une majoration de l'erreur associée à une formule de quadrature élémentaire.

**Proposition 5.1.6.** Soit  $\mathcal{Q}_n(f, a, b)$  définie en (5.1), une formule de quadrature élémentaire à  $(n+1)$  points  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  (distincts deux à deux).

La formule de quadrature est de degré d'exactitude  $n$  au moins si et seulement si pour tout  $i \in \llbracket 0, n \rrbracket$ , les poids  $w_i$  sont donnés par

$$w_i = \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} dx = \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-t_j}{t_i-t_j} dt, \quad \forall i \in \llbracket 0, n \rrbracket \quad (5.7)$$

avec  $t_i = (x_i - a)/(b - a)$ .

Si  $f \in \mathcal{C}^{n+1}([a, b]; \mathbb{R})$  alors on a

$$\left| \int_a^b f(x) dx - \mathcal{Q}_n(f, a, b) \right| \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \int_a^b \left| \prod_{i=0}^n (x - x_i) \right| dx \quad (5.8)$$

*Preuve.* Montrons tout d'abord que l'on a l'égalité suivante

$$\frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} dx = \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-t_j}{t_i-t_j} dt.$$

Par le changement de variables  $s : t \rightarrow a + (b-a)t$  on obtient

$$\int_a^b L_i(x) dx = (b-a) \int_0^1 L_i \circ s(t) dt$$

et l'on a  $x_i = s(t_i) = a + (b-a)t_i$  où  $t_i = (x_i - a)/(b - a)$ . On en déduit

$$\begin{aligned} \int_0^1 L_i \circ s(t) dt &= \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{s(t) - s(t_j)}{s(t_i) - s(t_j)} dt = \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(b-a)(t-t_j)}{(b-a)(t_i-t_j)} dt \\ &= \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-t_j}{t_i-t_j} dt \end{aligned}$$

L'égalité est donc démontré.

Pour démontrer la proposition, on note  $\mathcal{L}_n(f)$  le polynôme d'interpolation de Lagrange associés aux points  $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$  :

$$\mathcal{L}_n(f)(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad \text{avec} \quad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

On a alors

$$\int_a^b \mathcal{L}_n(f)(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx.$$

$\Rightarrow$  On suppose que la formule de quadrature est de degré d'exactitude  $n$ . Soit  $i \in \llbracket 0, n \rrbracket$ ,  $L_i \in \mathbb{R}_n[X]$  et on a alors par hypothèse

$$\mathcal{Q}_n(L_i, a, b) \stackrel{\text{hyp}}{=} \int_a^b L_i(x) dx.$$

Or comme  $L_i(x_j) = \delta_{i,j}$  on a

$$\mathcal{Q}_n(L_i, a, b) = (b-a) \sum_{j=0}^n w_j L_i(x_j) = (b-a) w_i.$$

On obtient donc

$$w_i = \frac{1}{b-a} \int_a^b L_i(x) dx.$$

⇐ On suppose les poids  $(w_i)_{i=0}^n$  donnés par (5.7). La formule de quadrature s'écrit alors

$$\mathcal{Q}_n(f, a, b) \stackrel{\text{hyp}}{=} \sum_{i=0}^n f(x_i) \int_a^b L_i(x) dx.$$

Soit  $P \in \mathbb{R}_n[X]$ . Par unicité du polynôme d'interpolation de Lagrange, on a  $P = \mathcal{L}_n(P)$  et

$$\begin{aligned} \int_a^b P(x) dx &= \int_a^b \mathcal{L}_n(P)(x) dx \\ &= \int_a^b \sum_{i=0}^n L_i(x) P(x_i) dx \\ &= \sum_{i=0}^n P(x_i) \int_a^b L_i(x) dx \\ &= (b-a) \sum_{i=0}^n w_i P(x_i) = \mathcal{Q}_n(P, a, b). \end{aligned}$$

La formule de quadrature est donc exacte pour tout les polynômes de degré  $n$  au moins.

Pour démontrer l'inégalité (5.8), comme  $f \in \mathcal{C}^{n+1}([a, b]; \mathbb{R})$ , on peut appliquer le théorème 4.1.2 pour obtenir

$$\forall x \in [a, b], \exists \xi_x \in [a, b], f(x) - \mathcal{L}_n(f)(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

On en déduit

$$\begin{aligned} |f(x) - \mathcal{L}_n(f)(x)| &= \left| \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \pi_n(x) \right| \\ &\leq \frac{\|f^{(n+1)}\|_{\infty}}{(n+1)!} |\pi_n(x)| \end{aligned}$$

L'application  $f - \mathcal{L}_n(f)$  étant intégrable sur  $[a, b]$ , l'application  $|f - \mathcal{L}_n(f)|$  l'est aussi. De même  $|\pi_n(x)|$  est intégrable sur  $[a, b]$ . On obtient alors

$$\int_a^b |f(x) - \mathcal{L}_n(f)(x)| dx \leq \frac{\|f^{(n+1)}\|_{\infty}}{(n+1)!} \int_a^b |\pi_n(x)| dx.$$

De plus

$$\left| \int_a^b f(x) - \mathcal{L}_n(f)(x) dx \right| \leq \int_a^b |f(x) - \mathcal{L}_n(f)(x)| dx$$

ce qui donne

$$\left| \int_a^b f(x) dx - \int_a^b \mathcal{L}_n(f)(x) dx \right| \leq \frac{\|f^{(n+1)}\|_{\infty}}{(n+1)!} \int_a^b \left| \prod_{i=0}^n (x - x_i) \right| dx.$$

La formule de quadrature est de degré d'exactitude  $n$  au moins et le polynôme d'interpolation de Lagrange  $\mathcal{L}_n(f)$  est de degré  $n$  donc on a

$$\begin{aligned} \int_a^b \mathcal{L}_n(f)(x) dx &= \mathcal{Q}_n(\mathcal{L}_n(f), a, b) \\ &= (b-a) \sum_{i=0}^n w_i \mathcal{L}_n(f)(x_i) \\ &= (b-a) \sum_{i=0}^n w_i f(x_i) \text{ car } \mathcal{L}_n(f)(x_i) = f(x_i) \\ &= \mathcal{Q}_n(f, a, b) \end{aligned}$$

ce qui donne

$$\begin{aligned} |\mathcal{E}_{a,b}(f)| &= \left| \int_a^b f(x) dx - \mathcal{Q}_n(f, a, b) \right| \\ &= \left| \int_a^b f(x) dx - \int_a^b \mathcal{L}_n(f)(x) dx \right| \\ &\leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \int_a^b \left| \prod_{i=0}^n (x - x_i) \right| dx. \end{aligned}$$

□

### EXERCICE 5.1.2

Soient  $(x_i)_{i=0}^n$  des points distincts 2 à 2 de l'intervalle  $[a, b]$  vérifiant

$$\forall i \in \llbracket 0, n \rrbracket, \quad \frac{x_i + x_{n-i}}{2} = \frac{a+b}{2}.$$

On note  $L_i \in \mathbb{R}_n[X]$ ,  $i \in \llbracket 0, n \rrbracket$  les  $(n+1)$  polynômes de base de Lagrange définis par

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

et vérifiant  $L_i(x_j) = \delta_{i,j}$ ,  $\forall (i, j) \in \llbracket 0, n \rrbracket^2$

**Q. 1**

Soit  $i \in \llbracket 0, n \rrbracket$ . Montrer que

$$\forall x \in \mathbb{R}, \quad L_i((a+b) - x) = L_{n-i}(x).$$

**Q. 2**

Soient  $(w_i)_{i=0}^n$  définis par

$$w_i = \frac{1}{b-a} \int_a^b L_i(t) dt, \quad \forall i \in \llbracket 0, n \rrbracket$$

Montrer que l'on a alors

$$\forall i \in \llbracket 0, n \rrbracket, \quad w_i = w_{n-i}$$

### Correction

**R. 1**

Soit  $i \in \llbracket 0, n \rrbracket$ . On note  $\varphi(x) = (a+b) - x$  le polynôme de degré 1 et  $P = L_i \circ \varphi$  le polynôme de  $\mathbb{R}_n[X]$  (la composé de 2 polynômes est de degré le produit des degrés des 2 polynômes).

On a

$$\forall j \in \llbracket 0, n \rrbracket, \quad x_{n-j} = (a+b) - x_j$$

et

$$P(x_j) = L_i((a+b) - x_j) = L_i(x_{n-j}) = \delta_{i, n-j} = \begin{cases} 1, & \text{si } i = n-j \\ 0, & \text{sinon} \end{cases} = \begin{cases} 1, & \text{si } j = n-i \\ 0, & \text{sinon} \end{cases} = \delta_{n-i, j}.$$

C'est à dire

$$\forall j \in \llbracket 0, n \rrbracket, \quad P(x_j) = \delta_{n-i, j}.$$

Or  $L_{n-i}$  est l'unique polynôme de  $\mathbb{R}_n[X]$  vérifiant la relation précédente dont  $P = L_{n-i}$  (voir Exercice 4.1.1, page 136).

**R. 2**

Soit  $i \in \llbracket 0, n \rrbracket$ . On note  $t = \varphi(x) = (a+b) - x$  le changement de variable affine. On a alors

$\varphi^{-1}(t) = (a + b) - t$  et

$$\begin{aligned} w_i &= \frac{1}{b-a} \int_a^b L_i(t) dt \\ &= \frac{1}{b-a} \int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} L_i \circ \varphi(x) \varphi'(x) dx \\ &= \frac{1}{b-a} \int_b^a L_i((a+b)-x) (-1) dx \\ &= \frac{1}{b-a} \int_a^b L_i((a+b)-x) dx \\ &= \frac{1}{b-a} \int_a^b L_{n-i}(x) dx \quad \text{d'après la question précédente} \\ &= w_{n-i}. \end{aligned}$$

Fin correction

De l'Exercice 5.1.2 et de la Proposition 5.1.5, on obtient le lemme suivant

**Lemme 5.1.1.** Soient  $(x_i)_{i=0}^n$  des points distincts 2 à 2 de l'intervalle  $[a, b]$  vérifiant

$$\forall i \in \llbracket 0, n \rrbracket, \quad \frac{x_i + x_{n-i}}{2} = \frac{a+b}{2}.$$

Soient  $(w_i)_{i=0}^n$  définis par

$$w_i = \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx, \quad \forall i \in \llbracket 0, n \rrbracket$$

On a alors

$$\forall i \in \llbracket 0, n \rrbracket, \quad w_i = w_{n-i}$$

et la formule de quadrature élémentaire associée est de degré d'exactitude au moins  $n$  si  $n$  est impaire et au moins  $n+1$  sinon.

**Proposition 5.1.7** (Degré maximal d'exactitude). Soit  $\mathcal{Q}_n(f, a, b)$  défini par (5.1) une formule de quadrature élémentaire de degré d'exactitude au moins  $n$ . Elle est alors de degré d'exactitude  $n+m$ ,  $m \in \mathbb{N}^*$ , au moins si et seulement si

$$\int_a^b \pi_n(x) Q(x) dx = 0, \quad \forall Q \in \mathbb{R}_{m-1}[X] \quad (5.9)$$

où  $\pi_n$  est le polynôme de degré  $n+1$  défini par

$$\pi_n(x) = \prod_{i=0}^n (x - x_i). \quad (5.10)$$

Le degré maximal d'exactitude d'une formule de quadrature élémentaire à  $n+1$  points est  $2n+1$ . De plus, on a

$$(5.9) \iff \int_a^b \pi_n(x) x^k dx = 0, \quad \forall k \in \llbracket 0, m-1 \rrbracket. \quad (5.11)$$

*Preuve.* On a par définition

$$\mathcal{Q}_n(f, a, b) = (b-a) \sum_{j=1}^n w_j f(x_j)$$

où les  $n+1$  points  $x_j$  sont distincts deux à deux dans l'intervalle  $[a, b]$ .

La formule de quadrature est exacte pour les polynômes de degré  $n + m$  si et seulement si

$$\int_a^b P(x)dx = \mathcal{Q}_n(P, a, b), \quad \forall P \in \mathbb{R}_{n+m}[X]$$

Soit  $P \in \mathbb{R}_{n+m}[X]$ , on peut effectuer la division euclidienne de  $P$  par  $\pi_n \in \mathbb{R}_{n+1}[X]$ . Il existe donc  $Q \in \mathbb{R}_{m-1}[X]$  (quotient) et  $R \in \mathbb{R}_n[X]$  (reste) tels que

$$P = Q\pi_n + R.$$

On a alors par linéarité de l'intégrale

$$\int_a^b P(x)dx = \int_a^b Q(x)\pi_n(x)dx + \int_a^b R(x)dx$$

et par linéarité de  $\mathcal{Q}_n$

$$\mathcal{Q}_n(P, a, b) = \mathcal{Q}_n(Q\pi_n, a, b) + \mathcal{Q}_n(R, a, b).$$

Par hypothèse, la formule de quadrature a pour degré d'exactitude  $n$  et comme  $R \in \mathbb{R}_n[X]$  on obtient

$$\int_a^b R(x)dx = \mathcal{Q}_n(R, a, b).$$

On en déduit alors que

$$\int_a^b P(x)dx - \mathcal{Q}_n(P, a, b) = \int_a^b Q(x)\pi_n(x)dx - \mathcal{Q}_n(Q\pi_n, a, b).$$

Par construction  $\pi_n(x_j) = 0, \forall j \in \llbracket 0, n \rrbracket$ , ce qui donne

$$\mathcal{Q}_n(Q\pi_n, a, b) = (b-a) \sum_{j=0}^n w_j Q(x_j)\pi_n(x_j) = 0$$

et donc

$$\int_a^b P(x)dx - \mathcal{Q}_n(P, a, b) = \int_a^b Q(x)\pi_n(x)dx. \quad (5.12)$$

⇐ si (5.9) est vérifié alors,

$$\int_a^b P(x)dx - \mathcal{Q}_n(P, a, b) = 0.$$

La formule de quadrature est donc de degré d'exactitude  $n + m$ .

⇒ si la formule de quadrature est de degré d'exactitude  $n + m$  alors pour tout  $Q \in \mathbb{R}_{m-1}[X]$ , le polynôme  $P = Q\pi_n \in \mathbb{R}_{n+m}[X]$  et donc

$$\int_a^b P(x)dx - \mathcal{Q}_n(P, a, b) = 0.$$

En utilisant (5.12), on obtient alors

$$\int_a^b Q(x)\pi_n(x)dx = 0.$$

- La plus grande valeur que puisse prendre  $m$  est  $n + 1$  (i.e.  $m + n = 2n + 1$ ). En effet si  $m = n + 2$  alors  $Q \in \mathbb{R}_{n+1}[X]$  dans (5.9). Or  $\pi_n \in \mathbb{R}_{n+1}[X]$  et en prenant  $Q = \pi_n$  on obtient

$$\int_a^b \pi_n(x)Q(x)dx = \int_a^b Q^2(x)dx > 0.$$

- Dans l'équivalence (5.11),  $\boxed{\Rightarrow}$  est immédiat car  $x \mapsto x^k \in \mathbb{R}_{m-1}[X]$ .  
Pour établir l'implication  $\boxed{\Leftarrow}$ , on utilise la linéarité de l'intégrale. En effet, soit  $P \in \mathbb{R}_{m-1}[X]$ , il existe  $(\alpha_0, \dots, \alpha_{m-1}) \in \mathbb{R}^m$  tel que

$$\forall x \in \mathbb{R}, \quad P(x) = \sum_{k=0}^{m-1} \alpha_k x^k.$$

On a alors

$$\begin{aligned} \int_a^b P(x) \pi_n(x) dx &= \int_a^b \pi_n(x) \sum_{k=0}^{m-1} \alpha_k x^k dx \\ &= \sum_{k=0}^{m-1} \alpha_k \int_a^b \pi_n(x) x^k dx \\ &\stackrel{\text{hyp}}{=} \sum_{k=0}^{m-1} \alpha_k \times 0 = 0. \end{aligned}$$

□

Cette proposition sera utilisée pour déterminer les points de quadrature de la méthode de Gauss-Legendre. Une fois connu les points de quadrature  $x_i$ , les poids  $w_i$  pourront être calculés par (5.7).

Si l'on choisit comme points de quadrature les  $n + 1$  points de la discrétisation régulière de l'intervalle  $[a, b]$ , on obtient la méthode de Newton-Cotes.

Bien d'autres méthodes peuvent être obtenues (avec d'autres points), certaines permettant le calcul d'intégrales avec poids de la forme  $\int_a^b w(x) f(x) dx$  :

- méthode de Newton-Cotes ouvertes,
- méthode de Gauss-Legendre,
- méthode de Gauss-Jacobi,
- méthode de Gauss-Tchebychev,
- méthode de Gauss-Laguerre,
- méthode de Gauss-Lobatto,
- méthode de Romberg...

### 5.1.3 Formules élémentaires de Newton-Cotes

Soit  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  une discrétisation régulière de l'intervalle  $[a, b]$ :

$$\forall i \in \llbracket 0, n \rrbracket, \quad x_i = a + ih \text{ avec } h = (b - a)/n.$$

On a alors

$$\forall i \in \llbracket 0, n \rrbracket, \quad \frac{x_i + x_{n-i}}{2} = \frac{a + b}{2}.$$

**Proposition 5.1.8.** Soient  $f \in C^0([a, b]; \mathbb{R})$  et  $(x_i)_{i \in \llbracket 0, n \rrbracket}$  une discrétisation régulière de l'intervalle  $[a, b]$ :  $x_i = a + ih$  avec  $h = (b - a)/n$ .

Les formules de quadrature élémentaires de Newton-Cotes s'écrivent sous la forme

$$\int_a^b f(x) dx \approx (b - a) \sum_{i=0}^n w_i f(x_i)$$

où les poids  $(w_i)_{i=0}^n$  sont donnés par (5.7).

Elles sont symétriques et leur degré d'exactitude (d.e. dans le tableau suivant) est égal à  $n$  si  $n$  est impair et à  $n + 1$  sinon.

| $n$ | d.e. | $w_i$ (poids)       |                      |                      |                      |                      |                      |                      |                      |                     | nom        |
|-----|------|---------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|---------------------|------------|
| 1   | 1    | $\frac{1}{2}$       | $\frac{1}{2}$        |                      |                      |                      |                      |                      |                      |                     | trapèze    |
| 2   | 3    | $\frac{1}{6}$       | $\frac{2}{3}$        | $\frac{1}{6}$        |                      |                      |                      |                      |                      |                     | Simpson    |
| 3   | 3    | $\frac{1}{8}$       | $\frac{3}{8}$        | $\frac{3}{8}$        | $\frac{1}{8}$        |                      |                      |                      |                      |                     | Newton     |
| 4   | 5    | $\frac{7}{90}$      | $\frac{16}{45}$      | $\frac{2}{15}$       | $\frac{16}{45}$      | $\frac{7}{90}$       |                      |                      |                      |                     | Villarceau |
| 5   | 5    | $\frac{19}{288}$    | $\frac{25}{96}$      | $\frac{25}{144}$     | $\frac{25}{144}$     | $\frac{25}{96}$      | $\frac{19}{288}$     |                      |                      |                     | ?          |
| 6   | 7    | $\frac{41}{840}$    | $\frac{9}{35}$       | $\frac{9}{280}$      | $\frac{34}{105}$     | $\frac{9}{280}$      | $\frac{9}{35}$       | $\frac{41}{840}$     |                      |                     | Weddle     |
| 7   | 7    | $\frac{751}{17280}$ | $\frac{3577}{17280}$ | $\frac{49}{640}$     | $\frac{2989}{17280}$ | $\frac{2989}{17280}$ | $\frac{49}{640}$     | $\frac{3577}{17280}$ | $\frac{751}{17280}$  |                     | ?          |
| 8   | 9    | $\frac{989}{28350}$ | $\frac{2944}{14175}$ | $-\frac{464}{14175}$ | $\frac{5248}{14175}$ | $-\frac{454}{2835}$  | $\frac{5248}{14175}$ | $-\frac{464}{14175}$ | $\frac{2944}{14175}$ | $\frac{989}{28350}$ | ?          |

Table 5.1: Méthodes de Newton-Cotes

Par exemple, la formule de Simpson ( $n = 2$ ) est

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \quad (5.13)$$

Pour un  $n$  donné, déterminer les coefficients  $w_i$  de la formule de Newton-Cotes est assez simple.

**Démonstration 1:** En effet, il suffit de remarquer que la formule doit être exacte si  $f$  est un polynôme de degré au plus  $n$ . Ensuite par linéarité de la formule, on est ramené à résoudre un système linéaire à  $n+1$  inconnues (les  $(w_i)_{i \in \llbracket 0, n \rrbracket}$ ) en écrivant que pour chaque monôme

$$(b-a) \sum_{i=0}^n w_i f(x_i) = \int_a^b f(x)dx, \quad \forall f \in \{1, X, X^2, \dots, X^n\} \subset \mathbb{R}_n[X]. \quad (5.14)$$

Par exemple, pour  $n = 2$ , on a  $b = a + 2h$ . A partir de (5.14) on obtient les trois équations :

$$\begin{cases} w_0 + w_1 + w_2 & = 1, & (f(x) = 1) \\ aw_0 + (a+h)w_1 + (a+2h)w_2 & = \frac{1}{b-a} \int_a^b x dx = a+h, & (f(x) = x) \\ a^2w_0 + (a+h)^2w_1 + (a+2h)^2w_2 & = \frac{1}{b-a} \int_a^b x^2 dx = \frac{1}{3}(3a^2 + 6ah + 4h^2), & (f(x) = x^2). \end{cases}$$

On a alors

$$\begin{cases} w_0 = -w_1 - w_2 + 1, \\ a(-w_1 - w_2 + 1) + (a+h)w_1 + (a+2h)w_2 = a+h, \\ a^2(-w_1 - w_2 + 1) + (a+h)^2w_1 + (a+2h)^2w_2 = a^2 + 2ah + \frac{4}{3}h^2. \end{cases}$$

c'est à dire

$$\begin{cases} w_0 = -w_1 - w_2 + 1, \\ w_1 + 2w_2 = 1, \\ 2a(w_1 + 2w_2) + h(w_1 + 4w_2) = 2a + 4h/3, \end{cases}$$

Par substitution, de la deuxième équation dans la troisième, on obtient enfin

$$\begin{cases} w_0 = -w_1 - w_2 + 1, \\ w_1 + 2w_2 = 1, \\ w_1 + 4w_2 = 4/3, \end{cases}$$

ce qui donne  $w_0 = w_2 = \frac{1}{6}$  et  $w_1 = \frac{2}{3}$ .

**Démonstration 2:** En utilisant la Proposition 5.1.6, on a

$$w_i \stackrel{\text{def}}{=} \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j} dx = \int_0^1 \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t-t_j}{t_i-t_j} dt, \quad \forall i \in \llbracket 0, n \rrbracket$$

On a donc dans le cadre des points équidistants sur  $[0, 1]$ ,  $t_i = i/n$ ,  $\forall i \in \llbracket 0, n \rrbracket$  et

$$\prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - t_j}{t_i - t_j} = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - j/n}{(i - j)/n} = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{nt - j}{i - j}$$

Par exemple, pour  $n = 2$ , on a

$$\begin{aligned} L_0(t) &= \frac{2t-1}{-1} \frac{2t-2}{-2} = (2\lambda-1)(\lambda-1) \\ L_1(t) &= \frac{2t}{1} \frac{2t-2}{-1} = -4(\lambda-1)\lambda \\ L_2(t) &= \frac{2t}{2} \frac{2t-1}{1} = (2\lambda-1)\lambda \end{aligned}$$

et

$$w_0 = \int_0^1 L_0(t) dt = \frac{1}{6}, \quad w_1 = \int_0^1 L_1(t) dt = \frac{2}{3}, \quad w_2 = \int_0^1 L_2(t) dt = \frac{1}{6}$$

On sait par le Lemme 5.1.1 que,  $n = 2$  étant paire, la formule de quadrature élémentaire de Simpson est au moins de degré d'exactitude 3. Pour montrer qu'elle est de degré d'exactitude 3, il suffit de vérifier qu'elle n'est plus exacte pour le monôme  $x^4$  sur l'intervalle  $[0, 1]$ .

En effet, on a

$$\frac{1}{6} \left( 0^4 + 4\left(\frac{1}{2}\right)^4 + 1^4 \right) = \frac{5}{24} \neq \int_0^1 x^4 dx = \frac{1}{5}$$

### EXERCICE 5.1.3

Q. 1

Écrire une fonction algorithmique `WeightsPointsNC` retournant les  $(n+1)$  points et les  $(n+1)$  poids de la formule de quadrature élémentaire de Newton-Cotes à  $(n+1)$  points.

Q. 2

Écrire une fonction algorithmique `QuadElemNC` retournant la valeur de  $\mathcal{Q}_n(f, a, b)$  correspondant à la formule de quadrature élémentaire de Newton-Cotes à  $(n+1)$  points.

### Correction

R. 1

**Algorithme 5.2** Fonction `WeightsPointsNC` retournant le tableau de points  $\mathbf{x}$  donnés correspondant à la discrétisation régulière intervalle  $[a, b]$ . et le tableau des poids  $\mathbf{w}$  associé à un

**Données :**  $a, b$  : deux réels,  $a < b$ ,  
 $n$  :  $n \in \mathbb{N}^*$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^{n+1}$  avec  $\mathbf{x}(i) = x_{i-1}$ ,  $\forall i \in \llbracket 1, n+1 \rrbracket$   
 et  $x_{i-1} = a + (i-1)h$ ,  $h = (b-a)/n$ ,  
 $\mathbf{w}$  : vecteur de  $\mathbb{R}^{n+1}$  avec  $\mathbf{w}(i) = w_{i-1}$ ,  $\forall i \in \llbracket 1, n+1 \rrbracket$

- 1: **Fonction**  $[\mathbf{x}, \mathbf{w}] \leftarrow \text{WeightsPointsNC}(a, b, n)$
- 2:  $\mathbf{x} \leftarrow a : (b-a)/n : b$
- 3:  $\mathbf{w} \leftarrow \text{WeightsFromPoints}(\mathbf{x}, a, b)$
- 4: **Fin**

R. 2

On a de manière générique l'algorithme suivant:

---

**Algorithme 5.3** Fonction `QuadElemGen` retourne la valeur de  $I = (b - a) \sum_{j=0}^n w_j f(x_j)$ .

---

**Données :**  $f$  : une fonction définie de  $[a, b]$  dans  $\mathbb{R}$ ,  
 $a, b$  : deux réels avec  $a < b$   
 $\mathbf{x}$  : vecteur de  $\mathbb{R}^{n+1}$  contenant  $(n + 1)$  points distincts deux à deux dans un intervalle  $[a, b]$  avec la convention  $\mathbf{x}(i) = x_{i-1}, \forall i \in \llbracket 1, n + 1 \rrbracket$   
 $\mathbf{w}$  : vecteur de  $\mathbb{R}^{n+1}$  tel que  $\mathbf{w}(i) = w_{i-1}, \forall i \in \llbracket 1, n + 1 \rrbracket$

**Résultat :**  $I$  : un réel

- 1: **Fonction**  $I \leftarrow \text{QuadElemGen}(f, a, b, \mathbf{x}, \mathbf{w})$
- 2:  $I \leftarrow 0$
- 3: **Pour**  $i \leftarrow 1$  à  $\text{Length}(\mathbf{x})$  **faire**
- 4:      $I \leftarrow I + \mathbf{w}(i) * f(\mathbf{x}(i))$
- 5: **Fin**
- 6:  $I \leftarrow (b - a) * I$
- 7: **Fin**

---

On peut noter que si l'on dispose de la fonction  $s \leftarrow \text{Dot}(\mathbf{u}, \mathbf{v})$  correspondant au produit scalaire de deux vecteurs du même espace alors on a directement

$$I \leftarrow (b - a) * \text{Dot}(\mathbf{w}, f(\mathbf{x})).$$

---

**Algorithme 5.4** Fonction `QuadElemGen` retourne la valeur de  $I = (b - a) \sum_{j=0}^n w_j f(x_j)$  où les poids  $w_i$  et les points  $x_i$  sont ceux définis par la formule de quadrature élémentaire de Newton-Cotes

---

**Données :**  $f$  : une fonction définie de  $[a, b]$  dans  $\mathbb{R}$ ,  
 $a, b$  : deux réels avec  $a < b$ ,  
 $n$  :  $n \in \mathbb{N}^*$

**Résultat :**  $I$  : un réel

- 1: **Fonction**  $I \leftarrow \text{QuadElemNC}(f, a, b, n)$
- 2:  $[\mathbf{x}, \mathbf{w}] \leftarrow \text{WeightsPointsNC}(a, b, n)$
- 3:  $I \leftarrow \text{QuadElemGen}(f, a, b, \mathbf{x}, \mathbf{w})$
- 4: **Fin**

---

~~~~~Fin correction~~~~~

**Remarque 5.1.1.** Pour les méthode de Newton-Cotes, il ne faut pas trop "monter" en ordre car le phénomène de Runge (forte oscillation possible du polynôme d'interpolation sur les bords de l'intervalle) peut conduire à de très grande erreurs. Au delà de  $n = 7$ , des poids négatifs apparaissent dans les formules et les rendent beaucoup plus sensibles aux erreurs d'arrondis.

Sur un exemple très simple, il est possible d'illustrer ce phénomène. Soit  $f(x) = 3x + 2$ . On a démontré que toutes les formules de Newton-Cotes à  $n + 1$  points,  $n \geq 1$ , sont exactes pour le calcul de  $\int_a^b f(x) dx$  car  $f$  est un polynôme de degré 1. De plus les poids  $(w_i)_{i \in \llbracket 0, n \rrbracket}$  peuvent être calculés sous forme fractionnaire : il est immédiat à partir de la formule (5.7) que  $w_i \in \mathbb{Q}$ .

En choisissant, par exemple,  $a = 0$  et  $b = 1$  les  $n + 1$  points  $x_i$  de la formule de Newton-Cotes sont aussi des nombres rationnels. La fonction  $f$  étant polynomiale, on obtient  $f(x_i) \in \mathbb{Q}$ . On en déduit que la formule de Newton-Cotes à  $n + 1$  points donne dans ce cas un résultat (exacte) sous forme fractionnaire. Numériquement, les poids  $w_i \in \mathbb{Q}$  et les points  $x_i \in \mathbb{Q}$  vont être approchés en commettant de très petites erreurs. Sous Sage, logiciel gratuit de calcul formel (entre autres), on peut programmer à la fois le calcul des méthodes de Newton-Cotes exactes (en restant dans le corps  $\mathbb{Q}$ ) et le calcul "approché" correspondant aux méthodes de Newton-Cotes *approchées* (i.e. les  $w_i$  et les  $x_i$  sont approchés avant le calcul de la somme). On représente en Figure 5.5 les erreurs obtenues en fonction de  $n$  par ces deux

approches. Ces courbes en échelle logarithmique en ordonnées ont été obtenues en ajoutant aux erreurs le nombre  $1e - 16$  (pour éviter de prendre le log de zéro).

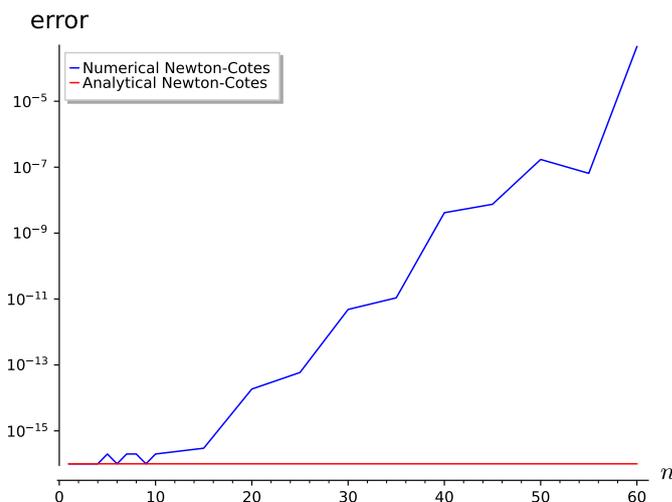


Figure 5.5: Instabilité des méthodes de Newton-Cotes élémentaires

Pour pallier ce problème, on étudiera les méthodes de quadrature composées.

### 5.1.4 Formules élémentaires de Gauss-Legendre

#### EXERCICE 5.1.4

Q. 1

Déterminer les points  $t_0, t_1$  de l'intervalle  $[-1, 1]$  et les poids  $w_0, w_1$  tel que la formule de quadrature

$$\int_{-1}^1 g(t)dt \approx 2 \sum_{i=0}^1 w_i g(t_i)$$

soit de degré d'exactitude 3.

Q. 2

En déduire une formule de quadrature pour le calcul de  $\int_a^b f(x)dx$  qui soit de degré d'exactitude 3.

#### Correction

R. 1

D'après la Proposition 5.1.6, si  $t_0$  et  $t_1$  sont distincts et dans l'intervalle  $[-1, 1]$  alors avec

$$w_0 = \frac{1}{2} \int_{-1}^1 \frac{t - t_1}{t_0 - t_1} dt \text{ et } w_1 = \frac{1}{2} \int_{-1}^1 \frac{t - t_0}{t_1 - t_0} dt$$

la formule de quadrature est de degré d'exactitude 1.

Pour déterminer les points  $t_0$  et  $t_1$ , on utilise la Proposition 5.1.7 (degré maximal d'exactitude) avec  $m = n + 1 = 2$ : la formule de quadrature est de degré  $2n + 1 = 3$  ssi

$$\int_{-1}^1 \pi_1(t) Q(t) dt = 0, \forall Q \in \mathbb{R}_1[X]$$

avec  $\pi_1(t) = (t - t_0)(t - t_1)$ . Par linéarité ceci est équivalent à

$$\int_{-1}^1 \pi_1(t) t^k dt = 0, \forall k \in \llbracket 0, 1 \rrbracket$$

c'est à dire

$$\int_{-1}^1 \pi_1(t) dt = 0 \text{ et } \int_{-1}^1 \pi_1(t) t dt = 0.$$

Or on a

$$\int_{-1}^1 \pi_1(t) dt = \int_{-1}^1 t^2 - (t_0 + t_1)t + t_0 t_1 dt = \frac{2}{3} + 2t_0 t_1$$

et

$$\int_{-1}^1 \pi_1(t) t dt = \int_{-1}^1 t^3 - (t_0 + t_1)t^2 + t_0 t_1 t dt = -\frac{2}{3}(t_0 + t_1).$$

On est amené à résoudre le système non linéaire

$$t_0 t_1 = -\frac{1}{3} \text{ et } -\frac{2}{3}(t_0 + t_1) = 0$$

ce qui donne  $t_0 = -\sqrt{3}/3$  et  $t_1 = \sqrt{3}/3$ .

Il reste à calculer  $w_0$  et  $w_1$ . On a

$$w_0 = \frac{1}{2} \int_{-1}^1 \frac{t - \sqrt{3}/3}{-2\sqrt{3}/3} dt = \frac{1}{2} \int_{-1}^1 \frac{-\sqrt{3}/3}{-2\sqrt{3}/3} dt = 1/2$$

et

$$w_1 = \frac{1}{2} \int_{-1}^1 \frac{t + \sqrt{3}/3}{2\sqrt{3}/3} dt = \frac{1}{2} \int_{-1}^1 \frac{\sqrt{3}/3}{2\sqrt{3}/3} dt = 1/2.$$

La formule de quadrature

$$\int_{-1}^1 g(t) dt \approx g\left(-\frac{\sqrt{3}}{3}\right) + g\left(\frac{\sqrt{3}}{3}\right)$$

est donc d'ordre 3.

#### R. 2

On effectue le changement de variable  $x = \varphi(t) = \frac{a+b}{2} + \frac{b-a}{2}t$ , en appliquant la Proposition 5.1.1 (changement de variable affine), pour obtenir que

$$(b-a) \left( \frac{1}{2} f(x_0) + \frac{1}{2} f(x_1) \right)$$

est une formule de quadrature approchant  $\int_a^b f(x) dx$  avec un degré d'exactitude de 3 où  $x_0 = \varphi(t_0)$  et  $x_1 = \varphi(t_1)$ .

~~~~~Fin correction~~~~~

Avant d'établir des résultats plus généraux, on donne quelques résultats classiques sur les polynômes de Legendre [?].

Les polynômes de Legendre peuvent être définis par la formule de récurrence de Bonnet

$$(n+1)P_{n+1}(t) = (2n+1)tP_n(t) - nP_{n-1}(t), \quad \forall n \geq 1 \quad (5.15)$$

avec  $P_0(t) = 1$  et  $P_1(t) = t$ .

On a les propriétés suivantes:

**prop.1** le polynôme de Legendre  $P_n$  est de degré  $n$ ,

**prop.2** la famille  $\{P_k\}_{k=0}^n$  est une base de  $\mathbb{R}_n[X]$ ,

**prop.3** pour tout  $(m, n) \in \mathbb{N}^2$ , on a

$$\int_{-1}^1 P_m(t) P_n(t) dt = \frac{2}{2n+1} \delta_{m,n}, \quad (5.16)$$

ce qui correspond à l'orthogonalité des polynômes de Legendre pour le produit scalaire

$$\langle P_m, P_n \rangle = \int_{-1}^1 P_m(t) P_n(t) dt.$$

**prop.4** Soit  $n \geq 1$ ,  $P_n$  est scindé sur  $\mathbb{R}$  et ses  $n$  racines, notées  $(t_i)_{i=0}^n$ , sont simples dans  $] -1, 1[$ , c'est à dire

$$P_n(t) = C \prod_{i=0}^{n-1} (t - t_i), \quad C \in \mathbb{R}^*$$

où les  $t_i$  sont 2 à 2 distincts (et ordonnés). Les  $(n+1)$  racines simples de  $P_{n+1}$  sont alors chacune dans l'un des  $(n+1)$  intervalles  $] -1, t_0[$ ,  $]t_0, t_1[$ ,  $\dots$ ,  $]t_{n-2}, t_{n-1}[$ ,  $]t_{n-1}, 1[$ .

**Proposition 5.1.9.** Soit  $(t_i)_{i=0}^n$  les  $(n+1)$  racines distinctes du polynôme de Legendre de degré  $(n+1)$ . On note  $x_i = \frac{a+b}{2} + \frac{b-a}{2}t_i$ ,  $\forall i \in \llbracket 0, n \rrbracket$  et  $w_i$  les poids donnés par (5.7). La formule de quadrature élémentaire

$$\int_a^b f(x)dx \approx (b-a) \sum_{i=0}^n w_i f(x_i)$$

est appelée la formule de quadrature de Gauss-Legendre. C'est l'unique formule de quadrature élémentaire à  $(n+1)$  points ayant pour degré d'exactitude  $2n+1$ .

| $n$ | exactitude | $w_i$ (poids)    | $t_i$ (points)               |
|-----|------------|------------------|------------------------------|
| 0   | 1          | 1                | 0                            |
| 1   | 3          | 1/2, 1/2         | $-\sqrt{1/3}, \sqrt{1/3}$    |
| 2   | 5          | 5/18, 8/18, 5/18 | $-\sqrt{3/5}, 0, \sqrt{3/5}$ |

Table 5.2: Méthodes de Gauss-Legendre sur  $[-1, 1]$

*Preuve.* D'après la Proposition 5.1.1, il suffit de démontrer ce résultat sur l'intervalle  $[-1, 1]$  à l'aide du changement de variable affine  $x = \varphi(t) = \frac{a+b}{2} + \frac{b-a}{2}t$ .

Montrons donc que la formule de quadrature élémentaire

$$\int_{-1}^1 g(t)dt \approx 2 \sum_{i=0}^n w_i g(t_i)$$

est de degré d'exactitude  $2n+1$ .

Comme les poids  $w_i$  sont donnés par (5.7), cette formule est de degré d'exactitude au moins  $n$ . Pour établir qu'elle est de degré d'exactitude  $2n+1$ , il faut, d'après la Proposition 5.1.7 avec  $m = n+1$ , que

$$\int_{-1}^1 \pi_n(t)Q(t)dt = 0, \quad \forall Q \in \mathbb{R}_n[X]$$

avec  $\pi_n(t) = \prod_{i=0}^n (t - t_i)$ . D'après les propriétés des polynômes de Legendre  $P_n$ , on a  $P_{n+1}(t) = C\pi_n(t)$  avec  $C \in \mathbb{R}^*$ . On doit donc avoir de manière équivalente

$$\int_{-1}^1 P_{n+1}(t)Q(t)dt = 0, \quad \forall Q \in \mathbb{R}_n[X].$$

Or, la famille des les polynômes de Legendre  $\{P_0, \dots, P_n\}$  est une base de  $\mathbb{R}_n[X]$  et comme les polynômes de Legendre sont orthogonaux, la relation précédente est vérifiée.

Pour démontrer l'unicité de la formule, il suffit d'établir l'unicité des points de quadrature, les poids étant calculés à partir de ces points.

L'unicité du  $(n+1)$ -uplet  $(x_0, \dots, x_n)$  revient à établir l'unicité du  $(n+1)$ -uplet  $(t_0, \dots, t_n)$ . Supposons qu'il existe  $(t_0, \dots, t_n)$  et  $(\tilde{t}_0, \dots, \tilde{t}_n)$ . Notons  $\pi_n(t) = \prod_{i=0}^n (t - t_i)$  et  $\tilde{\pi}_n(t) = \prod_{i=0}^n (t - \tilde{t}_i)$ . On a donc

$$\int_{-1}^1 \pi_n(t)Q(t)dt = \int_{-1}^1 \tilde{\pi}_n(t)Q(t)dt = 0, \quad \forall Q \in \mathbb{R}_n[X]$$

Le polynôme  $R = \pi_n - \tilde{\pi}_n$  est dans  $\mathbb{R}_n[X]$  car les polynômes  $\pi_n$  et  $\tilde{\pi}_n$  de  $\mathbb{R}_{n+1}[X]$  sont unitaires. On a alors

$$\int_{-1}^1 R(t)Q(t)dt = 0, \quad \forall Q \in \mathbb{R}_n[X]$$

En choisissant  $Q = \mathbb{R}$ , on obtient

$$\int_{-1}^1 R^2(t) dt = 0$$

ce qui entraîne  $R = 0$ . □

**Théorème 5.1.1.** Soient  $f \in \mathcal{C}^{2n+2}([a, b]; \mathbb{R})$  et  $\mathcal{Q}_n(f, a, b)$  la formule de quadrature de Gauss-Legendre définie dans la Proposition ???. Alors on a

$$\left| \int_a^b f(x) dx - \mathcal{Q}_n(f, a, b) \right| \leq \frac{\|f^{(2n+2)}\|_\infty}{(2n+2)!} \int_a^b \pi_n(x)^2 dx \quad (5.17)$$

où  $\pi_n(x) = \prod_{i=0}^n (x - x_i)$ , les  $x_i$  étant les points de la formule de quadrature.

*Preuve.* On va utiliser  $H_n$ , le polynôme d'interpolation de Lagrange-Hermite aux  $(n+1)$  points de la quadrature. D'après le Théorème 4.2.1, c'est l'unique polynôme de degré au plus  $2n+1$  vérifiant

$$H_n(x_i) = f(x_i) \text{ et } H'_n(x_i) = f'(x_i), \quad \forall i \in \llbracket 0, n \rrbracket.$$

La fonction  $f$  étant dans  $\mathcal{C}^{2n+2}([a, b]; \mathbb{R})$ , on peut appliquer le Théorème 4.2.2 et pour tout  $x \in [a, b]$ ,  $\exists \xi_x \in ]a, b[$  tel que

$$f(x) - H_n(x) = \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \prod_{i=0}^n (x - x_i)^2.$$

Ensuite on intègre cette relation:

$$\int_a^b (f(x) - H_n(x)) dx = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \prod_{i=0}^n (x - x_i)^2 dx.$$

Le polynôme  $H_n$  étant de degré  $2n+1$ , la formule de quadrature est donc exacte et on a:

$$\begin{aligned} \int_a^b H_n(x) dx &= \mathcal{Q}_n(H_n, a, b) \\ &= (b-a) \sum_{i=0}^n w_i H_n(x_i) \\ &= (b-a) \sum_{i=0}^n w_i f(x_i) \\ &= \mathcal{Q}_n(f, a, b). \end{aligned}$$

On a donc

$$\int_a^b f(x) dx - \mathcal{Q}_n(f, a, b) = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} (\pi_n(x))^2 dx.$$

Ce qui donne

$$\left| \int_a^b f(x) dx - \mathcal{Q}_n(f, a, b) \right| \leq \frac{\|f^{(2n+2)}\|_\infty}{(2n+2)!} \int_a^b \pi_n(x)^2 dx. \quad \square$$

### EXERCICE 5.1.5

L'objectif de cet exercice est de calculer les points et les poids de la formule de quadrature de Gauss-Legendre à  $(n+1)$  points. La formule de quadrature de Gauss-Legendre à  $(n+1)$  points sur  $[-1, 1]$  est donnée par

$$\int_{-1}^1 g(t) dt \approx 2 \sum_{i=0}^n w_i g(t_i)$$

où les  $(t_i)_{i=0}^n$  sont les  $(n+1)$  racines du polynôme de Legendre  $P_{n+1}(t)$ . Cette formule a pour degré d'exactitude  $2n+1$ .

Soient  $\langle P, Q \rangle = \int_{-1}^1 P(t)Q(t) dt$  le produit scalaire sur  $\mathbb{R}[X]$  et  $\|P\| = \langle P, P \rangle^{1/2}$  la norme associée.

Soit  $M_n$  le polynôme de Legendre normalisé de degré  $(n+1)$ ,  $M_n = \frac{P_n}{\|P_n\|}$ . On utilisera les résultats

sur les polynômes de Legendre rappelés en cours.

**Q. 1**

Montrer que

$$c_{n+1}M_{n+1}(t) = tM_n(t) - c_nM_{n-1}(t), \quad n > 1 \quad (5.1)$$

avec

$$M_0(t) = \sqrt{\frac{1}{2}}, \quad M_1(t) = \sqrt{\frac{3}{2}}t \quad \text{et} \quad c_n = \sqrt{\frac{n^2}{4n^2 - 1}}$$

On définit le vecteur  $\mathbf{M}(t)$  de  $\mathbb{R}^{n+1}$  par

$$\mathbf{M}(t) = (M_0(t), \dots, M_n(t))^t.$$

**Q. 2**

Montrer que l'on a

$$t\mathbf{M}(t) = \mathbb{A}\mathbf{M}(t) + c_{n+1}M_{n+1}(t)\mathbf{e}_{n+1} \quad (5.2)$$

où l'on explicitera la matrice tridiagonale  $\mathbb{A} \in \mathcal{M}_{n+1}(\mathbb{R})$  en fonction des coefficients  $c_1, \dots, c_n$ . Le vecteur  $\mathbf{e}_{n+1}$  étant le  $(n+1)$ -ème vecteur de la base canonique de  $\mathbb{R}^{n+1}$ .

**Q. 3**

En déduire que les  $(n+1)$  racines distinctes de  $M_{n+1} \in \mathbb{R}_{n+1}[X]$  sont les  $(n+1)$  valeurs propres de  $\mathbb{A}$ .

**Q. 4**

Montrer que

$$2 \sum_{k=0}^n w_k M_i(t_k) M_j(t_k) = \delta_{i,j}, \quad \forall (i, j) \in \llbracket 0, n \rrbracket^2 \quad (5.3)$$

où  $\delta_{i,j} = 0$ , si  $i \neq j$  et  $\delta_{i,i} = 1$ .

On note  $\mathbb{W} \in \mathcal{M}_{n+1}(\mathbb{R})$  la matrice diagonale, de diagonale  $(w_0, \dots, w_n)$  et  $\mathbb{P} \in \mathcal{M}_{n+1}(\mathbb{R})$  la matrice définie par  $\mathbb{P}_{i+1, j+1} = M_j(t_i)$ ,  $\forall (i, j) \in \llbracket 0, n \rrbracket^2$ .

**Q. 5**

- Montrer que  $2\mathbb{P}^t\mathbb{W}\mathbb{P} = \mathbb{I}$ .
- En déduire que  $\mathbb{W}^{-1} = 2\mathbb{P}\mathbb{P}^t$ .
- En déduire que  $\frac{1}{w_i} = 2 \sum_{k=0}^n (M_k(t_i))^2$ ,  $\forall i \in \llbracket 0, n \rrbracket$ .

On suppose que l'on dispose de la fonction **algorithmique** `eig(A)` retournant l'ensemble des valeurs propres d'une matrice symétrique  $\mathbb{A} \in \mathcal{M}_{n+1}(\mathbb{R})$  dans l'ordre croissant sous la forme d'un vecteur de  $\mathbb{R}^{n+1}$ .

**Q. 6**

- Ecrire la fonction `[t, w] ← GaussLegendre(n)` retournant le tableau des points  $\mathbf{t}$  et le tableau des poids  $\mathbf{w}$  en utilisant les résultats obtenus dans cet exercice.
- Ecrire la fonction `I ← QuadElemGaussLegendre(f, a, b, n)` retournant une approximation de  $\int_a^b f(x)dx$  en utilisant la formule de quadrature de Gauss-Legendre à  $(n+1)$  points sur l'intervalle  $[a, b]$ .

## Correction

**R. 1**

Par définition, on a  $M_n = \frac{P_n}{\|P_n\|}$  et

$$\|P_n\|^2 = \langle P_n, P_n \rangle = \int_{-1}^1 P_n(t)P_n(t)dx = \frac{2}{2n+1}.$$

On en déduit que

$$M_n = \sqrt{\frac{2n+1}{2}}P_n.$$

De plus, par la formule de récurrence de Bonnet, on obtient

$$M_0(t) = \sqrt{\frac{1}{2}} \text{ et } M_1(t) = \sqrt{\frac{3}{2}}t$$

ainsi que,  $\forall n \geq 1$ ,

$$\begin{aligned} (n+1)\sqrt{\frac{2}{2n+3}}M_{n+1}(t) &= (2n+1)\sqrt{\frac{2}{2n+1}}tM_n(t) - n\sqrt{\frac{2}{2n-1}}M_{n-1} \\ &= \sqrt{2(2n+1)}tM_n(t) - n\sqrt{\frac{2}{2n-1}}M_{n-1} \end{aligned}$$

En multipliant cette équation par  $\sqrt{\frac{1}{2(2n+1)}}$ , on a

$$(n+1)\sqrt{\frac{2}{2n+3}}\sqrt{\frac{1}{2(2n+1)}}M_{n+1}(t) = tM_n(t) - n\sqrt{\frac{2}{2n-1}}\sqrt{\frac{1}{2(2n+1)}}M_{n-1}$$

Or on a

$$n\sqrt{\frac{2}{2n-1}}\sqrt{\frac{1}{2(2n+1)}} = \sqrt{\frac{n^2}{(2n-1)(2n+1)}} = c_n$$

et

$$(n+1)\sqrt{\frac{2}{2n+3}}\sqrt{\frac{1}{2(2n+1)}} = \sqrt{\frac{(n+1)^2}{(2(n+1)-1)(2(n+1)+1)}} = c_{n+1}$$

ce qui démontre le résultat voulu.

**R. 2**

On déduit de la question précédente que

$$tM_0(t) = \sqrt{\frac{1}{3}}t = c_1M_1(t)$$

et

$$tM_i(t) = c_iM_{i-1}(t) + c_{i+1}M_{i+1}(t), \quad \forall i \in \llbracket 1, n \rrbracket.$$

Ces  $(n+1)$  équations peuvent s'écrire matriciellement sous la forme

$$\begin{aligned} t \begin{pmatrix} M_0(t) \\ M_1(t) \\ \vdots \\ \vdots \\ M_{n-1}(t) \\ M_n(t) \end{pmatrix} &= \begin{pmatrix} 0 & c_1 & 0 & \dots & \dots & 0 \\ c_1 & 0 & c_2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & c_{n-1} & 0 & c_n \\ 0 & \dots & \dots & 0 & c_n & 0 \end{pmatrix} \begin{pmatrix} M_0(t) \\ M_1(t) \\ \vdots \\ \vdots \\ M_{n-1}(t) \\ M_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ c_{n+1}M_{n+1}(t) \end{pmatrix} \\ &= \mathbb{A}\mathbf{M}(t) + c_{n+1}M_{n+1}(t)\mathbf{e}_{n+1}. \end{aligned}$$

**R. 3**

Le polynôme de Legendre  $P_{n+1} \in \mathbb{R}_{n+1}[X]$  admet  $(n+1)$  racines simples distinctes dans  $] -1, 1[$  notées  $(t_i)_{i=0}^n$ . (voir rappels) Donc le polynôme de Legendre normalisé  $M_{n+1} \in \mathbb{R}_{n+1}[X]$  a les mêmes racines et on déduit de la question précédente que

$$\mathbb{A}\mathbf{M}(t_i) = t_i\mathbf{M}(t_i), \quad \forall i \in \llbracket 0, n \rrbracket.$$

Comme les  $(n+1)$  racines de  $P_{n+1}$  séparent strictement les  $n$  racines de  $P_n$  (voir rappels), alors  $P_n(t_i) \neq 0$  et donc  $M_n(t_i) \neq 0$ . On en déduit que le vecteur  $\mathbf{M}(t_i)$  est non nul et,

$$\forall i \in \llbracket 0, n \rrbracket, (t_i, \mathbf{M}(t_i)) \text{ est un mode propre de } \mathbb{A}.$$

On peut noter que  $A$  est symétrique et donc ses valeurs propres sont réelles.

Les  $(n+1)$  valeurs propres de la matrice  $A$  sont les  $(n+1)$  racines de  $P_{n+1}$ , et donc les  $(n+1)$  points de la formule de quadrature de Gauss-Legendre.

**R. 4**

Par construction, on a

$$\int_{-1}^1 M_i(t)M_j(t)dx = \delta_{i,j}, \quad \forall (i,j) \in \llbracket 0, n \rrbracket.$$

On a  $M_i \in \mathbb{R}_i[X]$  et  $M_j \in \mathbb{R}_j[X]$ , ce qui donne  $M_iM_j \in \mathbb{R}_{i+j}[X]$  avec  $i+j \leq 2n$ . Or la formule de quadrature de Gauss-Legendre à  $(n+1)$  points a pour degré d'exactitude  $2n+1$ , elle est donc exacte pour le polynôme  $M_iM_j$ . On en déduit alors

$$\delta_{i,j} = \int_{-1}^1 M_i(t)M_j(t)dt = 2 \sum_{k=0}^n w_k M_i(t_k)M_j(t_k), \quad \forall (i,j) \in \llbracket 0, n \rrbracket$$

**R. 5**

a. Soit  $(i,j) \in \llbracket 1, n+1 \rrbracket^2$ , on a

$$\begin{aligned} (\mathbb{P}^t \mathbb{W} \mathbb{P})_{i,j} &= \sum_{k=1}^{n+1} (\mathbb{P}^t)_{i,k} (\mathbb{W} \mathbb{P})_{k,j} \\ &= \sum_{k=1}^{n+1} \mathbb{P}_{k,i} (\mathbb{W} \mathbb{P})_{k,j} \end{aligned}$$

et, comme  $\mathbb{W}$  est diagonale,

$$(\mathbb{W} \mathbb{P})_{k,j} = \sum_{l=1}^{n+1} \mathbb{W}_{k,l} \mathbb{P}_{l,j} = \mathbb{W}_{k,k} \mathbb{P}_{k,j}.$$

On obtient donc

$$(\mathbb{P}^t \mathbb{W} \mathbb{P})_{i,j} = \sum_{k=1}^{n+1} \mathbb{P}_{k,i} \mathbb{W}_{k,k} \mathbb{P}_{k,j} = \sum_{k=1}^{n+1} w_{k-1} M_{i-1}(t_{k-1}) M_{j-1}(t_{k-1}).$$

En utilisant la relation démontré dans la question précédente, on a

$$(\mathbb{P}^t \mathbb{W} \mathbb{P})_{i,j} = \frac{1}{2} \delta_{i-1, j-1} = \frac{1}{2} \delta_{i,j}$$

c'est à dire

$$\mathbb{P}^t \mathbb{W} \mathbb{P} = \frac{1}{2} \mathbb{I}$$

et on en déduit que  $\mathbb{P}$  et  $\mathbb{W}$  sont inversibles .

b. A partir de  $\mathbb{P}^t \mathbb{W} \mathbb{P} = \frac{1}{2} \mathbb{I}$ , on déduit

$$2\mathbb{I} = (\mathbb{P}^t \mathbb{W} \mathbb{P})^{-1} = 2\mathbb{I} = \mathbb{P}^{-1} \mathbb{W}^{-1} (\mathbb{P}^t)^{-1}$$

En multipliant par  $\mathbb{P}$  à gauche et par  $\mathbb{P}^t$  à droite, on obtient

$$\mathbb{W}^{-1} = 2\mathbb{P} \mathbb{P}^t.$$

c. Comme la matrice  $\mathbb{W}$  est diagonale inversible, son inverse est diagonale et on a

$$(\mathbb{W}^{-1})_{i,i} = \frac{1}{\mathbb{W}_{i,i}} = \frac{1}{w_{i-1}}, \quad \forall i \in \llbracket 1, n+1 \rrbracket.$$

On a donc

$$\begin{aligned}
 \frac{1}{w_{i-1}} &= 2(\mathbb{P}\mathbb{P}^t)_{i,i} \\
 &= 2 \sum_{j=1}^{n+1} \mathbb{P}_{i,j}(\mathbb{P}^t)_{j,i} \\
 &= 2 \sum_{j=1}^{n+1} \mathbb{P}_{i,j}^2 \\
 &= 2 \sum_{k=0}^n (\mathbb{M}_k(t_{i-1}))^2, \quad \forall i \in \llbracket 1, n+1 \rrbracket.
 \end{aligned}$$

**R. 6**

- a. Les  $(n+1)$  points  $(t_i)_{i=0}^n$  de la méthode de quadrature de Gauss-Legendre sur  $[-1, 1]$ , sont les racines du polynôme de Legendre  $P_{n+1}$  de degré  $n+1$ . Pour les calculer, on va utiliser le fait que ce sont les valeurs propres de la matrice symétrique  $\mathbb{A} \in \mathcal{M}_{n+1}(\mathbb{R})$

$$\begin{pmatrix}
 0 & c_1 & 0 & \dots & \dots & 0 \\
 c_1 & 0 & c_2 & 0 & \dots & 0 \\
 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\
 \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\
 0 & \dots & 0 & c_{n-1} & 0 & c_n \\
 0 & \dots & \dots & 0 & c_n & 0
 \end{pmatrix}$$

avec  $c_k = \sqrt{\frac{k^2}{4k^2-1}}$ ,  $\forall k \geq 1$ .

Pour calculer les poids  $(w_i)_{i=0}^n$ , on va utiliser la formule

$$\frac{1}{w_i} = 2 \sum_{k=0}^n (\mathbb{M}_k(t_i))^2, \quad \forall i \in \llbracket 0, n \rrbracket$$

conjointement avec la formule de récurrence

$$\mathbb{M}_k(t) = \frac{1}{c_k} (t\mathbb{M}_{k-1}(t) - c_{k-1}\mathbb{M}_{k-2}(t)), \quad k \geq 2, \quad \text{avec } \mathbb{M}_0(t) = \sqrt{\frac{1}{2}}, \quad \mathbb{M}_1(t) = \sqrt{\frac{3}{2}}t.$$

**Algorithme 5.5** Fonction `GaussLegendre` retournant le tableau des points  $\mathbf{t}$  et le tableau des poids  $\mathbf{w}$

**Données :**  $n$  :  $n \in \mathbb{N}$

**Résultat :**  $\mathbf{t}$  : vecteur de  $\mathbb{R}^{n+1}$  avec  $\mathbf{t}(i) = t_{i-1}, \forall i \in \llbracket 1, n+1 \rrbracket$

$\mathbf{w}$  : vecteur de  $\mathbb{R}^{n+1}$  avec  $\mathbf{w}(i) = w_{i-1}, \forall i \in \llbracket 1, n+1 \rrbracket$

```

1: Fonction  $[\mathbf{t}, \mathbf{w}] \leftarrow \text{GaussLegendre}(n)$ 
2:  $\mathbf{c} \leftarrow \mathbf{O}_n$ 
3:  $\mathbb{A} \leftarrow \mathbb{O}_{n+1, n+1}$ 
4: Pour  $k \leftarrow 1$  à  $n$  faire
5:    $\mathbf{c}(k) \leftarrow \text{sqrt}(k^2 / (4 * k^2 - 1))$ 
6:    $\mathbb{A}(k, k+1) \leftarrow \mathbf{c}(k)$ 
7:    $\mathbb{A}(k+1, k) \leftarrow \mathbf{c}(k)$ 
8: Fin
9:  $\mathbf{t} \leftarrow \text{eig}(\mathbb{A})$ 
10: Pour  $i \leftarrow 1$  à  $n+1$  faire
11:    $M0 \leftarrow \text{sqrt}(1/2)$ 
12:    $M1 \leftarrow \text{sqrt}(3/2) * \mathbf{t}(i)$ 
13:    $S \leftarrow M0^2 + M1^2$ 
14:   Pour  $k \leftarrow 2$  à  $n$  faire
15:      $M \leftarrow (1/\mathbf{c}(k)) * (M1 * \mathbf{t}(i) - \mathbf{c}(k-1) * M0)$ 
16:      $S \leftarrow S + M^2$ 
17:      $M0 \leftarrow M1$ 
18:      $M1 \leftarrow M$ 
19:   Fin
20:    $w(i) \leftarrow 1/(2 * S)$ 
21: Fin
22: Fin

```

b. On va utiliser la formule

$$I = (b - a) = \sum_{i=0}^n w_i f(x_i)$$

avec  $x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i$  où les points  $(t_i)_{i=0}^n$  et les poids  $(w_i)_{i=0}^n$  sont ceux de la méthode de quadrature de Gauss-Legendre sur  $[-1, 1]$ .

**Algorithme 5.6** Fonction `QuadElemGaussLegendre` retournant une approximation de  $\int_a^b f(x) dx$  en utilisant la formule de quadrature de Gauss-Legendre à  $n+1$  points sur l'intervalle  $[a, b]$ .

**Données :**  $f$  : une fonction de  $[a, b]$  à valeurs réels

$a, b$  : deux réels avec  $a < b$

$n$  :  $n \in \mathbb{N}$

**Résultat :**  $I$  : un réel

```

1: Fonction  $I \leftarrow \text{QuadElemGaussLegendre}(f, a, b, n)$ 
2:  $[\mathbf{t}, \mathbf{w}] \leftarrow \text{GaussLegendre}(n)$ 
3:  $I \leftarrow 0$ 
4: Pour  $i \leftarrow 1$  à  $n+1$  faire
5:    $I \leftarrow I + \mathbf{w}(i) * f((a+b)/2 + (b-a)/2 * \mathbf{t}(i))$ 
6: Fin
7:  $I \leftarrow (b-a) * I$ 
8: Fin

```

~~~~~ Fin correction ~~~~~

## 5.2 Méthodes de quadrature composées

Soit  $f$  une fonction définie et intégrable sur un intervalle  $[\alpha, \beta]$  donné. On propose de chercher une approximation de

$$I = \int_{\alpha}^{\beta} f(x) dx.$$

Ces méthodes consistent en l'utilisation de la relation de Chasles pour décomposer l'intégrale en une somme d'intégrales sur des domaines plus petits puis à approcher ces dernières par une formule de quadrature élémentaire à  $n + 1$  points.

La **méthode de quadrature composée associée à  $\mathcal{Q}_n$** , notée  $\mathcal{Q}_{k,n}^{\text{comp}}$ , est alors donnée par

$$\mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta) = \sum_{i=1}^k \mathcal{Q}_n(f, \alpha_{i-1}, \alpha_i) \approx \int_{\alpha}^{\beta} f(x) dx \quad (5.18)$$

**Définition 5.2.1.** Soit  $(\alpha_i)_{i \in [0, k]}$  une subdivision de l'intervalle  $[\alpha, \beta]$ :

$$\alpha = \alpha_0 < \alpha_1 < \dots < \alpha_k = \beta.$$

On a alors

$$\int_{\alpha}^{\beta} f(x) dx = \sum_{i=1}^k \int_{\alpha_{i-1}}^{\alpha_i} f(x) dx. \quad (5.19)$$

Soit  $\mathcal{Q}_n(g, a, b)$  la formule de quadrature élémentaire à  $n + 1$  points d'ordre  $p$  donnée par

$$\mathcal{Q}_n(g, a, b) \stackrel{\text{def}}{=} (b - a) \sum_{j=0}^n w_j g(x_j) \approx \int_a^b g(x) dx.$$

La **méthode de quadrature composée associée à  $\mathcal{Q}_n$** , notée  $\mathcal{Q}_{k,n}^{\text{comp}}$ , est donnée par

$$\mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta) = \sum_{i=1}^k \mathcal{Q}_n(f, \alpha_{i-1}, \alpha_i) \approx \int_{\alpha}^{\beta} f(x) dx \quad (5.20)$$

La proposition suivante est immédiate

**Proposition 5.2.1.** Soit  $\mathcal{Q}_n$  une formule de quadrature élémentaire à  $n + 1$  points. Si  $\mathcal{Q}_n$  est d'ordre  $p$  alors la méthode de quadrature composée associée est aussi d'ordre  $p$  : elle est exacte pour tout polynôme de degré  $p$ .

### 5.2.1 Exemples

Pour les trois formules composites qui suivent on choisit une discrétisation régulière. Soit  $(\alpha_i)_{i \in [0, k]}$  une discrétisation régulière de l'intervalle  $[\alpha; \beta]$  :  $\alpha_i = \alpha + ih$  avec  $h = (\beta - \alpha)/k$  le pas de la discrétisation.

#### Formule composite des points milieux

La formule élémentaire des points milieux est donnée par

$$\mathcal{Q}_0(g, a, b) \stackrel{\text{def}}{=} (b - a) g\left(\frac{a + b}{2}\right)$$

On note  $m_j = \frac{\alpha_{j-1} + \alpha_j}{2}$  le point milieu de l'intervalle  $[\alpha_{j-1}, \alpha_j]$ .

$$\int_{\alpha}^{\beta} f(x) dx = \sum_{j=1}^k \int_{\alpha_{j-1}}^{\alpha_j} f(x) dx \approx \sum_{j=1}^k \mathcal{Q}_0(f, \alpha_{j-1}, \alpha_j) = h \sum_{j=1}^k f(m_j) \quad (5.21)$$

On illustre graphiquement l'approximation d'une intégrale par cette formule en Figure 5.6.

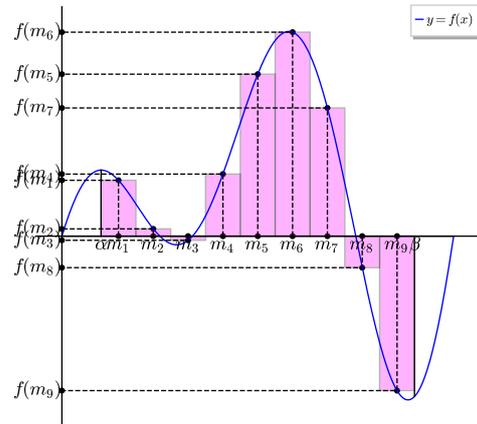


Figure 5.6: Formule composite des points milieux :  $\int_{\alpha}^{\beta} f(x)dx \approx h \sum_{j=1}^k f(m_j)$  (aire de la surface colorée)

### Formule composite des trapèzes

La formule élémentaire des points trapèzes est donnée par

$$\mathcal{Q}_1(g, a, b) \stackrel{\text{def}}{=} \frac{b-a}{2}(g(a) + g(b))$$

On obtient alors

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{j=1}^k \int_{\alpha_{j-1}}^{\alpha_j} f(x)dx \approx \sum_{j=1}^k \mathcal{Q}_0(f, \alpha_{j-1}, \alpha_j) = \frac{h}{2} \sum_{j=1}^k (f(\alpha_{j-1}) + f(\alpha_j)) \\ &\approx \frac{h}{2} \left( f(\alpha_0) + 2 \sum_{j=1}^{k-1} f(\alpha_j) + f(\alpha_k) \right) \end{aligned} \quad (5.22)$$

C'est une formule d'ordre 2 par rapport à  $h$ .

On illustre graphiquement l'approximation d'une intégrale par cette formule en Figure 5.7.

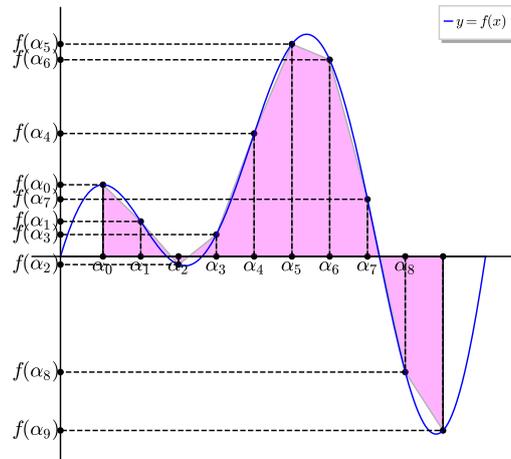


Figure 5.7: Formule composite des trapèzes :  $\int_{\alpha}^{\beta} f(x)dx \approx \frac{h}{2} \left( f(\alpha_0) + 2 \sum_{j=1}^{k-1} f(\alpha_j) + f(\alpha_k) \right)$  (aire de la surface colorée)

### Formule composite de Simpson

#### EXERCICE 5.2.1

Ecrire une fonction algorithmique `QuadSimpson` retournant une approximation de l'intégrale d'une fonction  $f$  sur l'intervalle  $[\alpha, \beta]$  utilisant la méthode de quadrature composée de Simpson en **minimisant** le nombre d'appels à la fonction  $f$ . On rappelle que la formule élémentaire de Simpson est donnée par

$$\mathcal{Q}_2(g, a, b) \stackrel{\text{def}}{=} \frac{b-a}{6} (g(a) + 4g(\frac{a+b}{2}) + g(b)).$$

**Correction** En notant  $m_j = \frac{\alpha_{j-1} + \alpha_j}{2}$  le point milieu de l'intervalle  $[\alpha_{j-1}, \alpha_j]$ , on obtient

$$\begin{aligned} \int_a^b f(x)dx &\sum_{j=1}^k \int_{\alpha_{j-1}}^{\alpha_j} f(x)dx \approx \sum_{j=1}^k \mathcal{Q}_2(f, \alpha_{j-1}, \alpha_j) = \frac{h}{6} \sum_{j=1}^k (f(\alpha_{j-1}) + 4f(m_j) + f(\alpha_j)) \\ &\approx \frac{h}{6} \left( 4 \sum_{j=1}^k f(m_j) + f(\alpha_0) + 2 \sum_{j=1}^{k-1} f(\alpha_j) + f(\alpha_k) \right) \end{aligned} \quad (5.23)$$

**Algorithme 5.7** Fonction `QuadSimpson` retourne une approximation de l'intégrale d'une fonction  $f$  sur l'intervalle  $[\alpha, \beta]$  utilisant la méthode de quadrature composée de Simpson en **minimisant** le nombre d'appels à la fonction  $f$ .

**Données :**  $f$  : une fonction définie de  $[\alpha, \beta]$  dans  $\mathbb{R}$ ,  
 $\alpha, \beta$  : deux réels avec  $\alpha < \beta$ ,  
 $k$  :  $n \in \mathbb{N}^*$

**Résultat :**  $I$  : un réel

1: **Fonction**  $I \leftarrow \text{QuadSimpson}(f, \alpha, \beta, k)$

2:  $h \leftarrow (\beta - \alpha)/k$

3:  $\mathbf{x} \leftarrow \alpha : h : \beta$

4:  $\mathbf{m} \leftarrow \alpha + h/2 : h : \beta$

5:  $S \leftarrow 0$

$\triangleright$  Calcul de  $\sum_{j=1}^k f(m_j)$

6: **Pour**  $j \leftarrow 1$  à  $k$  faire

7:  $S \leftarrow S + f(\mathbf{m}(j))$

8: **Fin**

9:  $I \leftarrow 4 * S$

10:  $S \leftarrow 0$

$\triangleright$  Calcul de  $\sum_{j=1}^{k-1} f(\alpha_j) = \sum_{j=2}^k f(\mathbf{x}_j)$

11: **Pour**  $j \leftarrow 2$  à  $k$  faire

12:  $S \leftarrow S + f(\mathbf{x}(j))$

13: **Fin**

14:  $I \leftarrow (h/6) * (I + 2 * S + f(\mathbf{x}(1)) + f(\mathbf{x}(k+1)))$

15: **Fin**

~~~~~Fin correction~~~~~

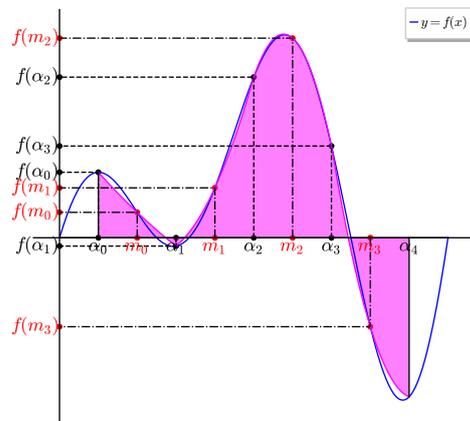


Figure 5.8: Formule composite de Simpson :  $\int_{\alpha}^{\beta} f(x) dx \approx \frac{h}{6} \left( 4 \sum_{j=1}^k f(m_j) + f(\alpha_0) + 2 \sum_{j=1}^{k-1} f(\alpha_j) + f(\alpha_k) \right)$   
 (aire de la surface colorée)

## 5.2.2 Erreurs des méthodes de quadrature composées

Soit  $\mathcal{Q}_{k,n}^{\text{comp}}$  une méthode de quadrature composée associée à une méthode de quadrature élémentaire  $\mathcal{Q}_n$ . On note  $\mathcal{E}_{\alpha,\beta}^{\text{comp}}(f)$  l'erreur de cette méthode de quadrature composée :

$$\mathcal{E}_{\alpha,\beta}^{\text{comp}}(f) = \int_{\alpha}^{\beta} f(x)dx - \mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta). \quad (5.24)$$

On a alors

$$\mathcal{E}_{\alpha,\beta}^{\text{comp}}(f) = \sum_{j=1}^k \left( \int_{\alpha_{j-1}}^{\alpha_j} f(x)dx - \mathcal{Q}_n(f, \alpha_{j-1}, \alpha_j) \right) = \sum_{j=1}^k \mathcal{E}_{\alpha_{j-1}, \alpha_j}(f)$$

Dans le cadre des méthodes composées de Newton-Cotes, on peut démontrer (voir [2]), la majoration suivante

$$\max_{x \in [a,b]} \left| \prod_{i=0}^n (x - x_i) \right| \leq C \frac{e^{-n}}{\sqrt{n} \log(n)} (b-a)^{n+1}. \quad (5.25)$$

où  $(x_i)_{i \in [0,n]}$  est la discrétisation régulière de  $[a,b]$  et  $C > 0$ . On suppose que  $f \in \mathcal{C}^{n+1}([a,b]; \mathbb{R})$ . En notant  $h_j = \alpha_j - \alpha_{j-1}$  et en utilisant les majorations (5.8) et (5.25) on obtient

$$\begin{aligned} |\mathcal{E}_{\alpha,\beta}^{\text{comp}}(f)| &\leq \sum_{j=1}^k |\mathcal{E}_{\alpha_{j-1}, \alpha_j}(f)| \\ &\leq \sum_{j=1}^k \frac{K_n}{(n+1)!} \max_{x \in [\alpha_{j-1}, \alpha_j]} |f^{(n+1)}(x)| h_j^{n+2} \quad \text{avec } K_n = C \frac{e^{-n}}{\sqrt{n} \log(n)} \\ &\leq K_n \frac{h^{n+1}}{(n+1)!} \max_{x \in [\alpha,\beta]} |f^{(n+1)}(x)| \sum_{j=1}^k h_j \quad \text{avec } h = \max_{j \in [1,k]} h_j \\ &\leq K_n (\beta - \alpha) \frac{h^{n+1}}{(n+1)!} \|f^{(n+1)}\|_{\infty} \end{aligned}$$

Cette majoration n'est pas optimale et n'est valable que pour les formules composées de Newton-Cotes. A l'aide des **noyaux de Peano**, on peut démontrer le théorème suivant :

**Théorème 5.2.1** ([2], page 43 (admis)). Soient  $\mathcal{Q}_{k,n}^{\text{comp}}$  une méthode de quadrature composée associée à une méthode de quadrature élémentaire  $\mathcal{Q}_n$  de degré d'exactitude  $p \geq n$  et  $f \in \mathcal{C}^{p+1}([\alpha, \beta]; \mathbb{R})$ . On a alors

$$\left| \int_{\alpha}^{\beta} f(x)dx - \mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta) \right| \leq C_p (\beta - \alpha) h^{p+1} \|f^{(p+1)}\|_{\infty} \quad (5.26)$$

avec  $h = \max_{j \in [1,k]} (\alpha_j - \alpha_{j-1})$  et  $C_p > 0$ . Ceci s'écrit aussi sous la forme

$$\left| \int_{\alpha}^{\beta} f(x)dx - \mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta) \right| = \mathcal{O}(h^{p+1}) \quad (5.27)$$

et son **ordre de convergence** est  $p + 1$ .

On illustre ce théorème pour les méthodes de Newton-Cotes composées  $\mathcal{Q}_{k,n}^{\text{comp}}$ , pour  $n \in [1, 8]$ , par les Figures 5.9 et 5.10.

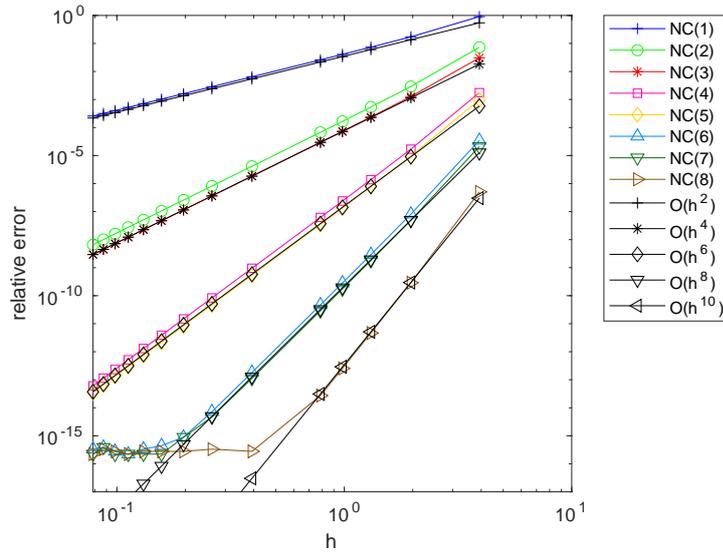


Figure 5.9: Erreur des méthodes de Newton-Cotes composées pour le calcul de  $\int_0^{5\pi/2} \cos(x)dx$ , NC(n) correspondant à  $Q_{k,n}^{\text{comp}}$  et  $h = \frac{5\pi}{2k}$ .

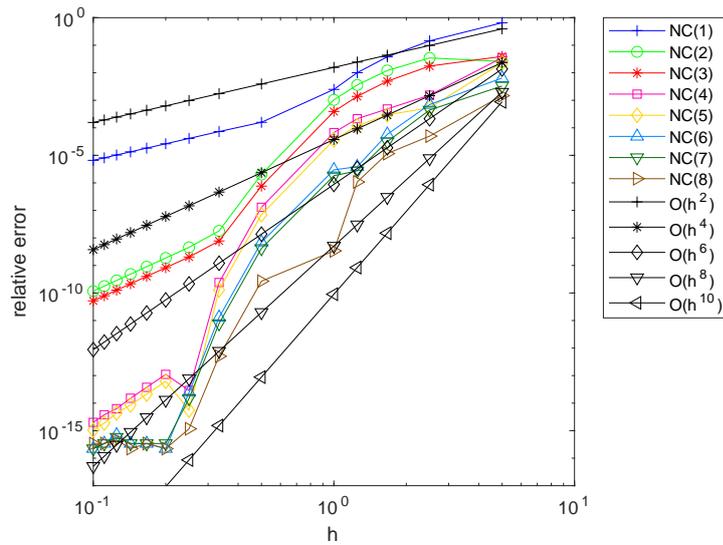


Figure 5.10: Erreur des méthodes de Newton-Cotes composées pour le calcul de  $\int_{-5}^5 \frac{1}{1+x^2}dx$ , NC(n) correspondant à  $Q_{k,n}^{\text{comp}}$  et  $h = \frac{10}{k}$ .

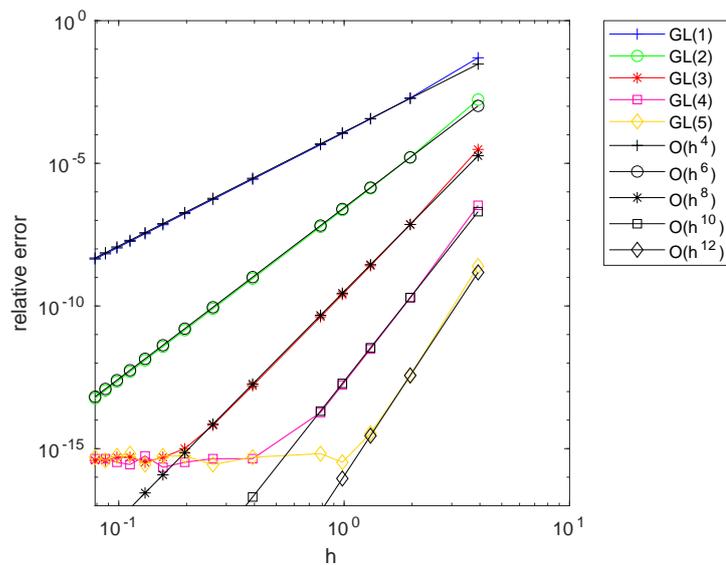


Figure 5.11: Erreur des méthodes de Gauss-Legendre composées pour le calcul de  $\int_0^{5\pi/2} \cos(x)dx$ , GL(n) correspondant à  $Q_{k,n}^{\text{comp}}$  et  $h = \frac{5\pi}{2k}$ .

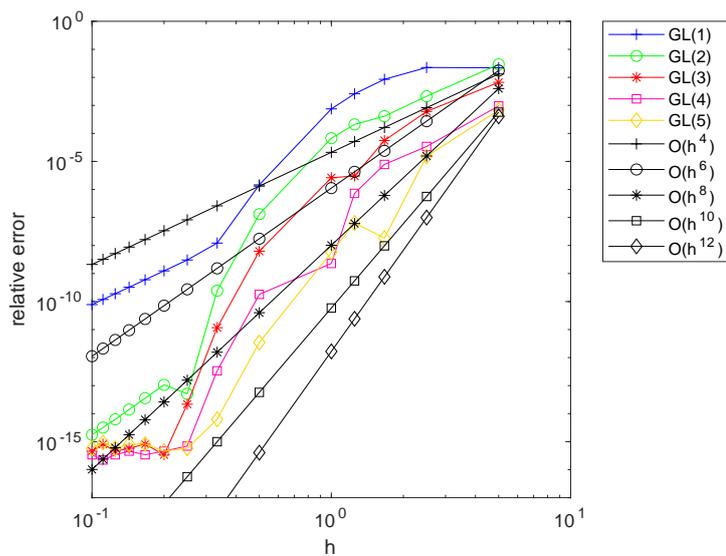


Figure 5.12: Erreur des méthodes de Gauss-Legendre composées pour le calcul de  $\int_{-5}^5 \frac{1}{1+x^2} dx$ , GL(n) correspondant à  $Q_{k,n}^{\text{comp}}$  et  $h = \frac{10}{k}$ .

### Comprendre les ordres de convergence de méthodes numériques

De manière générale, l'ordre de convergence de l'approximation d'une formule, d'un schéma, d'une méthode est donné par une majoration de la norme de la différence entre la solution exacte  $u_{\text{ex}}$  d'un problème et son approximation  $u_h$  où usuellement  $h$  est un paramètre correspondant à la *finesse* de l'approximation numérique (plus la méthode numérique est *précise*, plus  $h$  est proche de 0). On dit alors qu'une méthode numérique est convergente d'**ordre**  $p$  si

$$\|u_{\text{ex}} - u_h\| = \mathcal{O}(h^p).$$

On retrouve cette notion d'ordre dans la résolution numérique d'E.D.O.<sup>1</sup>, dans la résolution numérique d'E.D.P.<sup>2</sup> par différences finies, éléments finis, volumes finis. Ceci permettra de **vérifier/valider** les méthodes numériques implémentées.

Dans le cadre de l'intégration numérique, le Théorème 5.2.1 et plus particulièrement la formule (5.2.2), affirme que si une méthode de quadrature élémentaire  $\mathcal{Q}_n$  est de degré d'exactitude  $p \geq n$  et que  $f \in \mathcal{C}^{p+1}([\alpha, \beta]; \mathbb{R})$  alors

$$\left| \int_{\alpha}^{\beta} f(x) dx - \mathcal{Q}_{k,n}^{\text{comp}}(f, \alpha, \beta) \right| = \mathcal{O}(h^{p+1}).$$

et donc son ordre de convergence est  $p + 1$ .

Il est possible de "retrouver" numériquement l'**ordre** de convergence de méthodes numériques en représentant en échelle logarithmique (en abscisses et ordonnées) la fonction erreur  $h \rightarrow E(h)$  pour chacune des méthodes.

Par exemple, pour la méthode composite de Simpson, on a vu que pour un  $h = (b - a)/N$  donné (i.e. un  $N$  donné)

$$E(h) = \left| \int_a^b f(x) dx - \frac{h}{6} \sum_{i=1}^N (f(x_{i-1}) + 4f(m_i) + f(x_i)) \right| = \mathcal{O}(h^4)$$

Pour  $h$  suffisamment petit, il existe alors  $C > 0$  tel que

$$E(h) = Ch^4.$$

On en déduit que si l'on multiplie  $N$  par 10 alors l'erreur sera divisée par  $10^4$ . En effet, on a

$$E\left(\frac{h}{10}\right) = C\left(\frac{h}{10}\right)^4 = \frac{E(h)}{10^4}.$$

Pour illustrer ceci, on propose le Listing 5.1 où est calculé une approximation de  $\int_0^{\pi/2} \cos(x) dx$  par la méthode composée des trapèzes (degré d'exactitude 1) et par la méthode composée de Simpson (degré d'exactitude 3): on remarque que l'erreur pour la méthode des trapèzes entre  $N = 10$  et  $N = 100$  est bien divisée par  $10^2$  et, pour la méthode de Simpson, elle est bien divisée par  $10^4$ .

```
Listing 5.1: : script Matlab pour illustrer l'ordre des méthodes des Trapèzes et de Simpson
f=@(x) cos(x);
F=@(x) sin(x);
a=0;b=pi/2;
Iex=F(b)-F(a);
I1=QuadTrapeze(f,a,b,10);
I2=QuadTrapeze(f,a,b,100);
fprintf('Erreurs Trapeze: (N=10) %5e - (N=100) %5e\n', abs(I1-Iex), abs(I2-Iex));
I1=QuadSimpson(f,a,b,10);
I2=QuadSimpson(f,a,b,100);
fprintf('Erreurs Simpson: (N=10) %5e - (N=100) %5e\n', abs(I1-Iex), abs(I2-Iex));
```

Output

```
Erreurs Trapeze: (N=10) 2.05701e-03 - (N=100) 2.05618e-05
Erreurs Simpson: (N=10) 2.11547e-07 - (N=100) 2.11393e-11
```

Nous allons maintenant représenter graphiquement ce phénomène. On note tout d'abord que

$$\log(E(h)) = \log(C) + 4 \log(h),$$

et donc, en échelle logarithmique, on va représenter  $\log(h) \rightarrow \log(E(h))$  qui est une droite de pente 4. En Figure 5.13, on représente en échelle logarithmique les différentes erreurs ainsi que les fonctions  $h \rightarrow h^2$  et  $h \rightarrow h^4$ . Le code Matlab/Octave est donné en Listing 5.2.

```
f=@(x) cos(x);
F=@(x) sin(x);
a=0;b=pi/2;
Iex=F(b)-F(a);
LN=25:25:200;
k=1;
for N=LN
    H(k)=(b-a)/N;
```

<sup>1</sup>Equations Différentielles Ordinaires

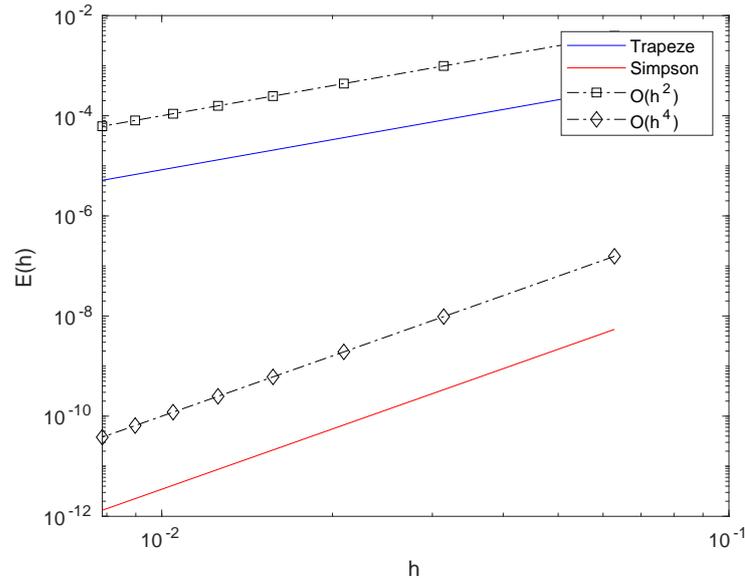
<sup>2</sup>Equations aux Dérivées Partielles

```

E1(k)=abs(QuadTrapeze(f,a,b,N)-Iex);
E2(k)=abs(QuadSimpson(f,a,b,N)-Iex);
k=k+1;
end
loglog(H,E1,'b',H,E2,'r',H,H.^2,'k-.s',...
      H,0.01*H.^4,'k-.d')
xlabel('h');ylabel('E(h)')
legend('Trapeze','Simpson','O(h^2)','O(h^4)')

```

Listing 5.2: Ordre des méthodes composites des trapèzes et de Simpson

Figure 5.13: Ordre des méthodes composites des Trapèzes et de Simpson pour le calcul de  $\int_0^{\pi/2} \cos(x)dx$ 

On donne en Listing 5.3, un exemple pour le calcul approché de  $\int_0^1 x^{3/2}dx$  dont les résultats posent question.

```

Listing 5.3: : Exemple 1, script Matlab/Octave pour illustrer des changements de comportements des ordres
f=@(x) x.^(3/2);
F=@(x) x.^(3/2+1)/(3/2+1);
a=0;b=1;
Iex=F(b)-F(a);
I1=QuadTrapeze(f,a,b,10);
I2=QuadTrapeze(f,a,b,100);
fprintf('Erreurs Trapeze: (N=10) %5e (N=100) %5e\n',abs(I1-Iex),abs(I2-Iex))
I1=QuadSimpson(f,a,b,10);
I2=QuadSimpson(f,a,b,100);
fprintf('Erreurs Simpson: (N=10) %5e (N=100) %5e\n',abs(I1-Iex),abs(I2-Iex))

```

Output

```

Erreurs Trapeze: (N=10) 1.16946e-03 - (N=100) 1.22452e-05
Erreurs Simpson: (N=10) 7.85521e-06 - (N=100) 2.48802e-08

```

On ne retrouve le *bon* ordre pour la méthode de Simpson car la fonction ne vérifie par les hypothèses du Théorème 5.2.1: la dérivée seconde de la fonction  $x \mapsto x^{3/2}$  n'est pas définie en 0.

On donne en Listing 5.4, un autre exemple pour le calcul approché de  $\int_{-1}^1 |x|dx$  dont les résultats posent question.

Listing 5.4: : Exemple 2, script Matlab/Octave pour illustrer des changements de comportements des erreurs

```
f=@(x) abs(x);
a=-1;b=1;
Iex=1;
I1=QuadTrapeze(f,a,b,10);
I2=QuadTrapeze(f,a,b,11);
I3=QuadTrapeze(f,a,b,12);
I4=QuadTrapeze(f,a,b,13);
fprintf('Erreurs Trapeze: (N=10) %5e (N=11) %5e (N=12) %5e (N=13) %5e\n',
        abs(I1-Iex),abs(I2-Iex),abs(I3-Iex),abs(I4-Iex));
I1=QuadSimpson(f,a,b,10);
I2=QuadSimpson(f,a,b,11);
I3=QuadSimpson(f,a,b,12);
I4=QuadSimpson(f,a,b,13);
fprintf('Erreurs Simpson: (N=10) %5e (N=11) %5e (N=12) %5e (N=13) %5e\n',
        abs(I1-Iex),abs(I2-Iex),abs(I3-Iex),abs(I4-Iex));
```

Output

```
Erreurs Trapeze: (N=10) 0.00000e+00 - (N=11) 8.26446e-03 - (N=12) 0.00000e+00 - (N=13) 5.91716e-03
Erreurs Simpson: (N=10) 0.00000e+00 - (N=11) 2.75482e-03 - (N=12) 0.00000e+00 - (N=13) 1.97239e-03
```

Cette fois la fonction n'est pas dérivable en 0! Le Théorème 5.2.1 ne peut s'appliquer. Toutefois, les deux formules donnent un résultat exacte pour  $N$  pair. En effet, dans ce cas le point 0 est un point de discrétisation des méthodes composées et dans ce cas on est ramené à calculer numériquement les deux intégrales  $\int_{-1}^0 (-x)dx$  et  $\int_0^1 (x)dx$  ce qui donnera des résultats exactes puisque les 2 méthodes sont exactes pour les polynômes de degré 1.

### 5.3 Intégrales multiples

On veut approcher, en utilisant la formule de Simpson, l'intégrale

$$I = \int_a^b \int_c^d f(x, y) dy dx$$

Par utilisation de la formule de quadrature de Simpson (5.13) en  $y$  on a

$$g(x) = \int_c^d f(x, y) dy \approx \tilde{g}(x) = \frac{d-c}{6} \left( f(x, c) + 4f(x, \frac{c+d}{2}) + f(x, d) \right).$$

Une nouvelle utilisation de Simpson en  $x$  donne

$$\begin{aligned} I &= \int_a^b g(x) dx \approx \frac{b-a}{6} \left( g(a) + 4g(\frac{a+b}{2}) + g(b) \right) \\ &\approx \frac{b-a}{6} \left( \tilde{g}(a) + 4\tilde{g}(\frac{a+b}{2}) + \tilde{g}(b) \right) \end{aligned}$$

En posant  $\alpha = \frac{a+b}{2}$  et  $\beta = \frac{c+d}{2}$ , on obtient la formule de quadrature de Simpson "2D" :

$$I \approx \frac{b-a}{6} \frac{d-c}{6} \left( \begin{aligned} &f(a, c) + 4f(a, \beta) + f(a, d) \\ &+ 4(f(\alpha, c) + 4f(\alpha, \beta) + f(\alpha, d)) \\ &+ f(b, c) + 4f(b, \beta) + f(b, d) \end{aligned} \right) \tag{5.28}$$

La méthodologie pour obtenir la formule composite de Simpson "2D" est la suivante

- Discrétisation régulière de  $[a, b]$ :  $\forall k \in \llbracket 0, n \rrbracket, x_k = a + kh_x$  avec  $h_x = \frac{b-a}{n}$ .
- Discrétisation régulière de  $[c, d]$ :  $\forall l \in \llbracket 0, m \rrbracket, y_l = c + lh_y$  avec  $h_y = \frac{d-c}{m}$ .
- Relation de Chasles :

$$\int_a^b \int_c^d f(x, y) dy dx = \sum_{k=1}^n \sum_{l=1}^m \int_{x_{k-1}}^{x_k} \int_{y_{l-1}}^{y_l} f(x, y) dy dx.$$

- Formule composite de Simpson "2D" :

$$\begin{aligned} &\int_a^b \int_c^d f(x, y) dy dx \\ &= \\ &\frac{h_x h_y}{36} \sum_{k=1}^n \sum_{l=1}^m \left( \begin{aligned} &f(x_{k-1}, y_{l-1}) + 4f(x_{k-1}, \beta_l) + f(x_{k-1}, y_l) \\ &+ 4(f(\alpha_k, y_{l-1}) + 4f(\alpha_k, \beta_l) + f(\alpha_k, y_l)) \\ &+ f(x_k, y_{l-1}) + 4f(x_k, \beta_l) + f(x_k, y_l) \end{aligned} \right) \end{aligned}$$

avec  $\alpha_k = \frac{x_{k-1} + x_k}{2}$  et  $\beta_l = \frac{y_{l-1} + y_l}{2}$ .

# Chapitre A

## Langage algorithmique

### A.1 Pseudo-langage algorithmique

---

Pour uniformiser l'écriture des algorithmes nous employons un pseudo-langage contenant l'indispensable :

- variables,
- opérateurs (arithmétiques, relationnels, logiques),
- expressions,
- instructions (simples et composées),
- fonctions.

Ce pseudo-langage sera de fait très proche du langage de programmation de Matlab.

#### A.1.1 Données et constantes

---

Une donnée est une valeur introduite par l'utilisateur (par ex. une température, une vitesse, ...). Une constante est un symbole ou un identificateur non modifiable (par ex.  $\pi$ , la constante de gravitation,...).

#### A.1.2 Variables

---

**Définition A.1.1.** *Une variable est un objet dont la valeur est modifiable, qui possède un nom et un type (entier, caractère, réel, complexe, ...). Elle est rangée en mémoire à partir d'une certaine adresse.*

## A.1.3 Opérateurs

## Opérateurs arithmétiques

| Nom             | Symbole | exemple |
|-----------------|---------|---------|
| addition        | +       | $a + b$ |
| soustraction    | -       | $a - b$ |
| opposé          | -       | $-a$    |
| produit         | *       | $a * b$ |
| division        | /       | $a/b$   |
| puissance $a^b$ | ^       | $a^b$   |

Table A.1: Opérateurs arithmétiques

## Opérateurs relationnels

| Nom               | Symbole | exemple  | Commentaires                                           |
|-------------------|---------|----------|--------------------------------------------------------|
| identique         | ==      | $a == b$ | vrai si $a$ et $b$ ont même valeur, faux sinon.        |
| différent         | ~=      | $a ~= b$ | faux si $a$ et $b$ ont même valeur, vrai sinon.        |
| inférieur         | <       | $a < b$  | vrai si $a$ est plus petit que $b$ , faux sinon.       |
| supérieur         | >       | $a > b$  | vrai si $a$ est plus grand que $b$ , faux sinon.       |
| inférieur ou égal | <=      | $a <= b$ | vrai si $a$ est plus petit ou égal à $b$ , faux sinon. |
| supérieur ou égal | >=      | $a >= b$ | vrai si $a$ est plus grand ou égal à $b$ , faux sinon. |

Table A.2: Opérateurs relationnels

## Opérateurs logiques

| Nom      | Symbole | exemple  | Commentaires                                                |
|----------|---------|----------|-------------------------------------------------------------|
| négation | ~       | $\sim a$ | vrai si $a$ est faux (ou nul), faux sinon.                  |
| ou       |         | $a b$    | vrai si $a$ <b>ou</b> $b$ est vrai (non nul), faux sinon.   |
| et       | &       | $a\&b$   | vrai si $a$ <b>et</b> $b$ sont vrais (non nul), faux sinon. |

Table A.3: Opérateurs logiques

## Opérateur d'affectation

| Nom         | Symbole | exemple          | Commentaires                                   |
|-------------|---------|------------------|------------------------------------------------|
| affectation | ←       | $a \leftarrow b$ | On affecte à la variable $a$ le contenu de $b$ |

Table A.4: Opérateurs d'affectation

## A.1.4 Expressions

**Définition A.1.2.** Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

**Exemple.** • Voici un exemple classique d'expression numérique :

$$(b * b - 4 * a * c) / (2 * a).$$

On appelle **opérandes** les identifiants  $a$ ,  $b$  et  $c$ , et les nombres 4 et 2. Les symboles  $*$ ,  $-$  et  $/$  sont les **opérateurs**.

- Voici un exemple classique d'expression booléenne (logique) :

$$(x < 3.14)$$

On teste  $(x < 3.14)$  avec  $x$  une variable numérique réelle. si  $x < 3.14$  alors cette expression renverra la valeur vraie (i.e. 1), faux sinon (i.e. 0).

## A.1.5 Instructions

**Définition A.1.3.** Une **instruction** est un ordre ou un groupe d'ordres qui déclenche l'exécution de certaines actions par l'ordinateur. Il y a deux types d'instructions : simple et structuré.

Les **instructions simples** sont essentiellement des ordres seuls et inconditionnels réalisant l'une des tâches suivantes :

- affectation d'une valeur à une variable.
- appel d'une fonction (procédure, subroutine, ... suivant les langages).

Les **instructions structurées** sont essentiellement :

- les instructions composées, groupe de plusieurs instructions simples,
- les instructions répétitives, permettant l'exécution répétée d'instructions simples, (i.e. boucles «pour», «tant que»)
- les instructions conditionnelles, lesquelles ne sont exécutées que si une certaine condition est respectée (i.e. «si»)

Les exemples qui suivent sont écrits dans un pseudo langage algorithmique mais sont facilement transposable dans la plupart des langages de programmation.

### Instructions simples

Voici un exemple de l'instruction simple d'affectation :

```
1: a ← 3.14 * R
```

On évalue l'expression  $3.14 * R$  et affecte le résultat à la variable  $a$ .

Un autre exemple est donné par l'instruction simple d'affichage :

```
affiche('bonjour')
```

Affiche la chaîne de caractères 'bonjour' à l'écran. Cette instruction fait appel à la fonction `affiche`.

### Instructions composées

### Instructions répétitives, boucle «pour»

---

#### Algorithm A.1 Exemple de boucle «pour»

---

**Données :**  $n$  : un entier.

```
1: S ← 0
2: Pour i ← 1 à n faire
3:   S ← S + cos(i2)
4: Fin
```

---

**Instruction répétitive, boucle «tant que»****Algorithm A.2** Exemple de boucle «tant que»

---

```

1:  $i \leftarrow 0, x \leftarrow 1$ 
2: Tantque  $i < 1000$  faire
3:    $x \leftarrow x + i * i$ 
4:    $i \leftarrow i + 1$ 
5: Fin

```

---

**Instruction répétitive, boucle «répéter ...jusqu'à»****Algorithm A.3** Exemple de boucle «répéter ...jusqu'à»

---

```

1:  $i \leftarrow 0, x \leftarrow 1$ 
2: Répéter
3:    $x \leftarrow x + i * i$ 
4:    $i \leftarrow i + 1$ 
5: jusqu'à  $i \geq 1000$ 

```

---

**Instructions conditionnelles «si»****Algorithm A.4** Exemple d'instructions conditionnelle «si»

---

**Données :** *age* : un réel.

```

1: Si  $age \geq 18$  alors
2:   affiche('majeur')
3: Sinon Si  $age \geq 0$  alors
4:   affiche('mineur')
5: Sinon
6:   affiche('en devenir')
7: Fin

```

---

**A.1.6 Fonctions**

Les fonctions permettent

- d'automatiser certaines tâches répétitives au sein d'un même programme,
- d'ajouter à la clarté d'un programme,
- l'utilisation de portion de code dans un autre programme,
- ...

**Fonctions prédéfinies**

Pour faciliter leur usage, tous les langages de programmation possèdent des fonctions prédéfinies. On pourra donc supposer que dans notre langage algorithmique un grand nombre de fonctions soient prédéfinies : par exemple, les fonctions mathématiques  $\sin$ ,  $\cos$ ,  $\exp$ ,  $\text{abs}$ ,  $\dots$  (pour ne citer qu'elles)

**Syntaxe**

On utilise la syntaxe suivante pour la définition d'une fonction

```

Fonction [args1, ..., argsn] ← NomFonction( arge1, ..., argem )
    instructions
Fin

```

La fonction se nomme **NomFonction**. Elle admet comme paramètres d'entrée (données) les  $m$  arguments  $arge_1, \dots, arge_m$  et comme paramètres de sortie (résultats) les  $n$  arguments  $args_1, \dots, args_n$ . Ces derniers doivent être déterminés dans le corps de la fonction (partie instructions).

Dans le cas où la fonction n'admet qu'un seul paramètre de sortie, l'écriture se simplifie :

```

Fonction args ← NomFonction( arge1, ..., argem )
    instructions
Fin

```

### Ecrire ses propres fonctions

Pour écrire une fonction «propre», il faut tout d'abord déterminer exactement ce que devra faire cette fonction.

Puis, il faut pouvoir répondre à quelques questions :

- Quelles sont les données (avec leurs limitations)?
- Que doit-on calculer ?

Et, ensuite il faut la **commenter** : expliquer son usage, type des paramètres, ....

### Exemple : résolution d'une équation du premier degré

Nous voulons écrire une fonction calculant la solution de l'équation

$$ax + b = 0,$$

où nous supposons que  $a \in \mathbb{R}^*$  et  $b \in \mathbb{R}$ . La solution de ce problème est donc

$$x = -\frac{b}{a}.$$

Les données de cette fonction sont  $a \in \mathbb{R}^*$  et  $b \in \mathbb{R}$ . Elle doit retourner  $x = -\frac{b}{a}$  solution de  $ax + b = 0$ .

---

**Algorithm A.5** Exemple de fonction : Résolution de l'équation du premier degré  $ax + b = 0$ .

---

**Données :**  $a$  : nombre réel différent de 0  
 $b$  : nombre réel.

**Résultat :**  $x$  : un réel.

```

1: Fonction x ← REPD( a, b )
2:   x ← -b/a
3: Fin

```

---

**Remarque A.1.1.** Cette fonction est très simple, toutefois pour ne pas «alourdir» le code nous n'avons pas vérifié la validité des données fournies.

#### EXERCICE 1.1.1

Écrire un algorithme permettant de valider cette fonction.

**Exemple : résolution d'une équation du second degré**

Nous cherchons les solutions réelles de l'équation

$$ax^2 + bx + c = 0, \quad (\text{A.1})$$

où nous supposons que  $a \in \mathbb{R}^*$ ,  $b \in \mathbb{R}$  et  $c \in \mathbb{R}$  sont donnés.

Mathématiquement, l'étude des solutions réelles de cette équation nous amène à envisager trois cas suivant les valeurs du discriminant  $\Delta = b^2 - 4ac$

- si  $\Delta < 0$  alors les deux solutions sont complexes,
- si  $\Delta = 0$  alors la solution est  $x = -\frac{b}{2a}$ ,
- si  $\Delta > 0$  alors les deux solutions sont  $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$  et  $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$ .

**EXERCICE 1.1.2**

- Ecrire la fonction **discriminant** permettant de calculer le discriminant de l'équation (A.1).
- Ecrire la fonction **RESD** permettant de résoudre l'équation (A.1) en utilisant la fonction **discriminant**.
- Ecrire un programme permettant de valider ces deux fonctions.

**EXERCICE 1.1.3**

Même question que précédemment dans le cas complexe (solutions et coefficients).

**A.1.7 Tableaux**

Pour simplifier la description de tableaux (données et/ou résultats d'un algorithme, par ex.), on va utiliser des notations mathématiques.

- Identification les tableaux 1D à des vecteurs:
  - $\mathbf{x}$  tableau de réels à une dimension et de longueur  $n \in \mathbb{N}^* \Rightarrow \mathbf{x} \in \mathbb{R}^n$ ,
  - $\mathbf{x}$  tableau d'entier à une dimension et de longueur  $n \in \mathbb{N}^* \Rightarrow \mathbf{x} \in \mathbb{N}^n$ ,
  - ...

**Choix indicage: l'indice du premier élément d'un tableau/vecteur est 1**

**Exemple.** Si dans un problème, un ensemble de réels  $(x_i)_{i=1}^n$  est utilisé, au niveau algorithmique son stockage se fera dans un tableau de  $n$  réels nommé  $\mathbf{X}$ , par exemple, et le lien entre la notation mathématique et la notation algorithmique devra être écrit:

$$\mathbf{X} \in \mathbb{R}^n \text{ tel que } \forall i \in \llbracket 1, n \rrbracket, \mathbf{X}(i) = x_i.$$

Ce lien est représenté graphiquement en Figure A.1.

|              |                 |                 |                 |         |                   |                   |                 |
|--------------|-----------------|-----------------|-----------------|---------|-------------------|-------------------|-----------------|
| $\mathbf{X}$ | $x_1$           | $x_2$           | $x_3$           | $\dots$ | $x_{n-2}$         | $x_{n-1}$         | $x_n$           |
|              | $\mathbf{X}(1)$ | $\mathbf{X}(2)$ | $\mathbf{X}(3)$ |         | $\mathbf{X}(n-2)$ | $\mathbf{X}(n-1)$ | $\mathbf{X}(n)$ |

Figure A.1: Représentation mémoire de la variable algorithmique  $\mathbf{X}$  contenant l'ensemble des réels  $(x_i)_{i=1}^n$ .

**Exemple.** Si dans un problème, un ensemble de réels  $(t_i)_{i=0}^n$  est utilisé, au niveau algorithmique son stockage se fera dans un tableau de  $(n + 1)$  réels nommé  $\mathbf{t}$ , par exemple, et le lien entre la notation mathématique et la notation algorithmique devra être écrit:

$$\mathbf{t} \in \mathbb{R}^{n+1} \text{ tel que } \forall i \in \llbracket 1, n + 1 \rrbracket, \mathbf{t}(i) = t_{i-1}$$

ou encore

$$\mathbf{t} \in \mathbb{R}^{n+1} \text{ tel que } \forall i \in \llbracket 0, n \rrbracket, \mathbf{t}(i + 1) = t_i.$$

Ce lien est représenté graphiquement en Figure A.1.

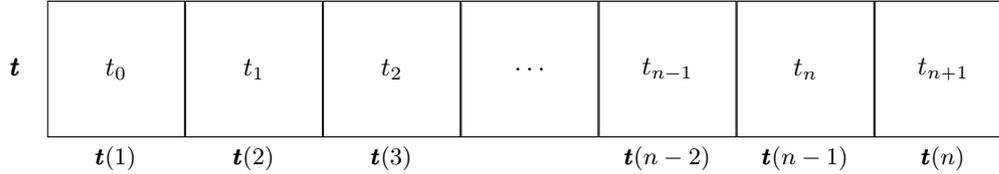


Figure A.2: Représentation mémoire de la variable algorithmique  $\mathbf{t}$  contenant l'ensemble des réels  $(t_i)_{i=0}^n$ .

- Identification des tableaux 2D à des matrices:

- $\mathbb{T}$  tableau de réels à  $m$  lignes et  $n$  colonnes  $\Rightarrow \mathbb{T} \in \mathcal{M}_{m,n}(\mathbb{R})$ ,
- $\mathbb{T}$  tableau de complexes à  $m$  lignes et  $n$  colonnes  $\Rightarrow \mathbb{T} \in \mathcal{M}_{m,n}(\mathbb{C})$ ,
- ...

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket, \mathbb{T}(i, j) \text{ (notation algo.)} \iff \mathbb{T}_{i,j} \text{ (notation math.)}$$

Lorsqu'un tableau est une donnée d'entrée d'une fonction algorithmique, ses dimensions sont supposées données de manière implicite (une seule dimension si tableau 1D!): il n'est pas utile de les passer en paramètre à la fonction.

**Exemple.** Soit  $(u_i)_{i=0}^n$ , on souhaite écrire une fonction `sum` permettant de calculer  $\sum_{i=0}^n u_i = 0^n u_i$ . Dans ce cas le tableau  $\mathbf{U} \in \mathbb{R}^{n+1}$  est la donnée d'entrée de cette fonction et la variable  $n$  est implicitement donnée et donc supposée connue dans la fonction. Toutefois, pour se rassurer, on peut écrire en tête du corps de la fonction

$$n \leftarrow \text{length}(\mathbf{U}) - 1$$

La fonction `sum` est écrite en Algorithme A.6.

---

**Algorithme A.6** Fonction `sum` permettant de calculer  $\sum_{i=0}^n u_i$ .

---

**Données :**  $\mathbf{U}$  : dans  $\mathbb{R}^{n+1}$  avec  $\mathbf{U}(i + 1) = u_i, \forall i \in \llbracket 0, n \rrbracket$ .

**Résultat :**  $S$  : le réel  $S = \sum_{i=0}^n u_i$ .

- 1: **Fonction**  $S \leftarrow \text{sum}(\mathbf{U})$
  - 2:  $n \leftarrow \text{length}(\mathbf{U}) - 1$  ▷ pas obligatoire
  - 3:  $S \leftarrow 0$
  - 4: **Pour**  $i \leftarrow 0$  à  $n$  **faire**
  - 5:      $S \leftarrow S + \mathbf{U}(i + 1)$
  - 6: **Fin**
  - 7: **Fin**
- 

**Exemple.** Soit  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ . On souhaite écrire une fonction `sumrows` permettant de calculer le vecteur  $\mathbf{s} \in \mathbb{R}^m$  tel que  $\mathbf{s}_i = \sum_{j=1}^n \mathbb{A}_{i,j}$ . Dans ce cas  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  est la donnée d'entrée de cette fonction et les variables  $m$  et  $n$  sont implicitement données. Cette fonction retournera  $\mathbf{s} \in \mathbb{R}^m$  où la dimension  $m$  sera aussi implicite (puisque récupérable à partir de  $\mathbf{s}$ ).

La fonction `sumrows` est écrite en Algorithme A.7.

---

**Algorithm A.7** Fonction `sumrows`


---

**Données :**  $\mathbb{A}$  : dans  $\mathcal{M}_{m,n}(\mathbb{R})$

**Résultat :**  $\mathbf{s}$  : dans  $\mathbb{R}^m$

```

1: Fonction  $\mathbf{s} \leftarrow \text{sumrows}(\mathbb{A})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire ▷  $m$  implicitement connu
3:      $\mathbf{s}(i) \leftarrow 0$ 
4:     Pour  $j \leftarrow 1$  à  $n$  faire ▷  $n$  implicitement connu
5:        $\mathbf{s}(i) \leftarrow \mathbf{s}(i) + \mathbb{A}(i, j)$ 
6:     Fin
7:   Fin
8: Fin

```

---

On pourrait ajouter en tête du corps de la fonction

$$m \leftarrow \text{nb de lignes de } \mathbb{A}, n \leftarrow \text{nb de colonnes de } \mathbb{A}$$

ou, avec une fonction `size` similaire à celle de Matlab/Octave

$$[m, n] \leftarrow \text{size}(\mathbb{A}).$$

## A.2 Méthodologie d'élaboration d'un algorithme

### A.2.1 Description du problème

- Spécification d'un ensemble de données  
Origine : énoncé, hypothèses, sources externes, ...
- Spécification d'un ensemble de buts à atteindre  
Origine : résultats, opérations à effectuer, ...
- Spécification des contraintes

### A.2.2 Recherche d'une méthode de résolution

- Clarifier l'énoncé.
- Simplifier le problème.
- Ne pas chercher à le traiter directement dans sa globalité.
- S'assurer que le problème est soluble (sinon problème d'indécidabilité!)
- Recherche d'une stratégie de construction de l'algorithme
- Décomposer le problème en sous problèmes partiels plus simples : raffinement.
- Effectuer des raffinements successifs.
- Le niveau de raffinement le plus élémentaire est celui des instructions.

### A.2.3 Réalisation d'un algorithme

Il doit être conçu indépendamment du langage de programmation et du système informatique (sauf cas très particulier)

- L'algorithme doit être exécuté en un nombre fini d'opérations.
- L'algorithme doit être spécifié clairement, sans la moindre ambiguïté.

- Le type de données doit être précisé.
- L'algorithme doit fournir au moins un résultat.
- L'algorithme doit être effectif : toutes les opérations doivent pouvoir être simulées par un homme en temps fini.

Pour écrire un algorithme détaillé, il faut tout d'abord savoir répondre à quelques questions :

- Que doit-il faire ? (i.e. Quel problème est-il censé résoudre?)
- Quelles sont les données nécessaires à la résolution de ce problème?
- Comment résoudre ce problème «à la main» (sur papier)?

Si l'on ne sait pas répondre à l'une de ces questions, l'écriture de l'algorithme est fortement compromise.

## A.2.4 Exercices

### EXERCICE 1.2.1 Algorithme pour une somme

Écrire un algorithme permettant de calculer

$$S(x) = \sum_{k=1}^n k \sin(2 * k * x)$$

**Correction 1.2.1** L'énoncé de cet exercice est imprécis. On choisit alors  $x \in \mathbb{R}$  et  $n \in \mathbb{N}$  pour rendre possible le calcul. Le problème est donc de calculer

$$\sum_{k=1}^n k \sin(2kx).$$

Toutefois, on aurait pu choisir  $x \in \mathbb{C}$  ou encore un tout autre problème :

$$\text{Trouver } x \in \mathbb{R} \text{ tel que } S(x) = \sum_{k=1}^n k \sin(2kx)$$

où  $n \in \mathbb{N}$  et  $S$ , fonction de  $\mathbb{R}$  à valeurs réelles, sont les données!

---

**Algorithme A.8** Calcul de  $S = \sum_{k=1}^n k \sin(2kx)$

---

**Données :**  $x$  : nombre réel,  
 $n$  : nombre entier.

**Résultat :**  $S$  : un réel.

- 1:  $S \leftarrow 0$
  - 2: **Pour**  $k \leftarrow 1$  à  $n$  **faire**
  - 3:  $S \leftarrow S + k * \sin(2 * k * x)$
  - 4: **Fin**
- 

~~~~~ Fin correction ~~~~~

### EXERCICE 1.2.2 Algorithme pour un produit

Écrire un algorithme permettant de calculer

$$P(z) = \prod_{n=1}^k \sin(2 * k * z/n)^k$$

**Correction 1.2.2** L'énoncé de cet exercice est imprécis. On choisit alors  $z \in \mathbb{R}$  et  $k \in \mathbb{N}$  pour rendre possible le calcul.

---

**Algorithme A.9** Calcul de  $P = \prod_{n=1}^k \sin(2kz/n)^k$

---

**Données :**  $z$  : nombre réel,  
 $k$  : nombre entier.

**Résultat :**  $P$  : un réel.

- 1:  $P \leftarrow 1$
  - 2: **Pour**  $n \leftarrow 1$  à  $k$  **faire**
  - 3:  $P \leftarrow P * \sin(2 * k * z/n)^k$
  - 4: **Fin**
- 

~~~~~Fin correction~~~~~

### EXERCICE 1.2.3 Série de Fourier

Soit la série de Fourier

$$x(t) = \frac{4A}{\pi} \left\{ \cos \omega t - \frac{1}{3} \cos 3\omega t + \frac{1}{5} \cos 5\omega t - \frac{1}{7} \cos 7\omega t + \dots \right\}.$$

Ecrire la fonction SFT permettant de calculer  $x_n(t)$ .

**Correction 1.2.3** Nous devons écrire la fonction permettant de calculer

$$x_n(t) = \frac{4A}{\pi} \sum_{k=1}^n (-1)^{k+1} \frac{1}{2k-1} \cos((2k-1)\omega t)$$

Les données de la fonction sont  $A \in \mathbb{R}$ ,  $\omega \in \mathbb{R}$ ,  $n \in \mathbb{N}^*$  et  $t \in \mathbb{R}$ .

Grâce à ces renseignements nous pouvons déjà écrire l'entête de la fonction :

---

**Algorithme A.10** En-tête de la fonction SFT retournant valeur de la série de Fourier en  $t$  tronquée au  $n$  premiers termes de l'exercice 1.2.3.

---

**Données :**  $t$  : nombre réel,  
 $n$  : nombre entier strictement positif  
 $A, \omega$  : deux nombres réels.

**Résultat :**  $x$  : un réel.

- 1: **Fonction**  $x \leftarrow \text{SFT}(t, n, A, \omega)$
  - 2: ...
  - 3: **Fin**
- 

Maintenant nous pouvons écrire progressivement l'algorithme pour aboutir au final à une version ne contenant que des opérations élémentaires.

**Algorithme A.11**  $\mathcal{R}_0$

$$1: x \leftarrow \frac{4A}{\pi} \sum_{k=1}^n \left( (-1)^{k+1} \frac{1}{2k-1} \times \cos((2k-1)\omega t) \right)$$

**Algorithme A.11**  $\mathcal{R}_1$

$$\left. \begin{array}{l} 1: S \leftarrow \sum_{k=1}^n (-1)^{k+1} \frac{1}{2k-1} \cos((2k-1)\omega t) \\ 2: x \leftarrow \frac{4A}{\pi} S \end{array} \right\}$$

**Algorithme A.11**  $\mathcal{R}_1$ 

```

1:  $S \leftarrow \sum_{k=1}^n (-1)^{k+1} \frac{1}{2k-1} \cos((2k-1)\omega t)$ 
2:  $x \leftarrow \frac{4A}{\pi} S$ 

```

**Algorithme A.11**  $\mathcal{R}_2$ 

```

1:  $S \leftarrow 0$ 
2: Pour  $k = 1$  à  $n$  faire
3:    $S \leftarrow S + (-1)^{k+1} \frac{1}{2k-1} * \cos((2k-1)\omega t)$ 
4: Fin
5:  $x \leftarrow \frac{4A}{\pi} S$ 

```

Finalement la fonction est

**Algorithme A.11** Fonction SFT retournant la valeur de la série de Fourier en  $t$  tronquée au  $n$  premiers termes de l'exercice 1.2.3.

**Données :**  $t$  : nombre réel,  
 $n$  : nombre entier strictement positif  
 $A, \omega$  : deux nombres réels.

**Résultat :**  $x$  : un réel.

```

1: Fonction  $x \leftarrow \text{SFT}(t, n, A, \omega)$ 
2:    $S \leftarrow 0$ 
3:   Pour  $k = 1$  à  $n$  faire
4:      $S \leftarrow S + ((-1)^{(k+1)} * \cos((2 * k - 1) * \omega * t) / (2 * k - 1))$ 
5:   Fin
6:    $x \leftarrow 4 * A * S / \pi$ 
7: Fin

```

~~~~~ Fin correction ~~~~~

**EXERCICE 1.2.4**

Reprendre les trois exercices précédents en utilisant les boucles «tant que».

## A.3 Principes de «bonne» programmation pour attaquer de «gros» problèmes

Tous les exemples vus sont assez courts. Cependant, il peut arriver que l'on ait des programmes plus longs à écrire (milliers de lignes, voir des dizaines de milliers de lignes). Dans l'industrie, il arrive que des équipes produisent des codes de millions de lignes, dont certains mettent en jeux des vies humaines (contrôler un avion de ligne, une centrale nucléaire, ...). Le problème est évidemment d'écrire des programmes sûrs. Or, *un programme à 100% sûr, cela n'existe pas!* Cependant, plus un programme est simple, moins le risque d'erreur est grand : c'est sur cette remarque de bon sens que se basent les «bonnes» méthodes. Ainsi :

*Tout problème compliqué doit être découpé en sous-problèmes plus simples*

Il s'agit, lorsqu'on a un problème  $P$  à résoudre, de l'analyser et de le décomposer en un ensemble de problèmes  $P_1, P_2, P_3, \dots$  plus simples. Puis,  $P_1$ , est lui-même analysé et décomposé en  $P_{11}, P_{12}, \dots$ , et  $P_2$  en  $P_{21}, P_{22}$ , etc. On poursuit cette analyse jusqu'à ce qu'on n'ait plus que des problèmes élémentaires à résoudre. Chacun de ces problèmes élémentaires est donc traité séparément dans un module, c'est à dire un morceau de programme relativement indépendant du reste. Chaque module sera *testé et validé* séparément dans la mesure du possible et naturellement *largement documenté*. Enfin ces modules élémentaires sont assemblés en modules de plus en plus complexes, jusqu'à remonter au problème initiale. A chaque niveau, il sera important de bien réaliser les phases de test, validation et documentation des modules.

Dans ce document, on s'évertue à écrire des algorithmes!  
**Ceux-ci ne seront pas optimisés<sup>a</sup>!**

<sup>a</sup>améliorés pour minimiser le nombre d'opérations élémentaires, l'occupation mémoire, ..

## A.4 Quelques simplifications d'écriture

Quelques exercices d'algorithmique numérique avec correction sont proposés en Annexe B.3.1 pour illustrer les simplifications d'écriture qui sont décrites dans cette section.

Au niveau algorithmique, un tableau de réels à une dimension et de longueur  $n \in \mathbb{N}^*$  peut être identifié par un vecteur de  $\mathbb{R}^n$ , tandis qu'un tableau de réels à deux dimensions,  $m$  lignes et  $n$  colonnes, peut être identifié par une matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

Lorsqu'un vecteur ou une matrice sera passé en paramètre d'entrée, il est inutile (voir "dangereux") de fournir leurs dimensions en paramètre d'entrée: elles sont implicitement connues! Dans la plupart des langages (avec bibliothèques adéquates), il existe des types ou objets `vector` ou `matrix` ainsi que des fonctions/méthodes/propriétés permettant de "récupérer" leurs dimensions. Par exemple, en Matlab/Octave, la fonction `size` pourra être utilisée.

Pour simplifier l'écriture des algorithmes numériques, il sera possible d'utiliser les "raccourcis" suivant pour accéder aux éléments d'une matrice  $A \in \mathcal{M}_{m,n}(\mathbb{R})$  (ou tableau à  $m$  lignes et  $n$  colonnes)

- $A(:, j)$  correspond au  $j$ -ème vecteur colonne de  $A$ ,  $j \in \llbracket 1, n \rrbracket$ . Si on écrit  $v \leftarrow A(:, j)$  alors l'accès aux éléments de  $v \in \mathbb{R}^m$  (ou tableau à  $m$  éléments) s'effectue avec la commande  $v(i)$  avec  $i \in \llbracket 1, m \rrbracket$ . De plus, au niveau algorithmique, si  $w$  est un vecteur colonne ou ligne de dimension  $m$ , alors  $A(:, j) \leftarrow w$  est autorisé et correspond mathématiquement à  $A_{i,j} = w_i, \forall i \in \llbracket 1, m \rrbracket$ .
- $A(i, :)$  correspond au  $i$ -ème vecteur ligne de  $A$ ,  $i \in \llbracket 1, m \rrbracket$ . Si on écrit  $u \leftarrow A(i, :)$  alors l'accès aux éléments de  $u \in \mathbb{R}^n$  (ou tableau à  $n$  éléments) s'effectue avec la commande  $u(j)$  avec  $j \in \llbracket 1, n \rrbracket$ . De plus, au niveau algorithmique, si  $z$  est un vecteur colonne ou ligne de dimension  $n$ , alors  $A(i, :) \leftarrow z$  est autorisé et correspond mathématiquement à  $A_{i,j} = z_j, \forall j \in \llbracket 1, n \rrbracket$ .

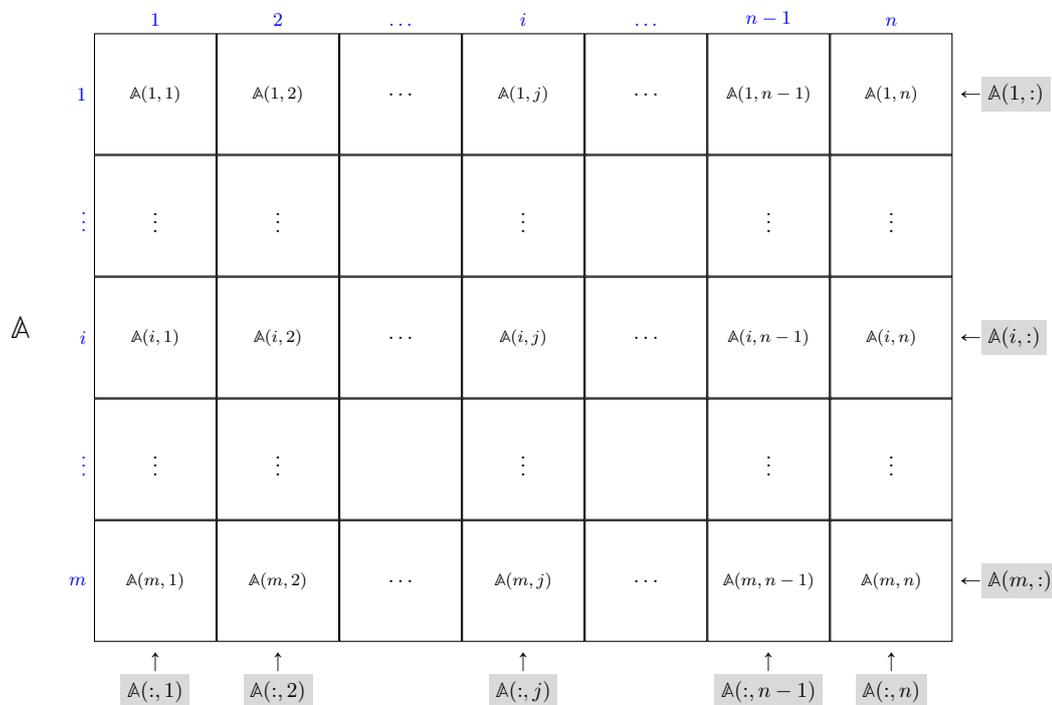


Figure A.3: Version simplifiée d'accès aux lignes et colonnes d'une matrice  $A \in \mathcal{M}_{m,n}(\mathbb{R})$ .

Avec les fonctions écrites dans l'Exercice 2.3.4, p.237, on a

| <i>Description</i>  | <i>Avec fonction</i>  | <i>Instruction simplifiée</i>              |
|---|---|--|
| Obtenir la $j$ -ème colonne d'une matrice $\mathbb{A}$ avec $j \in \llbracket 1, n \rrbracket$ , math. noté $\mathbb{A}_{:,j}$                | $\mathbf{u} \leftarrow \text{getMatCol}(\mathbb{A}, j)$             | $\mathbf{u} \leftarrow \mathbb{A}(:, j)$   |
| Obtenir la $i$ -ème ligne d'une matrice $\mathbb{A}$ avec $i \in \llbracket 1, m \rrbracket$ , math. noté $\mathbb{A}_{i,j}$                  | $\mathbf{v} \leftarrow \text{getMatRow}(\mathbb{A}, i)$             | $\mathbf{v} \leftarrow \mathbb{A}(i, :)$   |
| Remplacer la $j$ -ème colonne d'une matrice $\mathbb{A}$ par un vecteur $\mathbf{u} \in \mathbb{R}^m$ avec $j \in \llbracket 1, n \rrbracket$ | $\mathbb{A} \leftarrow \text{setMatCol}(\mathbb{A}, \mathbf{u}, j)$ | $\mathbb{A}(:, j) \leftarrow \mathbf{u}$   |
| Remplacer la $i$ -ème ligne d'une matrice $\mathbb{A}$ par un vecteur $\mathbf{v} \in \mathbb{R}^n$ avec $i \in \llbracket 1, m \rrbracket$   | $\mathbb{A} \leftarrow \text{setMatRow}(\mathbb{A}, \mathbf{v}, j)$ | $\mathbb{A}(i, :) \leftarrow \mathbf{v}$ . |

Table A.5: Commandes algorithmiques permettant l'accès à l'intégralité d'une ligne ou d'une colonne d'un tableau à  $m$  lignes et  $n$  colonne de noté  $\mathbb{A}$  que l'on identifiera à une matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .



# Chapitre B

## Annexes

### B.1 Analyse : rappels

#### B.1.1 En vrac

**Théorème B.1.1** (Théorème de Rolle). Soient  $a, b$  deux réels,  $a < b$ , et,  $f : [a, b] \rightarrow \mathbb{R}$ . On suppose que

- $f$  est continue sur  $[a, b]$ ,
- $f$  est dérivable sur  $]a, b[$ ,
- $f(a) = f(b)$ .

Alors il existe  $c \in ]a, b[$  tel que  $f'(c) = 0$ .

**Théorème B.1.2** (Théorème de Bolzano ou des valeurs intermédiaires). Soit  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  une application continue. Si  $f(a)$  et  $f(b)$  ne sont pas de même signe (i.e.  $f(a)f(b) < 0$ ) alors il existe au moins  $c \in ]a, b[$  tel que  $f(c) = 0$ .

**Théorème B.1.3** (Théorème des accroissements finis). Soient  $a$  et  $b$  deux réels,  $a < b$  et  $f$  une fonction continue sur l'intervalle fermé  $[a, b]$ , dérivable sur l'intervalle ouvert  $]a, b[$ . Alors il existe  $\xi \in ]a, b[$  tel que

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}.$$

**Proposition B.1.1** (Formule de Taylor-Lagrange d'ordre  $n$ ). Soit  $n \in \mathbb{N}$  et  $f \in \mathcal{C}^n([a, b])$  dont la dérivée  $n$ -ième est dérivable sur  $]a, b[$ . Alors

- pour tout  $x, y$  dans  $[a, b]$ ,  $x \neq y$ , il existe  $\xi \in ]\min(x, y), \max(x, y)[$  tel que

$$f(x) = f(y) + \sum_{k=1}^n \frac{(x-y)^k}{k!} f^{(k)}(y) + \frac{(x-y)^{n+1}}{(n+1)!} f^{(n+1)}(\xi) \quad (\text{B.1})$$

- $\forall t \in [a, b]$ ,  $\forall h \in \mathbb{R}^*$  vérifiant  $(t+h) \in [a, b]$ , il existe  $\xi \in ]\min(t, t+h), \max(t, t+h)[$  tel quel

$$f(t+h) = f(t) + \sum_{k=1}^n \frac{h^k}{k!} f^{(k)}(t) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(\xi) \quad (\text{B.2})$$

**Définition B.1.1.** Soient  $f$  et  $g$  deux fonctions définies au voisinage de  $a \in \mathbb{R}$ . On dit que  $f$  est **dominée** par  $g$  au voisinage de  $a$ , si, au voisinage de  $a$ , il existe une fonction  $\theta$  bornée telle que  $f = \theta g$ . Dans ce cas, on dit aussi que  $f$  **se comporte comme un grand  $\mathcal{O}$  de  $g$**  au voisinage de  $a$  et on note alors  $f \stackrel{a}{=} \mathcal{O}(g)$ .

**Définition B.1.2.** Soit  $f$  une fonction définie au voisinage de  $0$ . On dit que  $f(h)$  **se comporte comme un grand  $\mathcal{O}$  de  $h^p$**  (au voisinage de  $0$ ) si  $f$  est dominée par  $h \mapsto h^p$  au voisinage de  $0$  et on note alors  $f(h) = \mathcal{O}(h^p)$ .

**Proposition B.1.2** (Formule de Taylor-Landau d'ordre  $n$ ). Soit  $n \in \mathbb{N}$  et  $f \in \mathcal{C}^{n+1}([a, b])$ , alors  $\forall t \in [a, b]$ ,  $\forall h \in \mathbb{R}^*$  vérifiant  $(t+h) \in [a, b]$ , il existe  $\xi \in ]\min(t, t+h), \max(t, t+h)[$  tel quel

$$f(t+h) = f(t) + \sum_{k=1}^n \frac{h^k}{k!} f^{(k)}(t) + \mathcal{O}(h^{n+1}) \quad (\text{B.3})$$

**Corollaire B.1.1** (Théorème de la bijection). Si  $f$  est une fonction continue et strictement monotone sur un intervalle  $[a, b]$  et à valeurs réelles, alors elle constitue une bijection entre  $[a, b]$  et l'intervalle fermé dont les bornes sont  $f(a)$  et  $f(b)$ .

**Proposition B.1.3.** Soit  $f$  est une fonction bijective continue d'un intervalle ouvert  $I \subset \mathbb{R}$  sur un intervalle ouvert  $J \subset \mathbb{R}$ . Si  $f$  est dérivable en  $\alpha \in I$  et que  $f'(\alpha) \neq 0$  alors sa réciproque  $f^{-1}$  est dérivable en  $\beta = f(\alpha) \in J$  et

$$(f^{-1})'(\beta) = \frac{1}{f'(\alpha)} \quad \text{ou encore} \quad (f^{-1})'(\beta) = \frac{1}{f'(f^{-1}(\beta))}$$

## B.1.2 Espace métrique<sup>2</sup>

**Définition B.1.3** (Distance sur un ensemble). On appelle **distance** sur un ensemble  $E$ , une application  $d$  de  $E^2$  dans  $\mathbb{R}_+$  telle que pour tout  $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in E^3$  on a

- **symétrie** :  $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ ,
- **séparation** :  $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$ ,
- **inégalité triangulaire** :  $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$

Voici quelques exemples de distances:

- $d(x, y) = |x - y|$  dans  $\mathbb{R}$ ,  $\mathbb{C}$ ,  $\mathbb{Z}$  ou  $\mathbb{Q}$

<sup>2</sup>En grande partie extrait du site bibmat

- $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$  dans  $\mathbb{R}^n$ , où  $\|\cdot\|$  est l'une quelconque des normes habituelles.

**Définition B.1.4** (Espace métrique). *Un ensemble  $E$  muni d'une distance  $d$  est appelé **espace métrique** et on le note  $(E, d)$ .*

Soient  $(E, d)$  un espace métrique,  $\mathbf{a} \in E$  et  $r \in \mathbb{R}_+$ . On appelle

- **boule ouverte** de centre  $\mathbf{a}$  et de rayon  $r$  l'ensemble

$$\mathcal{B}(\mathbf{a}, r) = \{\mathbf{x} \in E; d(\mathbf{x}, \mathbf{a}) < r\},$$

- **boule fermée** de centre  $\mathbf{a}$  et de rayon  $r$  l'ensemble

$$\overline{\mathcal{B}(\mathbf{a}, r)} = \{\mathbf{x} \in E; d(\mathbf{x}, \mathbf{a}) \leq r\},$$

- **sphere** de centre  $\mathbf{a}$  et de rayon  $r$  l'ensemble

$$\mathcal{S}(\mathbf{a}, r) = \{\mathbf{x} \in E; d(\mathbf{x}, \mathbf{a}) = r\},$$

Une partie  $A \subset E$  est dite **bornée** si

$$\exists \mathbf{x} \in E, \exists R \in \mathbb{R}_+^*, A \subset \mathcal{B}(\mathbf{x}, R).$$

## Suites

**Définition B.1.5** (Suite convergente). *Soient  $(E, d)$  un **espace métrique** et  $(\mathbf{u}^{[k]})_{k \in \mathbb{N}}$  une suite d'éléments de  $E$ . On dit que la suite  $(\mathbf{u}^{[k]})_{k \in \mathbb{N}}$  converge si*

$$\exists \boldsymbol{\alpha} \in E, \forall \epsilon > 0, \exists N \in \mathbb{N}, \forall k > N, d(\mathbf{u}^{[k]}, \boldsymbol{\alpha}) < \epsilon. \quad (\text{B.4})$$

*Dans ce cas on dit que la suite  $(\mathbf{u}^{[k]})_{k \in \mathbb{N}}$  converge vers  $\boldsymbol{\alpha} \in E$ .*

Une suite qui ne converge vers aucun  $\boldsymbol{\alpha} \in E$  est dite **divergente** et vérifie

$$\forall \boldsymbol{\alpha} \in E, \exists \epsilon > 0, \forall N \in \mathbb{N}, \exists k > N, d(\mathbf{u}^{[k]}, \boldsymbol{\alpha}) \geq \epsilon.$$

Si la suite  $(\mathbf{u}^{[k]})$  de  $E$  converge vers  $\boldsymbol{\alpha}$  (nécessairement unique dans  $E$ ) alors on dit que  $\boldsymbol{\alpha}$  est la **limite** de  $(\mathbf{u}^{[k]})$  et on note

$$\lim_{k \rightarrow +\infty} \mathbf{u}^{[k]} = \boldsymbol{\alpha} \quad \text{ou} \quad \mathbf{u}^{[k]} \underset{k \rightarrow +\infty}{=} \boldsymbol{\alpha}.$$

## Ouverts et fermés

Soit  $(E, d)$  un espace métrique.

Soient  $\mathbf{x} \in E$  et  $V \subset E$ . On dit que  $V$  est un **voisinage** de  $\mathbf{x}$  s'il existe  $r \in \mathbb{R}_+^*$  tel que  $\mathcal{B}(\mathbf{x}, r) \subset V$ .

On dit que  $U \subset E$  est un **ouvert** de  $E$  si elle est voisinage de tous ses points:

$$\forall \mathbf{x} \in U, \exists r \in \mathbb{R}_+^*, \mathcal{B}(\mathbf{x}, r) \subset U \quad (\text{B.5})$$

**Proposition B.1.4.** •  $E$  et  $\emptyset$  sont des ouverts,

- une réunion quelconque d'ouverts est un ouvert,
- une intersection finie d'ouverts est un ouvert.

On dit que  $U \subset E$  est un **fermé** de  $E$  si son complémentaire est un ouvert de  $E$ .

**Proposition B.1.5.** •  $E$  et  $\emptyset$  sont des fermés,

- une réunion finie de fermés est un fermé,

- une intersection quelconque de fermés est un fermé.

Soit  $A \subset E$ .

- On dit que  $\mathbf{x} \in E$  est un **point intérieur** de  $A$  si

$$\exists r \in \mathbb{R}_+^*, \mathcal{B}(\mathbf{x}, r) \subset A.$$

On appelle **intérieur** de  $A$  et on note  $\overset{\circ}{A}$  l'ensemble des points intérieurs de  $A$ . L'ensemble  $\overset{\circ}{A}$  est un ouvert : c'est le plus grand ouvert contenu dans  $A$ .

- On dit que  $\mathbf{x} \in E$  est un **point adhérent** à  $A$  si

$$\forall r \in \mathbb{R}_+^*, \mathcal{B}(\mathbf{x}, r) \cap A \neq \emptyset.$$

On appelle **adhérence** de  $A$  et on note  $\bar{A}$  l'ensemble des points adhérents de  $A$ . L'ensemble  $\bar{A}$  est un fermé : c'est le plus petit fermé contenant  $A$ .

**Théorème B.1.4.** Soient  $A \subset E$  et  $\mathbf{x} \in E$ .

- $\mathbf{x} \in \bar{A}$  si et seulement s'il existe une suite  $(\mathbf{u}^{[k]})$  de  $A$  qui converge vers  $\mathbf{x}$ .
- $A$  est fermé si et seulement si, pour toute suite  $(\mathbf{u}^{[k]})$  de  $A$  qui converge vers  $\alpha \in E$ , alors  $\alpha \in A$ .

**Définition B.1.6.** Une partie  $K \subset E$  est dite **compacte** si, de toute suite  $(\mathbf{u}^{[k]})$  de  $K$ , on peut extraire une sous-suite convergente vers un élément de  $K$ .

- toute réunion finie de compacts est compacte,
- toute intersection quelconque de compacts est compacte,
- toute partie compacte de  $E$  est fermée et bornée.

### Limites et continuité

Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction. Soit  $\alpha \in E$ . on dit que  $f$  **admet une limite** en  $\alpha$  si

$$\exists \beta \in F, \forall \varepsilon > 0, \exists \delta > 0, \forall \mathbf{x} \in E, \left( d(\mathbf{x}, \alpha) < \delta \Rightarrow d(f(\mathbf{x}), \beta) < \varepsilon \right) \quad (\text{B.6})$$

Cette limite, si elle existe, est nécessairement unique, égale à  $\beta$ , et on note alors

$$\lim_{\mathbf{x} \rightarrow \alpha} f(\mathbf{x}) = \beta \quad \text{ou} \quad f(\mathbf{x}) \xrightarrow{\mathbf{x} \rightarrow \alpha} \beta.$$

**Proposition B.1.6.**  $f$  admet une limite  $\beta$  en  $\alpha$  si et seulement si pour toute suite  $\mathbf{u}^{[k]}$  de  $E$  qui converge vers  $\alpha$ , alors la suite  $f(\mathbf{u}^{[k]})$  de  $F$  converge vers  $\beta$ .

**Définition B.1.7.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction.

- $f$  est **continue en**  $\alpha \in E$  si  $f$  admet une limite en  $\alpha$  (nécessairement égale à  $f(\alpha)$ ) ou encore

$$\forall \varepsilon > 0, \exists \delta > 0, \forall \mathbf{x} \in E, \left( d(\mathbf{x}, \alpha) < \delta \Rightarrow d(f(\mathbf{x}), f(\alpha)) < \varepsilon \right). \quad (\text{B.7})$$

- $f$  est **continue sur**  $A \subset E$  si elle est continue en chaque point de  $A$ .

**Théorème B.1.5.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction.  $f$  est continue en  $\alpha \in E$  si et seulement si, pour toute suite  $\mathbf{u}^{[k]}$  de  $E$  qui converge vers  $\alpha$ , alors la suite  $f(\mathbf{u}^{[k]})$  de  $F$  converge vers  $f(\alpha)$ .

**Théorème B.1.6.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction. Les assertions suivantes sont équivalentes :

- $f$  est continue sur  $E$ ,
- L'image réciproque d'un ouvert de  $F$  par  $f$  est un ouvert de  $E$ ,
- L'image réciproque d'un fermé de  $F$  par  $f$  est un fermé de  $E$ .

**Définition B.1.8.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction. On dit que  $f$  est **Uniformément continue** sur  $A \subset E$  si

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ tel que } \forall (\mathbf{x}, \mathbf{y}) \in A^2, \left( d(\mathbf{x}, \mathbf{y}) < \delta \Rightarrow d(f(\mathbf{x}), f(\mathbf{y})) < \varepsilon \right).$$

**Définition B.1.9.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques, et  $f : E \rightarrow F$  une fonction.

- On dit que  $f$  est **Lipschitzienne** de rapport  $K \in \mathbb{R}_+$  ou  **$K$ -lipschitzienne** sur  $A \subset E$  si

$$\forall (\mathbf{x}, \mathbf{y}) \in A^2, d(f(\mathbf{x}), f(\mathbf{y})) \leq K d(\mathbf{x}, \mathbf{y}).$$

- On dit que  $f$  est **contractante** sur  $A \subset E$  si elle est **lipschitzienne** de rapport  $K \in [0, 1[$  sur  $A \subset E$ .

- Toute application lipschitzienne est uniformément continue.
- Toute application uniformément continue est continue.

Les réciproques sont fausses.

**Théorème B.1.7.** Soient  $(E, d)$  et  $(F, d)$  deux espaces métriques,  $K$  un compact de  $E$ , et  $f : K \rightarrow F$ . Si  $f$  est continue sur  $K$  alors  $f(K)$  est un compact de  $F$ .

**Théorème B.1.8** (Théorème de Heine). Toute fonction continue sur un compact est uniformément continue.

## Suites de Cauchy

**Définition B.1.10** (Suite de Cauchy). Soit  $(E, d)$  un **espace métrique**. Une suite  $(\mathbf{x}^{[k]})_{k \in \mathbb{N}}$  d'éléments de  $E$  est dite **de Cauchy** si

$$\forall \varepsilon > 0, \exists M \in \mathbb{N}, \text{ tel que } \forall (p, q) \in \mathbb{N}^2, p, q \geq M, d(\mathbf{x}^{[p]}, \mathbf{x}^{[q]}) < \varepsilon.$$

ce qui correspond à

$$\lim_{m \rightarrow +\infty} \sup_{p, q \geq m} d(\mathbf{x}^{[p]}, \mathbf{x}^{[q]}) = 0.$$

Une autre manière de l'écrire est

$$\forall \varepsilon > 0, \exists M \in \mathbb{N}, \text{ tel que } \forall k \in \mathbb{N}, k \geq M, \forall l \in \mathbb{N}, d(\mathbf{x}^{[k+l]}, \mathbf{x}^{[k]}) < \varepsilon.$$

ce qui correspond à

$$\lim_{m \rightarrow +\infty} \sup_{k \geq m, l \geq 0} d(\mathbf{x}^{[k+l]}, \mathbf{x}^{[k]}) = 0.$$

**Définition B.1.11** (Espace métrique complet). *Un espace métrique est dit **complet** si toute suite de Cauchy converge.*

**Proposition B.1.7.** *Si  $E$  est un espace vectoriel normé de norme  $\|\cdot\|$  alors  $E$  est un espace métrique pour la distance  $d$  issue de sa norme et définie par  $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ ,  $\forall (\mathbf{x}, \mathbf{y}) \in E^2$ .*

**Définition B.1.12** (Espace de Banach). *On appelle **espace de Banach** un espace vectoriel normé complet pour la distance issue de sa norme.*

Par exemple, les espaces vectoriels normés  $(\mathbb{R}, |\cdot|)$ ,  $(\mathbb{C}, |\cdot|)$ ,  $(\mathbb{R}^n, \|\cdot\|)$ ,  $(\mathbb{C}^n, \|\cdot\|)$  sont des espaces de Banach. Plus généralement, un espace vectoriel normé de dimension finie est un espace de Banach.

### Ordre de convergence

**Définition B.1.13** (Ordre de convergence). *Soient  $(E, d)$  un **espace métrique** et  $(\mathbf{u}^{[k]})_{k \in \mathbb{N}}$  une suite d'éléments de  $E$  **convergeant vers**  $\alpha \in E$  avec,  $\forall k \in \mathbb{N}$ ,  $\mathbf{u}^{[k]} \neq \alpha$ .*

*Soit  $p \in [1, +\infty[$ . On dit que cette suite **converge vers**  $\alpha$  avec un **ordre  $p$  au moins** si*

$$\exists C > 0, \exists k_0 \in \mathbb{N}, \text{ tels que } \forall k \geq k_0, d(\mathbf{u}^{[k+1]}, \alpha) \leq C d(\mathbf{u}^{[k]}, \alpha)^p. \quad (\text{B.8})$$

*où  $C < 1$  si  $p = 1$ .*

*On dit que cette suite **converge vers**  $\alpha$  avec un **ordre  $p$  (exactement)** si elle converge à l'ordre  $p$  au moins et si*

$$\forall \varepsilon > 0, \lim_{k \rightarrow +\infty} \frac{d(\mathbf{u}^{[k+1]}, \alpha)}{d(\mathbf{u}^{[k]}, \alpha)^{p+\varepsilon}} = +\infty. \quad (\text{B.9})$$

Soient  $(E, d)$  un **espace métrique** et  $(\mathbf{u}^{[k]})_{k \in \mathbb{N}}$  une suite d'éléments de  $E$  **convergeant vers**  $\alpha \in E$  avec,  $\forall k \in \mathbb{N}$ ,  $\mathbf{u}^{[k]} \neq \alpha$ .

Soit  $p \in [1, +\infty[$ .

La suite **converge vers**  $\alpha$  à l'ordre 1 (exactement) si

$$\exists \mu \in ]0, 1[, \text{ tel que } \lim_{k \rightarrow +\infty} \frac{d(\mathbf{u}^{[k+1]}, \alpha)}{d(\mathbf{u}^{[k]}, \alpha)} = \mu. \quad (\text{B.10})$$

Dans ce cas la convergence est dite **linéaire**.

- Si (B.10) est vérifiée pour  $\mu = 0$ , alors la convergence est dite **super-linéaire**.
- Si (B.10) n'est vérifiée pour aucun  $\mu \in ]0, 1[$ , alors la convergence est dite **sous-linéaire**.

La suite **converge vers**  $\alpha$  à l'ordre  $p > 1$  (exactement) si

$$\exists \mu > 0, \text{ tel que } \lim_{k \rightarrow +\infty} \frac{d(\mathbf{u}^{[k+1]}, \alpha)}{d(\mathbf{u}^{[k]}, \alpha)^p} = \mu. \quad (\text{B.11})$$

et dans ce cas la convergence est **super-linéaire**.

La convergence d'ordre 2 (resp. 3) est dite **quadratique** (resp. **cubique**).

## B.2 Algèbre linéaire

### NOTE

Toute cette partie peut être joyeusement omise par tout Homo sapiens *algebra linearis* compatible. Toutefois une lecture rapide permet de se rafraîchir la mémoire.

Soit  $V$  un **espace vectoriel** de dimension finie  $n$ , sur le corps  $\mathbb{R}$  des nombres réels, ou sur le corps  $\mathbb{C}$  des nombres complexes. Notons plus généralement  $\mathbb{K}$  le corps  $\mathbb{R}$  ou  $\mathbb{C}$ .

### B.2.1 Vecteurs

Une **base** de  $V$  est un ensemble  $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$  de  $n$  **vecteurs linéairement indépendants**. Le vecteur  $\mathbf{v} = \sum_{i=1}^n v_i \mathbf{e}_i$  sera représenté par le **vecteur colonne**

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

et on désignera par  $\mathbf{v}^t$  et  $\mathbf{v}^*$  les **vecteurs lignes** suivants

$$\mathbf{v}^t = (v_1 \quad v_2 \quad \dots \quad v_n), \quad \mathbf{v}^* = (\bar{v}_1 \quad \bar{v}_2 \quad \dots \quad \bar{v}_n)$$

où  $\bar{\alpha}$  est le nombre **complexe conjugué** du nombre  $\alpha$ .

- Définition B.2.1.**
- Le vecteur ligne  $\mathbf{v}^t$  est le **vecteur transposé** du vecteur colonne  $\mathbf{v}$ .
  - Le vecteur ligne  $\mathbf{v}^*$  est le **vecteur adjoint** du vecteur colonne  $\mathbf{v}$ .

**Définition B.2.2.** L'application  $\langle \bullet, \bullet \rangle : \mathbb{K}^n \times \mathbb{K}^n \rightarrow \mathbb{K}$  définie pour tout  $(\mathbf{u}, \mathbf{v}) \in \mathbb{K}^n \times \mathbb{K}^n$  par

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^t \cdot \mathbf{v} = \mathbf{v}^t \cdot \mathbf{u} = \sum_{i=1}^n u_i v_i, \quad \text{si } \mathbb{K} = \mathbb{R} \tag{B.12}$$

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^* \cdot \mathbf{v} = \overline{\mathbf{v}^* \cdot \mathbf{u}} = \overline{\langle \mathbf{v}, \mathbf{u} \rangle} = \sum_{i=1}^n \bar{u}_i v_i, \quad \text{si } \mathbb{K} = \mathbb{C} \tag{B.13}$$

est appelée **produit scalaire euclidien** si  $\mathbb{K} = \mathbb{R}$ , **hermitien**<sup>a</sup> si  $\mathbb{K} = \mathbb{C}$ . Pour rappeler la dimension de l'espace, on écrit

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle_n.$$

<sup>a</sup>La convention choisie pour le produit scalaire hermitien étant ici : linéarité à droite et semi-linéarité à gauche. Il est aussi possible de définir le produit scalaire hermitien par le complexe conjugué de (B.13) :

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^* \cdot \mathbf{u} = \sum_{i=1}^n u_i \bar{v}_i.$$

Dans ce cas le produit scalaire est une forme sesquilinéaire à droite.

**Définition B.2.3.** Soit  $V$  est un espace vectoriel muni d'un produit scalaire.

- ◇ Deux vecteurs  $\mathbf{u}$  et  $\mathbf{v}$  sont **Orthogonaux** si  $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ .
- ◇ Un vecteur  $\mathbf{v}$  est **orthogonal à une partie**  $U$  de  $V$  si

$$\forall \mathbf{u} \in U, \langle \mathbf{u}, \mathbf{v} \rangle = 0.$$

On note  $\mathbf{v} \perp U$ .

◇ Un ensemble de vecteurs  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$  de l'espace  $V$  est dit **orthonormal** si

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{i,j}, \quad \forall (i, j) \in \llbracket 1, k \rrbracket^2$$

où  $\delta_{ij}$  est le **symbole de Kronecker** :  $\delta_{i,j} = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{si } i \neq j. \end{cases}$

**Définition B.2.4.** Le vecteur nul de  $\mathbb{K}^n$  est représenté par  $\mathbf{0}_n$  ou  $\mathbf{0}$  lorsqu'il n'y a pas d'ambiguïté.

**Définition B.2.5.** Soit  $\mathbf{u} \in \mathbb{K}^n$  non nul. On définit l'**opérateur de projection sur  $\mathbf{u}$**  par

$$\text{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u} = \frac{1}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u} \mathbf{u}^* \mathbf{v}, \quad \forall \mathbf{v} \in \mathbb{K}^n. \quad (\text{B.14})$$

La matrice  $\mathbb{P}_{\mathbf{u}} = \mathbf{u} \mathbf{u}^*$  s'appelle la matrice de la projection orthogonale suivant le vecteur  $\mathbf{u}$ .

**Proposition B.2.1** (Procédé de Gram-Schmidt). Soit  $\{\mathbf{v}_i\}_{i \in \llbracket 1, n \rrbracket}$  une base de  $\mathbb{K}^n$ . On construit successivement les vecteurs  $\mathbf{u}_i$

$$\mathbf{u}_i = \mathbf{v}_i - \sum_{k=1}^{i-1} \text{proj}_{\mathbf{u}_k}(\mathbf{v}_i) = \mathbf{v}_i - \sum_{k=1}^{i-1} \frac{\langle \mathbf{u}_k, \mathbf{v}_i \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \mathbf{u}_k, \quad \forall i \in \llbracket 1, n \rrbracket.$$

Ils forment une **base orthogonale** de  $\mathbb{K}^n$  et  $\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) = \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_i)$ ,  $\forall i \in \llbracket 1, n \rrbracket$  (voir Exercice 2.3.13, page 257).

Pour construire une **base orthonormale**  $\{\mathbf{z}_i\}_{i \in \llbracket 1, n \rrbracket}$ , il suffit de normaliser les vecteurs de la base orthogonale:

$$\mathbf{z}_i = \frac{\mathbf{u}_i}{\langle \mathbf{u}_i, \mathbf{u}_i \rangle^{1/2}}, \quad \forall i \in \llbracket 1, n \rrbracket.$$

## B.2.2 Matrices

### Généralités

Une matrice à  $m$  lignes et  $n$  colonnes est appelée *matrice de type  $(m, n)$* , et on note  $\mathcal{M}_{m,n}(\mathbb{K})$ , ou simplement  $\mathcal{M}_{m,n}$ , l'espace vectoriel sur le corps  $\mathbb{K}$  formé par les matrices de type  $(m, n)$  à éléments dans  $\mathbb{K}$ .

Une matrice  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{K})$  d'éléments  $A_{ij} \in \mathbb{K}$  est notée

$$\mathbb{A} = (A_{ij})_{1 \leq i \leq m, 1 \leq j \leq n},$$

le premier indice  $i$  correspond aux lignes et le second  $j$  aux colonnes. On désigne par  $(\mathbb{A})_{ij}$  l'élément de la  $i^{\text{ème}}$  ligne et de la  $j^{\text{ème}}$  colonne. On peut aussi le noter  $A_{i,j}$ .

**Définition B.2.6.** La matrice nulle de  $\mathcal{M}_{m,n}(\mathbb{K})$  est représentée par  $\mathbb{O}_{m,n}$  ou  $\mathbb{O}$  lorsqu'il n'y a pas d'ambiguïté. Si  $m = n$  on peut aussi noter  $\mathbb{O}_n$  cette matrice.

**Définition B.2.7.** ◇ Soit une matrice  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{C})$ , on note  $\mathbb{A}^* \in \mathcal{M}_{n,m}(\mathbb{C})$  la **matrice adjointe** de la matrice  $\mathbb{A}$ , définie de façon unique par

$$\langle \mathbb{A} \mathbf{u}, \mathbf{v} \rangle_m = \langle \mathbf{u}, \mathbb{A}^* \mathbf{v} \rangle_n, \quad \forall \mathbf{u} \in \mathbb{C}^m, \quad \forall \mathbf{v} \in \mathbb{C}^n$$

qui entraîne  $(\mathbb{A}^*)_{ij} = \overline{A_{ji}}$ .

◇ Soit une matrice  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ , on note  $\mathbb{A}^t \in \mathcal{M}_{n,m}(\mathbb{R})$  la **matrice transposée** de la

matrice  $\mathbb{A}$ , définie de façon unique par

$$\langle \mathbb{A}\mathbf{u}, \mathbf{v} \rangle_m = \langle \mathbf{u}, \mathbb{A}^t \mathbf{v} \rangle_n, \quad \forall \mathbf{u} \in \mathbb{R}^n, \quad \forall \mathbf{v} \in \mathbb{R}^m$$

qui entraîne  $(\mathbb{A}^t)_{ij} = A_{ji}$ .

**Définition B.2.8.** Si  $\mathbb{A} \in \mathcal{M}_{m,p}(\mathbb{K})$  et  $\mathbb{B} \in \mathcal{M}_{p,n}(\mathbb{K})$ , le **produit**  $\mathbb{A}\mathbb{B} \in \mathcal{M}_{m,n}(\mathbb{K})$  est défini par

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket, \quad (\mathbb{A}\mathbb{B})_{ij} = \sum_{k=1}^p A_{ik} B_{kj}. \quad (\text{B.15})$$

**EXERCICE 2.2.1 résultats à savoir**

Soient  $\mathbb{A} \in \mathcal{M}_{m,p}(\mathbb{K})$  et  $\mathbb{B} \in \mathcal{M}_{p,n}(\mathbb{K})$ , montrer que

$$(\mathbb{A}\mathbb{B})^t = \mathbb{B}^t \mathbb{A}^t, \quad \text{si } \mathbb{K} = \mathbb{R}, \quad (\text{B.1})$$

$$(\mathbb{A}\mathbb{B})^* = \mathbb{B}^* \mathbb{A}^*, \quad \text{si } \mathbb{K} = \mathbb{C} \quad (\text{B.2})$$

Les matrices considérées jusqu'à la fin de ce paragraphe sont carrées.

**Définition B.2.9.** Si  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  alors les éléments  $A_{ii} = (\mathbb{A})_{ii}$  sont appelés **éléments diagonaux** et les éléments  $A_{ij} = (\mathbb{A})_{ij}, i \neq j$  sont appelés **éléments hors-diagonaux**.

**Définition B.2.10.** On appelle **matrice identité** de  $\mathcal{M}_n(\mathbb{K})$  la matrice dont les éléments diagonaux sont tous égaux à 1 et les éléments hors-diagonaux nuls. On la note  $\mathbb{I}$  ou encore  $\mathbb{I}_n$  et on a

$$(\mathbb{I})_{i,j} = \delta_{ij}, \quad \forall (i, j) \in \llbracket 1, n \rrbracket^2.$$

**Définition B.2.11.** Une matrice  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  est **inversible** ou **régulière** s'il existe une matrice  $\mathbb{B} \in \mathcal{M}_n(\mathbb{K})$  vérifiant

$$\mathbb{A}\mathbb{B} = \mathbb{B}\mathbb{A} = \mathbb{I} \quad (\text{B.16})$$

Dans le cas contraire, on dit que la matrice  $\mathbb{A}$  est **singulière** ou **non inversible**.

On peut noter que la matrice  $\mathbb{B}$  est unique. En effet, soient  $\mathbb{B}_1$  et  $\mathbb{B}_2$  vérifiant (B.16). On a alors  $\mathbb{A}\mathbb{B}_2 = \mathbb{I}$  et donc  $\mathbb{B}_1(\mathbb{A}\mathbb{B}_2) = \mathbb{B}_1$ . On a aussi  $\mathbb{B}_1\mathbb{A} = \mathbb{I}$  et donc  $(\mathbb{B}_1\mathbb{A})\mathbb{B}_2 = \mathbb{B}_2$ . Le produit des matrices étant associatif on a  $\mathbb{B}_1(\mathbb{A}\mathbb{B}_2) = (\mathbb{B}_1\mathbb{A})\mathbb{B}_2$  et donc  $\mathbb{B}_1 = \mathbb{B}_2$ .

**Définition B.2.12.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  une matrice inversible. On note  $\mathbb{A}^{-1} \in \mathcal{M}_n(\mathbb{K})$  l'unique matrice vérifiant

$$\mathbb{A}\mathbb{A}^{-1} = \mathbb{A}^{-1}\mathbb{A} = \mathbb{I}. \quad (\text{B.17})$$

Cette matrice est appelée **matrice inverse** de  $\mathbb{A}$ .

**Définition B.2.13.** Soit  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{K})$

- ◊ On note  $\ker(\mathbb{A}) = \{\mathbf{v} \in \mathbb{K}^n ; \mathbb{A}\mathbf{v} = 0\}$  le **noyau** de  $\mathbb{A}$ .
- ◊ On note  $\text{im}(\mathbb{A}) = \{\mathbb{A}\mathbf{v} \in \mathbb{K}^m ; \mathbf{v} \in \mathbb{K}^n\}$  l'**image** de  $\mathbb{A}$ .
- ◊ On note  $\text{rank}(\mathbb{A}) \stackrel{\text{def}}{=} \dim(\text{im}(\mathbb{A}))$  le **rang** de  $\mathbb{A}$ .

**Théorème B.2.1** ((théorème du rang)). Soit  $A \in \mathcal{M}_{m,n}(\mathbb{K})$ . On a

$$\text{rank}(A) + \dim(\ker(A)) = n$$

**Proposition B.2.2.** Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . Les propriétés suivantes sont équivalentes

- $A$  est inversible,
- $\text{rank}(A) = n$ ,
- $x \in \mathbb{K}^n$ ,  $Ax = 0 \Rightarrow x = 0$ , (i.e.  $\ker A = \{0\}$ )
- $\det(A) \neq 0$ ,
- toutes les valeurs propres de  $A$  sont non nulles,
- il existe  $B \in \mathcal{M}_n(\mathbb{K})$  tel que  $AB = I$ ,
- il existe  $B \in \mathcal{M}_n(\mathbb{K})$  tel que  $BA = I$ .

**EXERCICE 2.2.2 résultats à savoir**

Soient  $A \in \mathcal{M}_n(\mathbb{K})$  et  $B \in \mathcal{M}_n(\mathbb{K})$  inversibles. Montrer que  $AB$  inversible et

$$(A^t)^{-1} = (A^{-1})^t, \text{ si } \mathbb{K} = \mathbb{R}, \quad (\text{B.1})$$

$$(A^*)^{-1} = (A^{-1})^*, \text{ si } \mathbb{K} = \mathbb{C}. \quad (\text{B.2})$$

$$(AB)^{-1} = B^{-1}A^{-1} \quad (\text{B.3})$$

$$(A^{-1})^{-1} = A \quad (\text{B.4})$$

**Définition B.2.14.** Une matrice *carrée*  $A$  est :

- ◇ *symétrique* si  $A$  est réelle et  $A = A^t$ ,
- ◇ *hermitienne* si  $A = A^*$ ,
- ◇ *normale* si  $AA^* = A^*A$ ,
- ◇ *orthogonale* si  $A$  est réelle et  $AA^t = A^tA = I$ ,
- ◇ *unitaire* si  $AA^* = A^*A = I$ ,

**Proposition B.2.3.** • une matrice *symétrique* ou *hermitienne* est nécessairement *normale*.

- une matrice *orthogonale* (resp. *unitaire*) est nécessairement *normale* et *inversible* d'inverse  $A^t$  (resp.  $A^*$ ).

**Définition B.2.15.** Soit  $A \in \mathcal{M}_n(\mathbb{C})$  une matrice *hermitienne*.

- ◇ Elle est *définie positive* si 
$$\langle Au, u \rangle > 0, \forall u \in \mathbb{C}^n \setminus \{0\} \quad (\text{B.18})$$

- ◇ Elle est *semi définie positive* si 
$$\langle Au, u \rangle \geq 0, \forall u \in \mathbb{C}^n \setminus \{0\} \quad (\text{B.19})$$

**EXERCICE 2.2.3**

Soit  $A \in \mathcal{M}_n(\mathbb{C})$ .

Q. 1

Que peut-on dire de la matrice  $AA^*$ ? Et si la matrice  $A$  est inversible?

Q. 2

Proposer une technique permettant de générer une matrice hermitienne semi-définie positive à partir d'une matrice aléatoire quelconque.

Q. 3

Proposer une technique permettant de générer une matrice hermitienne définie positive à partir d'une matrice triangulaire inférieure inversible aléatoire.

**Définition B.2.16.** Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . La trace d'une matrice carrée  $A = (a_{ij})$  est définie par

$$\operatorname{tr}(A) = \sum_{i=1}^n a_{ii}.$$

**Définition B.2.17.** Soit  $\mathcal{T}_n$  le **groupe des permutations** de l'ensemble  $\{1, 2, \dots, n\}$ . A tout élément  $\sigma \in \mathcal{T}_n$ , on associe la **matrice de permutation** de  $\mathbb{P}_\sigma \in \mathcal{M}_n(\mathbb{K})$  est définie par

$$(\mathbb{P}_\sigma)_{i,j} = \delta_{i,\sigma(j)}.$$

**EXERCICE 2.2.4 résultats à savoir**

Montrer qu'une matrice de permutation est orthogonale.

**Définition B.2.18.** Soient  $A = (A_{i,j})_{i,j=1}^n \in \mathcal{M}_n(\mathbb{K})$  et  $\mathcal{T}_n$  le **groupe des permutations** de l'ensemble  $\{1, 2, \dots, n\}$ . Le **déterminant** d'une matrice  $A$  est défini par

$$\det(A) = \sum_{\sigma \in \mathcal{T}_n} \varepsilon_\sigma \prod_{j=1}^n A_{\sigma(j),j}$$

où  $\varepsilon_\sigma$  désigne la signature de la permutation  $\sigma$ .

**Proposition B.2.4** (Méthode de Laplace ou des cofacteurs). Soit  $A = (A_{i,j})_{i,j=1}^n \in \mathcal{M}_n(\mathbb{K})$ . On note  $A^{[i,j]} \in \mathcal{M}_{n-1}(\mathbb{K})$  la matrice obtenue en supprimant la ligne  $i$  et la colonne  $j$  de  $A$ . On a alors le **développement par rapport à la ligne**  $i \in \llbracket 1, n \rrbracket$

$$\det(A) = \sum_{j=1}^n (-1)^{i+j} A_{i,j} \det(A^{[i,j]}), \quad (\text{B.20})$$

et le **développement par rapport à la colonne**  $j \in \llbracket 1, n \rrbracket$

$$\det(A) = \sum_{i=1}^n (-1)^{i+j} A_{i,j} \det(A^{[i,j]}). \quad (\text{B.21})$$

Le terme  $(-1)^{i+j} \det(A^{[i,j]})$  est appelé le **cofacteur** du terme  $A_{i,j}$ .

**Définition B.2.19.** Soit  $A \in \mathcal{M}_n(\mathbb{K})$ . On dit que  $\lambda \in \mathbb{C}$  est **valeur propre** de  $A$  s'il existe  $\mathbf{u} \in \mathbb{C}^n$  non nul tel que

$$A\mathbf{u} = \lambda\mathbf{u}. \quad (\text{B.22})$$

Le vecteur  $\mathbf{u}$  est appelé **vecteur propre** associé à la valeur propre  $\lambda$ .  
Le couple  $(\lambda, \mathbf{u})$  est appelé **élément propre** de  $\mathbb{A}$ .

**Définition B.2.20.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ . Soit  $\lambda \in \mathbb{C}$  une valeur propre de  $\mathbb{A}$ . Le sous-espace

$$E_\lambda = \{\mathbf{u} \in \mathbb{C}^n : \mathbb{A}\mathbf{u} = \lambda\mathbf{u}\} = \ker(\mathbb{A} - \lambda\mathbb{I}) \quad (\text{B.23})$$

est appelé **sous-espace propre** associé à la valeur propre  $\lambda$ . La dimension de  $E_\lambda$  est appelée **multiplicité géométrique** de la valeur propre  $\lambda$ .

**Définition B.2.21.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ . Le polynôme de degré  $n$  défini par

$$\mathcal{P}_\mathbb{A}(\lambda) = \det(\mathbb{A} - \lambda\mathbb{I}) \quad (\text{B.24})$$

est appelé **polynôme caractéristique** de la matrice  $\mathbb{A}$ .

**Proposition B.2.5.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ .

- ◇ Les racines complexes du polynôme caractéristique  $\mathcal{P}_\mathbb{A}$  sont les valeurs propres de la matrice  $\mathbb{A}$ .
- ◇ Si la racine  $\lambda$  de  $\mathcal{P}_\mathbb{A}$  est de multiplicité  $k$ , on dit que la valeur propre  $\lambda$  est de **Multiplicité algébrique**  $k$ .
- ◇ La matrice  $\mathbb{A}$  possède  $n$  valeurs propres distinctes ou non.

**Définition B.2.22.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ . On note  $\lambda_i(\mathbb{A})$ ,  $i \in \llbracket 1, n \rrbracket$ , les  $n$  valeurs propres de  $\mathbb{A}$ . Le **spectre** de la matrice  $\mathbb{A}$  est le sous-ensemble

$$\text{Sp}(\mathbb{A}) = \bigcup_{i=1}^n \{\lambda_i(\mathbb{A})\} \quad (\text{B.25})$$

du plan complexe.

**Proposition B.2.6.** Soient  $\mathbb{A}$  et  $\mathbb{B}$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$ . On a les relations suivantes

$$\text{tr}(\mathbb{A}) = \sum_{i=1}^n \lambda_i(\mathbb{A}), \quad (\text{B.26})$$

$$\det(\mathbb{A}) = \prod_{i=1}^n \lambda_i(\mathbb{A}), \quad (\text{B.27})$$

$$\text{tr}(\mathbb{A}\mathbb{B}) = \text{tr}(\mathbb{B}\mathbb{A}), \quad (\text{B.28})$$

$$\text{tr}(\mathbb{A} + \mathbb{B}) = \text{tr} \mathbb{A} + \text{tr} \mathbb{B}, \quad (\text{B.29})$$

$$\det(\mathbb{A}\mathbb{B}) = \det(\mathbb{A}) \det(\mathbb{B}) = \det(\mathbb{B}\mathbb{A}), \quad (\text{B.30})$$

$$\det(\mathbb{A}^*) = \overline{\det(\mathbb{A})}. \quad (\text{B.31})$$

**Définition B.2.23.** Le **rayon spectral** d'une matrice  $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$  est le nombre  $\geq 0$  défini par

$$\rho(\mathbb{A}) = \max \{|\lambda_i(\mathbb{A})|; i \in \llbracket 1, n \rrbracket\}$$

## Matrices particulières

**Définition B.2.24.** Une matrice carrée  $A \in \mathcal{M}_n$  est :

- ◇ **diagonale** si  $a_{ij} = 0$  pour  $i \neq j$ ,
- ◇ **triangulaire supérieure** si  $a_{ij} = 0$  pour  $i > j$ ,
- ◇ **triangulaire inférieure** si  $a_{ij} = 0$  pour  $i < j$ ,
- ◇ **triangulaire** si elle est triangulaire supérieure ou triangulaire inférieure
- ◇ **à diagonale dominante** si

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad \forall i \in \llbracket 1, n \rrbracket, \quad (\text{B.32})$$

- ◇ **à diagonale strictement dominante** si

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i \in \llbracket 1, n \rrbracket. \quad (\text{B.33})$$

**Proposition B.2.7.** Soient  $A$  et  $B$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$  triangulaires inférieures (resp. triangulaires supérieures). Alors la matrice  $AB$  est aussi triangulaire inférieure (resp. triangulaire supérieure).

De plus on a

$$(AB)_{i,i} = A_{i,i}B_{i,i}, \quad \forall i \in \llbracket 1, n \rrbracket.$$

*Preuve.* (voir Exercice 2.3.8, page 248) □

**Proposition B.2.8.** Soit  $A \in \mathcal{M}_n(\mathbb{K})$  une matrice triangulaire inférieure (resp. triangulaire supérieure).

- a.  $A$  est inversible si et seulement si ses éléments diagonaux sont tous non nuls (i.e.  $A_{i,i} \neq 0$ ,  $\forall i \in \llbracket 1, n \rrbracket$ ).
- b. Si  $A$  est inversible alors son inverse est triangulaire inférieure (resp. triangulaire supérieure) et

$$(A^{-1})_{i,i} = \frac{1}{(A)_{i,i}}$$

*Preuve.* (voir Exercice 2.3.18, page 260) □

**Définition B.2.25.** On appelle **matrice bande** une matrice  $A$  telle que  $a_{ij} \neq 0$  pour  $|j - i| \leq c$ .  $c$  est la **demi largeur de bande**.

Lorsque  $c = 1$ , la matrice est dite **tridiagonale**. Lorsque  $c = 2$ , la matrice est dite **pentadiagonale**.

**Définition B.2.26.** On appelle **sous-matrice** d'une matrice donnée, la matrice obtenue en supprimant certaines lignes et certaines colonnes. En particulier, si on supprime les  $(n - k)$  dernières lignes et colonnes d'une matrice carrée  $A$  d'ordre  $n$ , on obtient la **sous matrice principale** d'ordre  $k$ .

**Définition B.2.27.** On appelle **matrice bloc** une matrice  $\mathbb{A} \in \mathcal{M}_{N,M}$  écrite sous la forme

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & \cdots & \mathbb{A}_{1,q} \\ \vdots & & \vdots \\ \mathbb{A}_{p,1} & \cdots & \mathbb{A}_{p,q} \end{pmatrix}$$

où  $\forall i \in \llbracket 1, p \rrbracket, \forall j \in \llbracket 1, q \rrbracket, \mathbb{A}_{i,j}$  est une matrice de  $\mathcal{M}_{n_i, m_j}$ . On a  $N = \sum_{i=1}^p n_i$  et  $M = \sum_{j=1}^q m_j$ .

On dit que  $\mathbb{A}$  est une matrice **bloc-carrée** si  $p = q$  et si tous les blocs diagonaux sont des matrices carrées.

**Propriété B.2.1** (Multiplication de matrices blocs). Soient  $\mathbb{A} \in \mathcal{M}_{N,M}$  et  $\mathbb{B} \in \mathcal{M}_{M,S}$ . Le produit  $\mathbb{P} = \mathbb{A}\mathbb{B} \in \mathcal{M}_{N,S}$  peut s'écrire sous forme bloc si les matrices  $\mathbb{A}$  et  $\mathbb{B}$  sont compatibles par blocs : il faut que le nombre de blocs colonne de  $\mathbb{A}$  soit égale au nombre de blocs ligne de  $\mathbb{B}$  avec correspondance des dimensions.

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & \cdots & \mathbb{A}_{1,q} \\ \vdots & & \vdots \\ \mathbb{A}_{p,1} & \cdots & \mathbb{A}_{p,q} \end{pmatrix} \text{ et } \mathbb{B} = \begin{pmatrix} \mathbb{B}_{1,1} & \cdots & \mathbb{B}_{1,r} \\ \vdots & & \vdots \\ \mathbb{B}_{q,1} & \cdots & \mathbb{B}_{q,r} \end{pmatrix}$$

avec  $\mathbb{A}_{i,k} \in \mathcal{M}_{n_i, m_k}$  et  $\mathbb{B}_{k,j} \in \mathcal{M}_{m_k, s_j}$  pour tout  $i \in \llbracket 1, p \rrbracket, k \in \llbracket 1, q \rrbracket$  et  $j \in \llbracket 1, r \rrbracket$ . La matrice produit  $\mathbb{P}$  s'écrit alors sous la forme bloc

$$\mathbb{P} = \begin{pmatrix} \mathbb{P}_{1,1} & \cdots & \mathbb{P}_{1,r} \\ \vdots & & \vdots \\ \mathbb{P}_{p,1} & \cdots & \mathbb{P}_{p,r} \end{pmatrix}$$

avec  $\forall i \in \llbracket 1, p \rrbracket, \forall j \in \llbracket 1, r \rrbracket \mathbb{P}_{i,j} \in \mathcal{M}_{n_i, s_j}$  et

$$\mathbb{P}_{i,j} = \sum_{k=1}^q \mathbb{A}_{i,k} \mathbb{B}_{k,j}.$$

**Définition B.2.28.** On dit qu'une matrice bloc-carrée  $\mathbb{A}$  est **triangulaire inférieure** (resp. **supérieure**) **par blocs** si elle peut s'écrire sous la forme d'une matrice bloc avec les sous matrices  $\mathbb{A}_{i,j} = 0$  pour  $i < j$  (resp.  $i > j$ ). Elle s'écrit donc sous la forme

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathbb{A}_{n,1} & \cdots & \cdots & \mathbb{A}_{n,n} \end{pmatrix} \text{ (resp. } \mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & \cdots & \cdots & \mathbb{A}_{n,1} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbb{A}_{n,n} \end{pmatrix} \text{)}.$$

**Définition B.2.29.** On dit qu'une matrice bloc-carrée  $\mathbb{A}$  est **diagonale par blocs** ou **bloc-diagonale** si elle peut s'écrire sous la forme d'une matrice bloc avec les sous matrices  $\mathbb{A}_{i,j} = 0$  pour  $i \neq j$ . Elle s'écrit donc sous la forme

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{A}_{n,n} \end{pmatrix}$$

**Proposition B.2.9.** Soit  $A$  une matrice bloc-carré décomposée en  $n \times n$  blocs. Si  $A$  est **bloc-diagonale** ou **triangulaire par blocs** alors son déterminant est le produit des déterminant des blocs diagonaux :

$$\det A = \prod_{i=1}^n \det A_{i,i} \tag{B.34}$$

**Proposition B.2.10.** Soit  $A$  une matrice bloc-carré **inversible** décomposée en  $n \times n$  blocs.

- Si  $A$  est **bloc-diagonale** alors son inverse (décomposée en  $n \times n$  blocs) est aussi **bloc-diagonale**.
- Si  $A$  est **triangulaire inférieure par blocs** (resp. supérieure) alors son inverse (décomposée en  $n \times n$  blocs) est aussi **triangulaire inférieure par blocs** (resp. supérieure).

Dans ces deux cas les blocs diagonaux de la matrice inverse sont les inverses des blocs diagonaux de  $A$ . On a donc

$$\begin{aligned}
 A &= \begin{pmatrix} A_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{n,n} \end{pmatrix} & \text{et } A^{-1} &= \begin{pmatrix} A_{1,1}^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{n,n}^{-1} \end{pmatrix} \\
 A &= \begin{pmatrix} A_{1,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A_{n,1} & \cdots & \cdots & A_{n,n} \end{pmatrix} & \text{et } A^{-1} &= \begin{pmatrix} A_{1,1}^{-1} & 0 & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \bullet & \cdots & \bullet & A_{n,n}^{-1} \end{pmatrix} \\
 A &= \begin{pmatrix} A_{1,1} & \cdots & \cdots & A_{n,1} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{n,n} \end{pmatrix} & \text{et } A^{-1} &= \begin{pmatrix} A_{1,1}^{-1} & \bullet & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{n,n}^{-1} \end{pmatrix}
 \end{aligned}$$

### B.2.3 Normes vectorielles et normes matricielles

**Définition B.2.30.** Une **norme** sur un espace vectoriel  $V$  est une application  $\|\bullet\| : V \rightarrow \mathbb{R}^+$  qui vérifie les propriétés suivantes

- ◊  $\|\mathbf{v}\| = 0 \iff \mathbf{v} = 0$ ,
- ◊  $\|\alpha\mathbf{v}\| = |\alpha| \|\mathbf{v}\|, \forall \alpha \in \mathbb{K}, \forall \mathbf{v} \in V$ ,
- ◊  $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|, \forall (\mathbf{u}, \mathbf{v}) \in V^2$  (inégalité triangulaire).

Une norme sur  $V$  est également appelée **norme vectorielle**. On appelle **espace vectoriel normé** un espace vectoriel muni d'une norme.

Les trois normes suivantes sont les plus couramment utilisées :

$$\begin{aligned}\|\mathbf{v}\|_1 &= \sum_{i=1}^n |v_i| \\ \|\mathbf{v}\|_2 &= \left( \sum_{i=1}^n |v_i|^2 \right)^{1/2} \\ \|\mathbf{v}\|_\infty &= \max_i |v_i|.\end{aligned}$$

**Théorème B.2.2.** Soit  $V$  un espace de dimension finie. Pour tout nombre réel  $p \geq 1$ , l'application  $\|\bullet\|_p$  définie par

$$\|\mathbf{v}\|_p = \left( \sum_{i=1}^n |v_i|^p \right)^{1/p}$$

est une norme.

**Proposition B.2.11.** Pour  $p > 1$  et  $\frac{1}{p} + \frac{1}{q} = 1$ , on a  $\forall \mathbf{u}, \mathbf{v} \in \mathbb{K}^n$

$$\sum_{i=1}^n |u_i v_i| \leq \left( \sum_{i=1}^n |u_i|^p \right)^{1/p} \left( \sum_{i=1}^n |v_i|^q \right)^{1/q} = \|\mathbf{u}\|_p \|\mathbf{v}\|_q. \quad (\text{B.35})$$

Cette inégalité s'appelle l'**inégalité de Hölder**.

**Définition B.2.31.** Deux **normes**  $\|\bullet\|$  et  $\|\bullet\|'$ , définies sur un même espace vectoriel  $V$ , sont **équivalentes** s'il existe deux constantes  $C$  et  $C'$  telles que

$$\|\mathbf{v}'\| \leq C \|\mathbf{v}\| \quad \text{et} \quad \|\mathbf{v}\| \leq C' \|\mathbf{v}'\| \quad \text{pour tout } \mathbf{v} \in V. \quad (\text{B.36})$$

**Proposition B.2.12.** Sur un espace vectoriel de dimension finie toutes les normes sont équivalentes.

**Définition B.2.32.** Une **norme matricielle** sur  $\mathcal{M}_n(\mathbb{K})$  est une application  $\|\bullet\| : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  vérifiant

- $\|\mathbb{A}\| = 0 \iff \mathbb{A} = 0$ ,
- $\|\alpha \mathbb{A}\| = |\alpha| \|\mathbb{A}\|, \forall \alpha \in \mathbb{K}, \forall \mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ ,
- $\|\mathbb{A} + \mathbb{B}\| \leq \|\mathbb{A}\| + \|\mathbb{B}\|, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$  (inégalité triangulaire)
- $\|\mathbb{A}\mathbb{B}\| \leq \|\mathbb{A}\| \|\mathbb{B}\|, \forall (\mathbb{A}, \mathbb{B}) \in \mathcal{M}_n(\mathbb{K})^2$

**Proposition B.2.13.** Etant donné une norme vectorielle  $\|\bullet\|$  sur  $\mathbb{K}^n$ , l'application  $\|\bullet\|_s : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  définie par

$$\|\mathbb{A}\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \|\mathbf{v}\| \leq 1}} \|\mathbb{A}\mathbf{v}\| = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \|\mathbf{v}\| = 1}} \|\mathbb{A}\mathbf{v}\|, \quad (\text{B.37})$$

est une norme matricielle, appelée **norme matricielle subordonnée** (à la norme vectorielle donnée).

De plus

$$\|\mathbb{A}\mathbf{v}\| \leq \|\mathbb{A}\|_s \|\mathbf{v}\| \quad \forall \mathbf{v} \in \mathbb{K}^n \quad (\text{B.38})$$

et la norme  $\|\mathbb{A}\|$  peut se définir aussi par

$$\|\mathbb{A}\|_s = \inf \{ \alpha \in \mathbb{R} : \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n \}. \quad (\text{B.39})$$

Il existe au moins un vecteur  $\mathbf{u} \in \mathbb{K}^n$  tel que

$$\mathbf{u} \neq 0 \text{ et } \|\mathbb{A}\mathbf{u}\| = \|\mathbb{A}\|_s \|\mathbf{u}\|. \quad (\text{B.40})$$

Enfin une norme subordonnée vérifie toujours

$$\|\mathbb{I}\|_s = 1 \quad (\text{B.41})$$

**Théorème B.2.3.** Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ . On a

$$\|\mathbb{A}\|_1 \stackrel{\text{déf.}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|_1}{\|\mathbf{v}\|_1} = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |a_{ij}| \quad (\text{B.42})$$

$$\|\mathbb{A}\|_2 \stackrel{\text{déf.}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sqrt{\rho(\mathbb{A}^* \mathbb{A})} = \sqrt{\rho(\mathbb{A} \mathbb{A}^*)} = \|\mathbb{A}^*\|_2 \quad (\text{B.43})$$

$$\|\mathbb{A}\|_\infty \stackrel{\text{déf.}}{=} \sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|_\infty}{\|\mathbf{v}\|_\infty} = \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |a_{ij}| \quad (\text{B.44})$$

La norme  $\|\bullet\|_2$  est invariante par transformation unitaire :

$$\mathbb{U}\mathbb{U}^* = \mathbb{I} \implies \|\mathbb{A}\|_2 = \|\mathbb{A}\mathbb{U}\|_2 = \|\mathbb{U}\mathbb{A}\|_2 = \|\mathbb{U}^*\mathbb{A}\mathbb{U}\|_2. \quad (\text{B.45})$$

Par ailleurs, si la matrice  $\mathbb{A}$  est normale :

$$\mathbb{A}\mathbb{A}^* = \mathbb{A}^*\mathbb{A} \implies \|\mathbb{A}\|_2 = \rho(\mathbb{A}). \quad (\text{B.46})$$

**Proposition B.2.14.** a. Si une matrice  $\mathbb{A}$  est hermitienne, ou symétrique (donc normale), on a  $\|\mathbb{A}\|_2 = \rho(\mathbb{A})$ .

b. Si une matrice  $\mathbb{A}$  est unitaire, ou orthogonale (donc normale), on a  $\|\mathbb{A}\|_2 = 1$ .

**Théorème B.2.4.** a. Soit  $\mathbb{A}$  une matrice carrée quelconque et  $\|\bullet\|$  une norme matricielle subordonnée ou non, quelconque. Alors

$$\rho(\mathbb{A}) \leq \|\mathbb{A}\|. \quad (\text{B.47})$$

b. Etant donné une matrice  $\mathbb{A}$  et un nombre  $\varepsilon > 0$ , il existe au moins une norme matricielle subordonnée telle que

$$\|\mathbb{A}\| \leq \rho(\mathbb{A}) + \varepsilon. \quad (\text{B.48})$$

**Théorème B.2.5.** L'application  $\|\bullet\|_E : \mathcal{M}_n \rightarrow \mathbb{R}^+$  définie par

$$\|\mathbb{A}\|_E = \left( \sum_{(i,j) \in \llbracket 1, n \rrbracket^2} |a_{ij}|^2 \right)^{1/2} = \sqrt{\text{tr}(\mathbb{A}^* \mathbb{A})}, \quad (\text{B.49})$$

pour toute matrice  $\mathbb{A} = (a_{ij})$  d'ordre  $n$ , est une norme matricielle non subordonnée (pour  $n \geq 2$ ), invariante par transformation unitaire et qui vérifie

$$\|\mathbb{A}\|_2 \leq \|\mathbb{A}\|_E \leq \sqrt{n} \|\mathbb{A}\|_2, \forall \mathbb{A} \in \mathcal{M}_n. \quad (\text{B.50})$$

De plus  $\|\mathbb{I}\|_E = \sqrt{n}$ .

**Théorème B.2.6.** a. Soit  $\|\bullet\|$  une norme matricielle subordonnée, et  $\mathbb{B}$  une matrice vérifiant

$$\|\mathbb{B}\| < 1.$$

Alors la matrice  $(\mathbb{I} + \mathbb{B})$  est inversible, et

$$\|(\mathbb{I} + \mathbb{B})^{-1}\| \leq \frac{1}{1 - \|\mathbb{B}\|}.$$

b. Si une matrice de la forme  $(\mathbb{I} + \mathbb{B})$  est singulière, alors nécessairement

$$\|\mathbb{B}\| \geq 1$$

pour toute norme matricielle, subordonnée ou non.

## B.2.4 Réduction des matrices

**Définition B.2.33.** Soit  $A : V \rightarrow V$  une application linéaire, représenté par une matrice carrée  $\mathbb{A} \in \mathcal{M}_n$  relativement à une base  $\{\mathbf{e}_i\}_{i \in \llbracket 1, n \rrbracket}$ . Relativement à une autre base  $\{\mathbf{f}_i\}_{i \in \llbracket 1, n \rrbracket}$ , la même application est représentée par la matrice

$$\mathbb{B} = \mathbb{P}^{-1} \mathbb{A} \mathbb{P} \quad (\text{B.51})$$

où  $\mathbb{P}$  est la matrice inversible dont le  $j$ -ème vecteur colonne est formé des composantes du vecteur  $\mathbf{f}_j$  dans la base  $\{\mathbf{e}_i\}_{i \in \llbracket 1, n \rrbracket}$  :

$$\mathbb{P} = \begin{pmatrix} \langle \mathbf{e}_1, \mathbf{f}_1 \rangle & \langle \mathbf{e}_1, \mathbf{f}_2 \rangle & \cdots & \langle \mathbf{e}_1, \mathbf{f}_n \rangle \\ \langle \mathbf{e}_2, \mathbf{f}_1 \rangle & \langle \mathbf{e}_2, \mathbf{f}_2 \rangle & \ddots & \vdots \\ \vdots & \ddots & \ddots & \langle \mathbf{e}_{n-1}, \mathbf{f}_n \rangle \\ \langle \mathbf{e}_n, \mathbf{f}_1 \rangle & \cdots & \langle \mathbf{e}_n, \mathbf{f}_{n-1} \rangle & \langle \mathbf{e}_n, \mathbf{f}_n \rangle \end{pmatrix} \quad (\text{B.52})$$

La matrice  $\mathbb{P}$  est appelée **matrice de passage de la base  $\{\mathbf{e}_i\}_{i \in \llbracket 1, n \rrbracket}$  dans le base  $\{\mathbf{f}_i\}_{i \in \llbracket 1, n \rrbracket}$** .

**Définition B.2.34.** On dit que la matrice carrée  $\mathbb{A}$  est diagonalisable s'il existe une matrice inversible  $\mathbb{P}$  telle que la matrice  $\mathbb{P}^{-1} \mathbb{A} \mathbb{P}$  soit diagonale.

**Remarque B.2.1.** On notera que, dans le cas où  $\mathbb{A} \in \mathcal{M}_n$  est diagonalisable, les éléments diagonaux de la matrice  $\mathbb{P}^{-1} \mathbb{A} \mathbb{P}$  sont les valeurs propres  $\lambda_1, \lambda_2, \dots, \lambda_n$  de la matrice  $\mathbb{A}$ , et que le  $j$ -ème vecteur colonne  $\mathbf{p}_j$  de la matrice  $\mathbb{P}$  est formé des composantes, dans la même base que  $\mathbb{A}$ , d'un vecteur propre associé à la valeur propre  $\lambda_j$ . On a

$$\mathbb{P}^{-1} \mathbb{A} \mathbb{P} = \text{diag}(\lambda_1, \dots, \lambda_n) \iff \mathbb{A} \mathbf{p}_j = \lambda_j \mathbf{p}_j, \quad \forall j \in \llbracket 1, n \rrbracket. \quad (\text{B.53})$$

C'est à dire qu'une matrice est diagonalisable si, et seulement si, il existe une base de vecteurs propres.

**Théorème B.2.7.** a. Etant donnée une matrice carrée  $\mathbb{A}$ , il existe une matrice unitaire  $\mathbb{U}$  telle que la matrice  $\mathbb{U}^{-1} \mathbb{A} \mathbb{U}$  soit triangulaire.

b. Etant donnée une matrice normale  $\mathbb{A}$ , il existe une matrice unitaire  $\mathbb{U}$  telle que la matrice  $\mathbb{U}^{-1} \mathbb{A} \mathbb{U}$  soit diagonale.

c. Etant donnée une matrice symétrique  $\mathbb{A}$ , il existe une matrice orthogonale  $\mathbb{O}$  telle que la matrice  $\mathbb{O}^{-1} \mathbb{A} \mathbb{O}$  soit diagonale.

## B.2.5 Suites de vecteurs et de matrices

**Définition B.2.35.** Soit  $V$  un espace vectoriel muni d'une norme  $\|\bullet\|$ , on dit qu'une suite  $(\mathbf{v}_k)$  d'éléments de  $V$  **converge vers un élément**  $\mathbf{v} \in V$ , si

$$\lim_{k \rightarrow \infty} \|\mathbf{v}_k - \mathbf{v}\| = 0$$

et on écrit

$$\mathbf{v} = \lim_{k \rightarrow \infty} \mathbf{v}_k.$$

**Théorème B.2.8.** Soit  $\mathbb{B}$  une matrice carrée. Les conditions suivantes sont équivalentes :

- $\lim_{k \rightarrow \infty} \mathbb{B}^k = 0$ ,
- $\lim_{k \rightarrow \infty} \mathbb{B}^k \mathbf{v} = 0$  pour tout vecteur  $\mathbf{v}$ ,
- $\rho(\mathbb{B}) < 1$ ,
- $\|\mathbb{B}\| < 1$  pour au moins une norme matricielle subordonnée  $\|\bullet\|$ .

**Théorème B.2.9.** Soit  $\mathbb{B}$  une matrice carrée, et  $\|\bullet\|$  une norme matricielle quelconque. Alors

$$\lim_{k \rightarrow \infty} \|\mathbb{B}^k\|^{1/k} = \rho(\mathbb{B}).$$

## B.3 Receuil d'exercices

### B.3.1 Algorithmique numérique

#### EXERCICE 2.3.1

Soient  $\mathbf{u} \in \mathbb{R}^n$  et  $\mathbf{v} \in \mathbb{R}^n$ .

Q. 1

Ecrire une fonction `dot` permettant de calculer le produit scalaire du vecteur  $\mathbf{u}$  par  $\mathbf{v}$ , noté mathématiquement par  $\langle \mathbf{u}, \mathbf{v} \rangle$ .

Q. 2

Ecrire une fonction `norm2` permettant de calculer la norme euclidienne du vecteur  $\mathbf{u}$  donnée par  $\|\mathbf{u}\|_2 \stackrel{\text{def}}{=} \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ .

Soient  $a$  et  $b$  deux réels.

Q. 3

Ecrire une fonction `aUpbV` permettant de calculer le vecteur  $\mathbf{w} \stackrel{\text{def}}{=} a\mathbf{u} + b\mathbf{v}$ .

Q. 4

Ecrire un programme algorithmique permettant de calculer  $\|\mathbf{u} - \mathbf{v}\|_2$  avec  $\mathbf{u} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$  et  $\mathbf{v} =$

$$\begin{pmatrix} 2 \\ -3 \\ 2 \end{pmatrix}$$

#### Correction

R. 1

On a

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n \mathbf{u}_i \mathbf{v}_i \in \mathbb{R}.$$

Voici une correction possible où l'on différencie la notation mathématique  $\mathbf{u}_i$ ,  $i$ -ème composante du vecteur  $\mathbf{u}$ , et la notation algorithmique  $\mathbf{U}(i)$  permettant d'accéder à l'élément  $i$  du tableau/vecteur  $\mathbf{U}$ , avec  $i \in \llbracket 1, n \rrbracket$ .

---

**Algorithme B.1** Fonction dot, produit scalaire entre deux vecteurs.

---

**Données :**  $\mathbf{U}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{U}(i) = \mathbf{u}_i$ .  
 $\mathbf{V}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{V}(i) = \mathbf{v}_i$ .

**Résultat :**  $r$  : le réel tel que  $r = \langle \mathbf{u}, \mathbf{v} \rangle$ .

---

```

1: Fonction  $r \leftarrow \text{dot}(\mathbf{U}, \mathbf{V})$ 
2:    $r \leftarrow 0$ 
3:   Pour  $i \leftarrow 1$  à  $n$  faire    ▷  $n$  est implicitement connu
4:      $r \leftarrow r + \mathbf{U}(i) * \mathbf{V}(i)$ 
5:   Fin
6: Fin

```

---

On aurait pu aussi utiliser la notation algorithmique  $\mathbf{u} \in \mathbb{R}^n$  pour décrire un vecteur/tableau de  $\mathbb{R}^n$  et on aurait alors la correspondance suivante entre notation mathématique et algorithmique:

| <i>Description</i>   | <i>Mathématiques</i> | <i>Algorithmique</i> |
|--|----------------------|----------------------|
| Composante $i \in \llbracket 1, n \rrbracket$ de $\mathbf{u} \in \mathbb{R}^n$ | $\mathbf{u}_i$       | $\mathbf{u}(i)$      |

Cependant pour bien différencier ces deux types de notations, on continue dans cet exercice à noter  $\mathbf{U}$  le vecteur algorithmique correspondant au vecteur mathématique  $\mathbf{u}$ .

R. 2

Voici une correction possible:

---

**Algorithme B.2** Fonction norm2

---

**Données :**  $\mathbf{U}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{U}(i) = \mathbf{u}_i$ .

**Résultat :**  $r$  : le réel tel que  $r = \|\mathbf{u}\|_2 \stackrel{\text{def}}{=} \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ .

---

```

1: Fonction  $r \leftarrow \text{norm2}(\mathbf{U})$ 
2:    $r \leftarrow \text{Sqrt}(\text{dot}(\mathbf{U}, \mathbf{U}))$ 
3: Fin

```

---

R. 3

On a  $\mathbf{w} \stackrel{\text{def}}{=} a\mathbf{u} + b\mathbf{v} \in \mathbb{R}^n$  et plus précisément

$$\forall i \in \llbracket 1, n \rrbracket, \mathbf{w}_i \stackrel{\text{def}}{=} a\mathbf{u}_i + b\mathbf{v}_i.$$

Voici une correction possible:

**Algorithme B.3** Fonction aUpbV

**Données :**  $\mathbf{U}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{U}(i) = \mathbf{u}_i$ .

$\mathbf{V}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{V}(i) = \mathbf{v}_i$ .

$a$  : un réel,

$b$  : un réel.

**Résultat :**  $\mathbf{W}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  
 $\forall i \in \llbracket 1, n \rrbracket, \mathbf{W}(i) = \mathbf{w}_i$ ,  
avec  $\mathbf{w}_i \stackrel{\text{def}}{=} a\mathbf{u}_i + b\mathbf{v}_i$ .

1: **Fonction**  $\mathbf{W} \leftarrow \text{aUpbV}(a, \mathbf{U}, b, \mathbf{V})$

2: **Pour**  $i \leftarrow 1$  à  $n$  **faire**

3:      $\mathbf{W}(i) \leftarrow a * \mathbf{U}(i) + b * \mathbf{V}(i)$

4: **Fin**

5: **Fin**

**R. 4**

Voici un premier programme possible:

$$1: \mathbf{U} \leftarrow \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}, \mathbf{V} \leftarrow \begin{pmatrix} 2 \\ -3 \\ 2 \end{pmatrix}$$

$$2: \mathbf{x} \leftarrow \text{norm2}(\text{aUpbV}(1, \mathbf{U}, -1, \mathbf{V}))$$

$\triangleright x$  contient le réel  $\|\mathbf{u} - \mathbf{v}\|_2$

On peut aussi écrire directement

$$1: \mathbf{x} \leftarrow \text{norm2}\left(\text{aUpbV}\left(1, \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}, -1, \begin{pmatrix} 2 \\ -3 \\ 2 \end{pmatrix}\right)\right)$$

$\triangleright x$  contient le réel  $\|\mathbf{u} - \mathbf{v}\|_2$

~~~~~ Fin correction ~~~~~

**EXERCICE 2.3.2**

**Q. 1**

Ecrire une fonction `VecZeros` retournant le vecteur nul de  $\mathbb{R}^n$ .

**Q. 2**

Ecrire une fonction `VecConst` retournant le vecteur de  $\mathbb{R}^n$  dont toutes les composantes valent  $\alpha \in \mathbb{R}$ .

**Q. 3**

Soient  $\mathbf{u} \in \mathbb{R}^n$  et  $\alpha \in \mathbb{R}$ . Ecrire une fonction `VecPlusConst` retournant le vecteur de  $\mathbf{v} \in \mathbb{R}^n$  tel que

$$\forall i \in \llbracket 1, n \rrbracket, \mathbf{v}_i = \mathbf{u}_i + \alpha.^a$$

<sup>a</sup>Cette opération n'est pas une opération algébrique dans  $\mathbb{R}^n$ , c'est à dire mathématiquement, on ne peut pas écrire  $\mathbf{u} + \alpha$ !

On suppose que l'on dispose de la fonction algorithmique `rand()` retournant un réel aléatoire suivant la loi uniforme sur  $[0, 1]$ .

**Q. 4**

Ecrire une fonction `VecRand` retournant un vecteur de  $\mathbb{R}^n$  dont toutes les composantes sont aléatoires suivant la loi uniforme sur  $[a, b]$ .

## Correction

R. 1

Voici une correction possible

---

**Algorithme B.4** Fonction `VecZeros` retournant le vecteur nul de  $\mathbb{R}^n$ .

---

**Données :**  $n$  : un entier  $> 0$ .

**Résultat :**  $z$  : le vecteur nul de  $\mathbb{R}^n$ .

---

```

1: Fonction  $z \leftarrow \text{VecZeros}(n)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $z(i) \leftarrow 0$ 
4:   Fin
5: Fin

```

---

R. 2

Voici une correction possible

---

**Algorithme B.5** Fonction `VecConst` retournant le vecteur  $v$  de  $\mathbb{R}^n$  dont toutes les composantes valent  $\alpha \in \mathbb{R}$ .

---

**Données :**  $n$  : un entier  $> 0$ ,

$\alpha$  : un réel.

**Résultat :**  $v$  : vecteur de  $\mathbb{R}^n$ .

---

```

1: Fonction  $v \leftarrow \text{VecConst}(n, \alpha)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $v(i) \leftarrow \alpha$ 
4:   Fin
5: Fin

```

---

R. 3

Voici une correction possible

---

**Algorithme B.6** Fonction `VecPlusConst` retournant le vecteur  $v$  de  $\mathbb{R}^n$  tel que  $\forall i \in \llbracket 1, n \rrbracket, v_i = u_i + \alpha$  avec

---

**Données :**  $u$  : vecteur de  $\mathbb{R}^n$ ,

$\alpha$  : un réel.

**Résultat :**  $v$  : vecteur de  $\mathbb{R}^n$ .

---

```

1: Fonction  $v \leftarrow \text{VecPlusConst}(u, \alpha)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $v(i) \leftarrow u(i) + \alpha$ 
4:   Fin
5: Fin

```

---

R. 4

On rappelle le changement de variable permettant de passer de l'intervalle  $[0, 1]$  à l'intervalle  $[a, b]$ 

$$\forall t \in [0, 1], \quad x = a + (b - a)t, \quad \text{avec } x \in [a, b].$$

Donc, si  $T$  est une variable aléatoire suivant la loi uniforme sur  $[0, 1]$ , alors  $X = a + (b - a)T$  est une variable aléatoire qui suit la loi uniforme sur  $[a, b]$ .

Voici une correction possible

**Algorithme B.7** Fonction VecRand retournant un vecteur dont toutes les composantes sont aléatoires suivant la loi uniforme sur  $[a, b]$ .

**Données :**  $n$  : un entier  $> 0$ ,  
 $a$  : un réel,  
 $b$  : un réel, tel que  $a < b$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{VecRand}(n, a, b)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $\mathbf{x}(i) \leftarrow a + (b - a) * \text{rand}()$ 
4:   Fin
5: Fin

```

Fin correction

### EXERCICE 2.3.3

**Q. 1** Ecrire une fonction `MatZeros` retournant la matrice nulle de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

**Q. 2** Ecrire une fonction `MatConst` retournant la matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$  dont toutes les composantes valent  $\alpha \in \mathbb{R}$ .

**Q. 3** Soient  $A \in \mathcal{M}_{m,n}(\mathbb{R})$  et  $\alpha \in \mathbb{R}$ . Ecrire une fonction `MatPlusConst` retournant la matrice  $B \in \mathcal{M}_{m,n}(\mathbb{R})$  telle que

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket, \quad B_{i,j} = A_{i,j} + \alpha.^a$$

<sup>a</sup>Cette opération n'est pas une opération algébrique dans  $\mathcal{M}_{m,n}(\mathbb{R})$ , c'est à dire mathématiquement, on ne peut pas écrire  $A + \alpha$ !

On suppose que l'on dispose de la fonction algorithmique `rand()` retournant un réel aléatoire suivant la loi uniforme sur  $[0, 1]$ .

**Q. 4** Ecrire une fonction `MatRand` retournant une matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$  dont toutes les composantes sont aléatoires suivant la loi uniforme sur  $[a, b]$ .

### Correction

**R. 1** Voici une correction possible

---

**Algorithme B.8** Fonction MatZeros la matrice nulle de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

**Données :**  $m$  : un entier  $> 0$ ,  
 $n$  : un entier  $> 0$ .

**Résultat :**  $\mathbb{Z}$  : la matrice nulle de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{Z} \leftarrow \text{MatZeros}(m, n)$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $n$  faire
4:        $\mathbb{Z}(i, j) \leftarrow 0$ 
5:     Fin
6:   Fin
7: Fin

```

---

**R. 2**

Voici une correction possible

---

**Algorithme B.9** Fonction MatConst retournant la matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$  dont toutes les composantes valent  $\alpha \in \mathbb{R}$ .

---

**Données :**  $m$  : un entier  $> 0$ ,  
 $n$  : un entier  $> 0$ ,  
 $\alpha$  : un réel.

**Résultat :**  $\mathbb{Z}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

```

1: Fonction  $v \leftarrow \text{MatConst}(m, n, \alpha)$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $n$  faire
4:        $\mathbb{Z}(i, j) \leftarrow \alpha$ 
5:     Fin
6:   Fin
7: Fin

```

---

**R. 3**

Voici une correction possible

---

**Algorithme B.10** Fonction MatPlusConst retournant la matrice  $\mathbb{B} \in \mathcal{M}_{m,n}(\mathbb{R})$  telle que  $\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket$ ,  $\mathbb{B}_{i,j} = \mathbb{A}_{i,j} + \alpha$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\alpha$  : un réel.

**Résultat :**  $\mathbb{B}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{B} \leftarrow \text{MatPlusConst}(\mathbb{A}, \alpha)$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $n$  faire
4:        $\mathbb{B}(i, j) \leftarrow \mathbb{A}(i, j) + \alpha$ 
5:     Fin
6:   Fin
7: Fin

```

---

**R. 4**

On rappelle le changement de variable permettant de passer de l'intervalle  $[0, 1]$  à l'intervalle  $[a, b]$

$$\forall t \in [0, 1], \quad x = a + (b - a)t, \quad \text{avec } x \in [a, b].$$

Donc, si  $T$  est une variable aléatoire suivant la loi uniforme sur  $[0, 1]$ , alors  $X = a + (b - a)T$  est une variable aléatoire qui suit la loi uniforme sur  $[a, b]$ .

Voici une correction possible

---

**Algorithme B.11** Fonction `MatRand` retournant une matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$  dont toutes les composantes sont aléatoires suivant la loi uniforme sur  $[a, b]$ .

---

**Données :**  $m$  : un entier  $> 0$ ,  
 $n$  : un entier  $> 0$ ,  
 $a$  : un réel,  
 $b$  : un réel, tel que  $a < b$ .

**Résultat :**  $\mathbb{X}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{X} \leftarrow \text{MatRand}(m, n, a, b)$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $n$  faire
4:        $\mathbb{X}(i, j) \leftarrow a + (b - a) * \text{rand}()$ 
5:     Fin
6:   Fin
7: Fin

```

---

Fin correction

#### EXERCICE 2.3.4

Soit  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ .

Q. 1

- Ecrire une fonction algorithmique `setMatCol` permettant d'initialiser le  $j$ -ème vecteur colonne de  $\mathbb{A}$  par un vecteur  $\mathbf{x} \in \mathbb{R}^m$ , avec  $j \in \llbracket 1, n \rrbracket$ .
- Ecrire une fonction algorithmique `getMatCol` retournant le  $j$ -ème vecteur colonne de  $\mathbb{A}$  avec  $j \in \llbracket 1, n \rrbracket$ .

Q. 2

- Ecrire une fonction algorithmique `setMatRow` permettant d'initialiser le  $i$ -ème vecteur ligne de  $\mathbb{A}$  par un vecteur  $\mathbf{y} \in \mathbb{R}^n$ , avec  $i \in \llbracket 1, m \rrbracket$ .
- Ecrire une fonction algorithmique `getMatRow` retournant le  $i$ -ème vecteur ligne de  $\mathbb{A}$  avec  $i \in \llbracket 1, m \rrbracket$ .

#### Correction

R. 1

Mathématiquement, le  $j$ -ème vecteur colonne de  $\mathbb{A}$  le vecteur (colonne) de  $\mathbb{R}^m$  donné par

$$\begin{pmatrix} \mathbb{A}_{1,j} \\ \vdots \\ \mathbb{A}_{m,j} \end{pmatrix}.$$

- Voici une correction possible

---

**Algorithme B.12** Fonction setMatCol

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{n,m}(\mathbb{R})$   
 $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ ,  
 $j$  : entier dans  $\llbracket 1, m \rrbracket$ .

**Résultat :**  $\mathbb{A}$  : la matrice  $\mathbb{A}$  modifiée.

```

1: Fonction  $\mathbb{A} \leftarrow \text{setMatCol}(\mathbb{A}, \mathbf{x}, j)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $\mathbb{A}(i, j) \leftarrow \mathbf{x}(i)$ 
4:   Fin
5: Fin

```

---

b. Voici une correction possible

---

**Algorithme B.13** Fonction getMatCol

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$   
 $j$  : entier dans  $\llbracket 1, n \rrbracket$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^m$ , colonne  $j$  de  $\mathbb{A}$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{getMatCol}(\mathbb{A}, j)$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $\mathbf{x}(i) \leftarrow \mathbb{A}(i, j)$ 
4:   Fin
5: Fin

```

---

**R. 2**

Le  $i$ -ème vecteur ligne de  $\mathbb{A}$  est le vecteur ligne de  $\mathbb{R}^n$  donné par

$$(\mathbb{A}_{i,1} \quad \dots \quad \mathbb{A}_{i,n}).$$

a. Voici une correction possible

---

**Algorithme B.14** Fonction setMatRow

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$   
 $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ ,  
 $i$  : entier dans  $\llbracket 1, m \rrbracket$ .

**Résultat :**  $\mathbb{A}$  : la matrice  $\mathbb{A}$  modifiée.

```

1: Fonction  $\mathbb{A} \leftarrow \text{setMatRow}(\mathbb{A}, \mathbf{x}, i)$ 
2:   Pour  $j \leftarrow 1$  à  $n$  faire
3:      $\mathbb{A}(i, j) \leftarrow \mathbf{x}(j)$ 
4:   Fin
5: Fin

```

---

b. Voici une correction possible

**Algorithme B.15** Fonction getMatRow

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$   
 $i$  : entier dans  $\llbracket 1, m \rrbracket$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{R}^n$ , ligne  $i$  de  $\mathbb{A}$ .

```

1: Fonction  $\mathbf{x} \leftarrow \text{getMatCol}(\mathbb{A}, i)$ 
2:   Pour  $j \leftarrow 1$  à  $m$  faire
3:      $\mathbf{x}(j) \leftarrow \mathbb{A}(i, j)$ 
4:   Fin
5: Fin

```

Fin correction

**EXERCICE 2.3.5****Q. 1**

Soient  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ ,  $\mathbb{B} \in \mathcal{M}_{m,n}(\mathbb{R})$ ,  $\alpha \in \mathbb{R}$  et  $\beta \in \mathbb{R}$ . Ecrire une fonction `aApbB` permettant de calculer la matrice  $\mathbb{W} \stackrel{\text{def}}{=} \alpha\mathbb{A} + \beta\mathbb{B}$ .

**Q. 2**

Soient  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  et  $\mathbf{u}$  un vecteur (colonne).

- A quelle(s) condition(s) le produit de la matrice  $\mathbb{A}$  par le vecteur  $\mathbf{u}$  existe-t-il?
- Sous les conditions précédentes, on note  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .
  - A quel espace appartient  $\mathbf{v}$ ?
  - Rappeler précisément la formule permettant de calculer les composantes de  $\mathbf{v}$ .
- Ecrire une fonction `ProdMatVec` permettant de calculer  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .
- Donner un exemple algorithmique d'utilisation.

**Correction****R. 1**

La matrice  $\mathbb{W}$  est aussi dans  $\mathcal{M}_{m,n}(\mathbb{R})$  et on a

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket, \quad \mathbb{W}_{i,j} = \alpha\mathbb{A}_{i,j} + \beta\mathbb{B}_{i,j}.$$

Voici une correction possible

---

**Algorithme B.16** Fonction aApbB retournant la matrice  $W \stackrel{\text{def}}{=} \alpha A + \beta B$ .

---

**Données :**  $A$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $B$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\alpha$  : un réel,  
 $\beta$  : un réel.

**Résultat :**  $W$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ .

---

```

1: Fonction  $W \leftarrow \text{aApbB}(\alpha, A, \beta, B)$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $n$  faire
4:        $W(i, j) \leftarrow \alpha * A(i, j) + \beta * B(i, j)$ 
5:     Fin
6:   Fin
7: Fin

```

---

R. 2

- a. Pour que le produit de la matrice  $A \in \mathcal{M}_{m,n}(\mathbb{R})$  par un vecteur (colonne)  $u$  soit possible il faut et il suffit que la dimension du vecteur (nombre de lignes) soit égale au nombre de colonnes de  $A$ , et donc  $u \in \mathbb{R}^n$ .
- b. Avec  $u \in \mathbb{R}^n$ , on a alors  $v \stackrel{\text{def}}{=} Au \in \mathbb{R}^m$ . et

$$\forall i \in \llbracket 1, m \rrbracket, \quad v_i = \sum_{j=1}^n A_{i,j} u_j.$$

- c. Voici une manière d'obtenir l'algorithme en utilisant une méthode de raffinements successifs et les notations mathématiques pour accéder aux composantes des vecteurs et des matrices

**Algorithme B.17**  $\mathcal{R}_0$

1: Calcul de  $v$

**Algorithme B.17**  $\mathcal{R}_1$

```

1: Pour  $i = 1$  à  $m$  faire
2:    $v_i \leftarrow \sum_{j=1}^n A_{i,j} u_j$ 
3: Fin

```

**Algorithme B.17**  $\mathcal{R}_1$

1: Pour  $i \leftarrow 1$  à  $m$  faire

$$v_i \leftarrow \sum_{j=1}^n A_{i,j} u_j$$

2:  
 3: Fin

**Algorithme B.17**  $\mathcal{R}_2$

1: Pour  $i \leftarrow 1$  à  $m$  faire

```

2:    $v_i \leftarrow 0$ 
3:   Pour  $j \leftarrow 1$  à  $n$  faire
4:      $v_i \leftarrow v_i + A_{i,j} u_j$ 
5:   Fin

```

6: Fin

Pour écrire la fonction algorithmique, il nous faut passer des notations mathématiques aux notations algorithmiques pour les matrices et vecteurs. Les correspondances sont données en Table B.3.

| <i>Description</i>                                                                                                                       | <i>Mathématiques</i> | <i>Algorithmique</i> |
|------------------------------------------------------------------------------------------------------------------------------------------|----------------------|----------------------|
| Composante $i \in \llbracket 1, p \rrbracket$ de $\mathbf{x} \in \mathbb{R}^p$                                                           | $\mathbf{x}_i$       | $\mathbf{x}(i)$      |
| Composante $(i, j) \in \llbracket 1, p \rrbracket \times \llbracket 1, q \rrbracket$ de $\mathbb{Z} \in \mathcal{M}_{p,q}(\mathbb{R})$ , | $\mathbb{Z}_{i,j}$   | $\mathbb{Z}(i, j)$   |

Table B.1: Correspondance entre les notations mathématiques et algorithmiques pour les matrices et vecteurs

On obtient alors la fonction

---

**Algorithme B.17** Fonction ProdMatVec retournant le vecteur  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice  $\mathbb{A}$  par le vecteur  $\mathbf{u}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbf{u}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{v}$  : vecteur de  $\mathbb{R}^m$  tel que  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .

---

```

1: Fonction  $\mathbf{v} \leftarrow \text{ProdMatVec}(\mathbb{A}, \mathbf{u})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:      $\mathbf{v}(i) \leftarrow 0$ 
4:     Pour  $j \leftarrow 1$  à  $n$  faire
5:        $\mathbf{v}(i) \leftarrow \mathbf{v}(i) + \mathbb{A}(i, j) * \mathbf{u}(j)$ 
6:     Fin
7:   Fin
8: Fin

```

---

d. Par exemple,

$$\mathbf{x} \leftarrow \text{ProdMatVec}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\right)$$

~~~~~ Fin correction ~~~~~

### EXERCICE 2.3.6

**Q. 1** Soient  $\mathbf{u}$  et  $\mathbf{v}$  deux vecteurs de  $\mathbb{R}^n$ . Ecrire la fonction `ProdSca` permettant de retourner le produit scalaire de ces deux vecteurs noté mathématiquement  $\langle \mathbf{u}, \mathbf{v} \rangle$ .

L'objectif de la suite de l'exercice est de voir plusieurs manières de programmer le produit d'une matrice par un vecteur. Soient  $\mathbf{u} \in \mathbb{R}^p$  et  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ .

**Q. 2**

- Rappeler précisément les hypothèses et les formules permettant le calcul de  $\mathbf{v} = \mathbb{A}\mathbf{u}$ .
- Ecrire la fonction `ProdMatVec` permettant de retourner  $\mathbb{A}\mathbf{u}$ .

Voici quelques notations mathématiques (non standards) que l'on va utiliser. Si  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  alors

- $\mathbb{A}_{:,j}$  correspond au  $j$ -ème vecteur colonne de  $\mathbb{A}$ .
- $\mathbb{A}_{i,:}$  correspond au  $i$ -ème vecteur ligne de  $\mathbb{A}$ .

On suppose que l'on dispose des commandes suivantes au niveau algorithmique:

| Description   | Avec fonction   | Instruction simplifiée                     |
|---|---|--|
| Obtenir la $j$ -ème colonne d'une matrice $\mathbb{A}$ avec $j \in \llbracket 1, n \rrbracket$ , math. noté $\mathbb{A}_{:,j}$                | $\mathbf{u} \leftarrow \text{getMatCol}(\mathbb{A}, j)$             | $\mathbf{u} \leftarrow \mathbb{A}(:, j)$   |
| Obtenir la $i$ -ème ligne d'une matrice $\mathbb{A}$ avec $i \in \llbracket 1, m \rrbracket$ , math. noté $\mathbb{A}_{i,:}$                  | $\mathbf{v} \leftarrow \text{getMatRow}(\mathbb{A}, i)$             | $\mathbf{v} \leftarrow \mathbb{A}(i, :)$   |
| Remplacer la $j$ -ème colonne d'une matrice $\mathbb{A}$ par un vecteur $\mathbf{u} \in \mathbb{R}^m$ avec $j \in \llbracket 1, n \rrbracket$ | $\mathbb{A} \leftarrow \text{setMatCol}(\mathbb{A}, \mathbf{u}, j)$ | $\mathbb{A}(:, j) \leftarrow \mathbf{u}$   |
| Remplacer la $i$ -ème ligne d'une matrice $\mathbb{A}$ par un vecteur $\mathbf{v} \in \mathbb{R}^n$ avec $i \in \llbracket 1, m \rrbracket$   | $\mathbb{A} \leftarrow \text{setMatRow}(\mathbb{A}, \mathbf{v}, i)$ | $\mathbb{A}(i, :) \leftarrow \mathbf{v}$ . |

Table B.2: Instructions algorithmiques pour accéder/modifier une ligne ou une colonne d'une matrice  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ .

Q. 3

- Sous les hypothèses de Q.2 et avec  $\mathbf{v} = \mathbb{A}\mathbf{u}$ , écrire mathématiquement  $\mathbf{v}_i$  comme un produit scalaire en utilisant les notations précisées ci-dessus.
- Ecrire la fonction `ProdMatVecFun` permettant de retourner  $\mathbb{A}\mathbf{u}$  en utilisant la fonction `ProdSca` et les fonctions de la Table B.2 (i.e. sans utiliser les instructions simplifiées).
- Ecrire la fonction `ProdMatVecSim` permettant de retourner  $\mathbb{A}\mathbf{u}$  en utilisant la fonction `ProdSca` et les instructions simplifiées de la Table B.2 (i.e. sans utiliser les fonctions).

Soient  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  et  $\mathbb{B} \in \mathcal{M}_{p,q}(\mathbb{R})$ .

Q. 4

- Rappeler précisément les hypothèses et les formules permettant le calcul de  $\mathbb{G} = \mathbb{A}\mathbb{B}$ .
- Ecrire la fonction `ProdMatMat` permettant de retourner  $\mathbb{G}$ .

Q. 5

Sous les hypothèses trouvées en Q.4, on note  $\mathbb{G} = \mathbb{A}\mathbb{B}$ .

- En utilisant les notations mathématiques d'accès aux lignes et colonnes d'une matrice, écrire  $\mathbb{G}_{i,j}$  comme un produit scalaire entre un vecteur ligne de  $\mathbb{A}$  et un vecteur colonne de  $\mathbb{B}$ .
- Ecrire la fonction `ProdMatMatFun1` permettant de retourner  $\mathbb{G}$  en utilisant la formule trouvée en Q.5 a., des fonctions de la Table B.2 (i.e. sans utiliser les instructions simplifiées) et la fonction `ProdSca`.
- Ecrire la fonction `ProdMatMatSim1` permettant de retourner  $\mathbb{G}$  en utilisant la formule trouvée en Q.5 a., des instructions simplifiées de la Table B.2 et la fonction `ProdSca`.

Q. 6

Sous les hypothèses trouvées en Q.4, on note  $\mathbb{G} = \mathbb{A}\mathbb{B}$ .

- En utilisant les notations mathématiques d'accès aux lignes et colonnes d'une matrice, écrire  $\mathbb{G}_{:,j}$  ( $j$ -ème vecteur colonne de  $\mathbb{G}$ ) comme le produit de la matrice  $\mathbb{A}$  par un vecteur colonne de  $\mathbb{B}$ .
- Ecrire la fonction `ProdMatMatFun2` permettant de retourner  $\mathbb{G}$  en utilisant la formule trouvée en Q.6 a., des fonctions de la Table B.2 (i.e. sans utiliser les instructions simplifiées) et la fonction `ProdMatVecFun`.
- Ecrire la fonction `ProdMatMatSim2` permettant de retourner  $\mathbb{G}$  en utilisant la formule trouvée en Q.6 a., des instructions simplifiées de la Table B.2 et la fonction `ProdMatVecSim`.

## Correction

R. 1

On a

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n u_i v_i \in \mathbb{R}.$$

Voici une correction possible où l'on identifie la notation mathématique  $u_i$ ,  $i$ -ème composante du vecteur  $\mathbf{u}$ , et la notation algorithmique  $\mathbf{u}(i)$  permettant d'accéder à l'élément  $i$  du tableau/vecteur  $\mathbf{u}$ , avec  $i \in \llbracket 1, n \rrbracket$ .

**Algorithme B.18** Fonction ProSca, produit scalaire entre deux vecteurs.

**Données :**  $\mathbf{u}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  $\forall i \in \llbracket 1, n \rrbracket, \mathbf{u}(i) = u_i$ .  
 $\mathbf{v}$  : vecteur/tableau de  $\mathbb{R}^n$ , tel que  $\forall i \in \llbracket 1, n \rrbracket, \mathbf{v}(i) = v_i$ .

**Résultat :**  $r$  : le réel tel que  $r = \langle \mathbf{u}, \mathbf{v} \rangle$ .

```

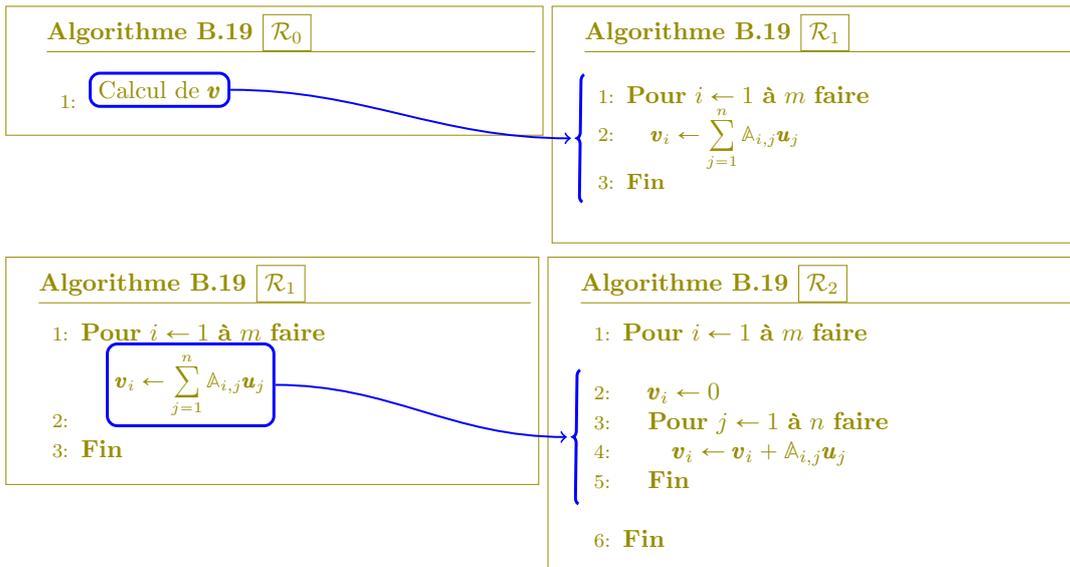
1: Fonction  $r \leftarrow \text{ProSca}(\mathbf{u}, \mathbf{v})$   $r \leftarrow 0$ 
2:   Pour  $i \leftarrow 1$  à  $n$  faire
3:      $r \leftarrow r + \mathbf{u}(i) * \mathbf{v}(i)$ 
4:   Fin
5: Fin
    
```

R. 2

- a. Pour que le produit de la matrice  $A \in \mathcal{M}_{m,n}(\mathbb{R})$  par un vecteur (colonne)  $\mathbf{u}$  soit possible il faut et il suffit que la dimension du vecteur (nombre de lignes) soit égale au nombre de colonnes de  $A$ , et donc  $\mathbf{u} \in \mathbb{R}^n$ .
- b. Avec  $\mathbf{u} \in \mathbb{R}^n$ , on a alors  $\mathbf{v} \stackrel{\text{def}}{=} A\mathbf{u} \in \mathbb{R}^m$ . et

$$\forall i \in \llbracket 1, m \rrbracket, v_i = \sum_{j=1}^n A_{i,j} u_j.$$

- c. Voici une manière d'obtenir l'algorithme en utilisant une méthode de raffinements successifs et les notations mathématiques pour accéder aux composantes des vecteurs et des matrices



Pour écrire la fonction algorithmique, il nous faut passer des notations mathématiques aux notations algorithmiques pour les matrices et vecteurs. Les correspondances sont données en Table B.3.

| Description  | Mathématiques      | Algorithmique      |
|--|--------------------|--------------------|
| Composante $i \in \llbracket 1, p \rrbracket$ de $\mathbf{x} \in \mathbb{R}^p$   | $\mathbf{x}_i$     | $\mathbf{x}(i)$    |
| Composante $(i, j) \in \llbracket 1, p \rrbracket \times \llbracket 1, q \rrbracket$ de $\mathbb{Z} \in \mathcal{M}_{p,q}(\mathbb{R})$ , | $\mathbb{Z}_{i,j}$ | $\mathbb{Z}(i, j)$ |

Table B.3: Correspondance entre les notations mathématiques et algorithmiques pour les matrices et vecteurs

On obtient alors la fonction

---

**Algorithme B.19** Fonction ProdMatVec1 retournant le vecteur  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice  $\mathbb{A}$  par le vecteur  $\mathbf{u}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbf{u}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{v}$  : vecteur de  $\mathbb{R}^m$  tel que  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .

---

```

1: Fonction  $\mathbf{v} \leftarrow \text{ProdMatVec1}(\mathbb{A}, \mathbf{u})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:      $\mathbf{v}(i) \leftarrow 0$ 
4:     Pour  $j \leftarrow 1$  à  $n$  faire
5:        $\mathbf{v}(i) \leftarrow \mathbf{v}(i) + \mathbb{A}(i, j) * \mathbf{u}(j)$ 
6:     Fin
7:   Fin
8: Fin

```

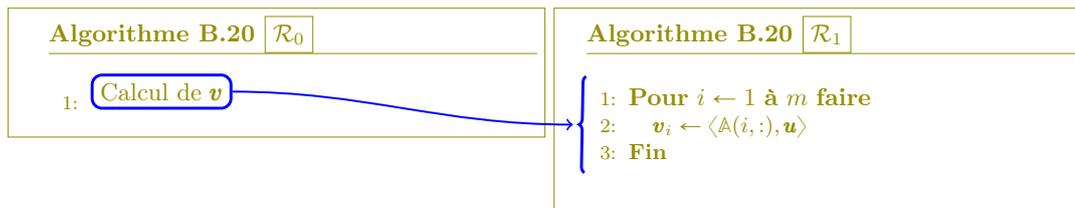
---

R. 3

- a. On peut voir le calcul de  $\mathbf{v}_i$ ,  $i \in \llbracket 1, m \rrbracket$  comme étant le produit scalaire de la  $i$ -ème ligne de  $\mathbb{A}$  par le vecteur  $\mathbf{u}$ . On a donc

$$\forall i \in \llbracket 1, m \rrbracket, \mathbf{v}_i = \langle \mathbb{A}(i, :), \mathbf{u} \rangle.$$

- b. Voici une manière d'obtenir l'algorithme en utilisant une méthode de raffinements successifs et les notations mathématiques pour accéder aux composantes des vecteurs et des matrices



On obtient alors en utilisant des fonctions de la Table B.2

---

**Algorithme B.20** Fonction ProdMatVecFun retournant le vecteur  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice  $\mathbb{A}$  par le vecteur  $\mathbf{u}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbf{u}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{v}$  : vecteur de  $\mathbb{R}^m$  tel que  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .

---

```

1: Fonction  $\mathbf{v} \leftarrow \text{ProdMatVec1}(\mathbb{A}, \mathbf{u})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:      $\mathbf{v}(i) \leftarrow \text{ProdSca}(\text{getMatRow}(\mathbb{A}, i), \mathbf{u})$ 
4:   Fin
5: Fin

```

---

c. En utilisant les instructions simplifiées de la Table B.2

**Algorithme B.21** Fonction ProdMatVecFun retournant le vecteur  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice  $\mathbb{A}$  par le vecteur  $\mathbf{u}$ .

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbf{u}$  : vecteur de  $\mathbb{R}^n$ .

**Résultat :**  $\mathbf{v}$  : vecteur de  $\mathbb{R}^m$  tel que  $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ .

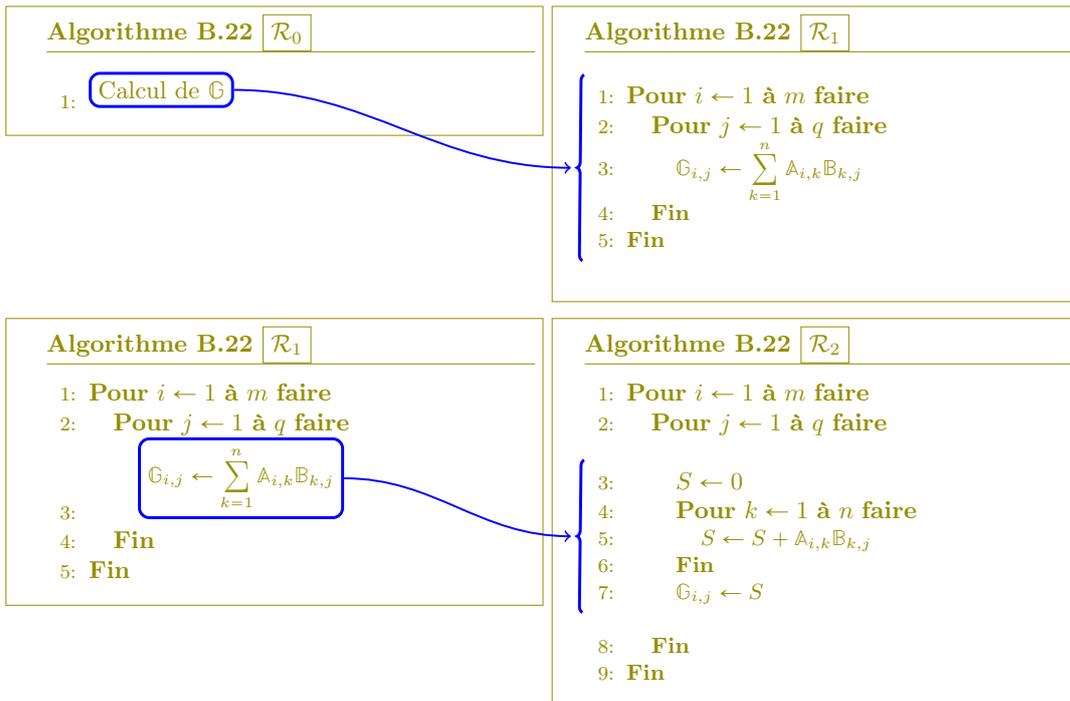
- 1: **Fonction**  $\mathbf{v} \leftarrow \text{ProdMatVec1}(\mathbb{A}, \mathbf{u})$
- 2: **Pour**  $i \leftarrow 1$  à  $m$  **faire**
- 3:      $\mathbf{v}(i) \leftarrow \text{ProdSca}(\mathbb{A}(i, :), \mathbf{u})$
- 4: **Fin**
- 5: **Fin**

R. 4

a. Le calcul du produit matricielle  $\mathbb{A}\mathbb{B}$ , avec  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  et  $\mathbb{B} \in \mathcal{M}_{p,q}(\mathbb{R})$ , est possible si et seulement si le nombre de colonnes de  $\mathbb{A}$  est égale au nombre de lignes de  $\mathbb{B}$ , c'est à dire  $n = p$ . Dans ce cas, la matrice résultat  $\mathbb{C} \in \mathcal{M}_{m,q}(\mathbb{R})$  et on a

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, q \rrbracket, \quad \mathbb{C}_{i,j} = \sum_{k=1}^n \mathbb{A}_{i,k} \mathbb{B}_{k,j}.$$

b. Voici une manière d'obtenir l'algorithme en utilisant une méthode de raffinements successifs et les notations mathématiques usuels pour accéder aux composantes des vecteurs et des matrices



On obtient alors en utilisant les notations algorithmiques

---

**Algorithme B.22** Fonction ProdMatMatFun retournant la matrice  $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice  $\mathbb{A}$  par la matrice  $\mathbb{B}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbb{B}$  : matrice de  $\mathcal{M}_{n,q}(\mathbb{R})$ .

**Résultat :**  $\mathbb{G}$  : matrice de  $\mathcal{M}_{m,q}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{G} \leftarrow \text{ProdMatMat}(\mathbb{A}, \mathbb{B})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $q$  faire
4:        $S \leftarrow 0$ 
5:       Pour  $k \leftarrow 1$  à  $n$  faire
6:          $S \leftarrow S + \mathbb{A}(i, k) * \mathbb{B}(k, j)$ 
7:       Fin
8:        $\mathbb{G}(i, j) \leftarrow S$ 
9:     Fin
10:  Fin
11: Fin

```

---

R. 5

- a. En effet la composante  $(i, j)$  de  $\mathbb{G}$  est le produit scalaire de la ligne  $i$  de  $\mathbb{A}$  et de la colonne  $j$  de  $\mathbb{B}$ . On a donc

$$\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, q \rrbracket, \quad \mathbb{G}_{i,j} = \langle \mathbb{A}_{i,:}, \mathbb{B}_{:,j} \rangle.$$

- b. Voici l'algorithme utilisant les notations mathématiques permettant d'accéder aux lignes et colonnes des matrices

**Algorithme B.23**  $\mathcal{R}_0$

---

```

1: Pour  $i \leftarrow 1$  à  $m$  faire
2:   Pour  $j \leftarrow 1$  à  $q$  faire
3:      $\mathbb{G}_{i,j} \leftarrow \langle \mathbb{A}_{i,:}, \mathbb{B}_{:,j} \rangle$ 
4:   Fin
5: Fin

```

On obtient alors en utilisant les fonctions algorithmiques de la Table B.2

---

**Algorithme B.23** Fonction ProdMatMatFun1 retournant la matrice  $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice  $\mathbb{A}$  par la matrice  $\mathbb{B}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbb{B}$  : matrice de  $\mathcal{M}_{n,q}(\mathbb{R})$ .

**Résultat :**  $\mathbb{G}$  : matrice de  $\mathcal{M}_{m,q}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{G} \leftarrow \text{ProdMatMatFun1}(\mathbb{A}, \mathbb{B})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $q$  faire
4:        $\mathbb{G}(i, j) \leftarrow \text{ProSca}(\text{getMatRow}(\mathbb{A}, i), \text{getMatCol}(\mathbb{B}, j))$ 
5:     Fin
6:   Fin
7: Fin

```

---

- c. En utilisant des instructions simplifiées de la Table B.2 et la fonction ProdSca, on a

---

**Algorithme B.24** Fonction `ProdMatMatSim1` retournant la matrice  $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice  $\mathbb{A}$  par la matrice  $\mathbb{B}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbb{B}$  : matrice de  $\mathcal{M}_{n,q}(\mathbb{R})$ .

**Résultat :**  $\mathbb{G}$  : matrice de  $\mathcal{M}_{m,q}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{G} \leftarrow \text{ProdMatMatFun1}(\mathbb{A}, \mathbb{B})$ 
2:   Pour  $i \leftarrow 1$  à  $m$  faire
3:     Pour  $j \leftarrow 1$  à  $q$  faire
4:        $\mathbb{G}(i, j) \leftarrow \text{ProSca}(\mathbb{A}(i, :), \mathbb{B}(:, j))$ 
5:     Fin
6:   Fin
7: Fin

```

---

**R. 6**

- a. En effet la colonne  $j$  de  $\mathbb{G}$  correspond au produit de la matrice  $\mathbb{A}$  par la colonne  $j$  de  $\mathbb{B}$ . On a donc

$$\forall j \in \llbracket 1, q \rrbracket, \quad \mathbb{G}_{:,j} = \mathbb{A}\mathbb{B}_{:,j}.$$

- b. Voici l'algorithme utilisant les notations mathématiques permettant d'accéder aux lignes et colonnes des matrices

**Algorithme B.25**  $\mathcal{R}_0$

---

```

1: Pour  $j \leftarrow 1$  à  $q$  faire
2:    $\mathbb{G}_{:,j} \leftarrow \mathbb{A}_{i,:}\mathbb{B}_{:,j}$ 
3: Fin

```

On obtient alors en utilisant les fonctions algorithmiques de la Table B.2

---

**Algorithme B.25** Fonction `ProdMatMatFun2` retournant la matrice  $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice  $\mathbb{A}$  par la matrice  $\mathbb{B}$ .

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbb{B}$  : matrice de  $\mathcal{M}_{n,q}(\mathbb{R})$ .

**Résultat :**  $\mathbb{G}$  : matrice de  $\mathcal{M}_{m,q}(\mathbb{R})$ .

---

```

1: Fonction  $\mathbb{G} \leftarrow \text{ProdMatMatFun2}(\mathbb{A}, \mathbb{B})$ 
2:   Pour  $j \leftarrow 1$  à  $q$  faire
3:      $\mathbb{G}(:, j) \leftarrow \text{ProdMatVecFun}(\mathbb{A}, \text{getMatCol}(\mathbb{B}, j))$ 
4:   Fin
5: Fin

```

---

- c. En utilisant les notations algorithmiques simplifiées et la fonction `ProdMatVecSim` déjà écrite, on a

**Algorithme B.26** Fonction ProdMatMatSim2 retournant la matrice  $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice  $\mathbb{A}$  par la matrice  $\mathbb{B}$ .

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_{m,n}(\mathbb{R})$ ,  
 $\mathbb{B}$  : matrice de  $\mathcal{M}_{n,q}(\mathbb{R})$ .

**Résultat :**  $\mathbb{G}$  : matrice de  $\mathcal{M}_{m,q}(\mathbb{R})$ .

1: **Fonction**  $\mathbb{G} \leftarrow \text{ProdMatMatSim2}(\mathbb{A}, \mathbb{B})$   
 2: **Pour**  $j \leftarrow 1$  à  $q$  **faire**  
 3:      $\mathbb{G}(:, j) \leftarrow \text{ProdMatVecSim}(\mathbb{A}, \mathbb{B}(:, j))$   
 4: **Fin**  
 5: **Fin**

~~~~~Fin correction~~~~~

### B.3.2 Algèbre linéaire

#### Sur les matrices

##### EXERCICE 2.3.7

Soit  $\mathbb{A} \in \mathcal{M}_{m,n}(\mathbb{R})$  et  $\mathbb{B} \in \mathcal{M}_{n,m}(\mathbb{R})$  telles que

$$\langle \mathbb{A}\mathbf{u}, \mathbf{v} \rangle_m = \langle \mathbf{u}, \mathbb{B}\mathbf{v} \rangle_n, \quad \forall \mathbf{u} \in \mathbb{R}^n, \quad \forall \mathbf{v} \in \mathbb{R}^m.$$

Exprimer les éléments de la matrice  $\mathbb{B}$  en fonction de ceux de la matrice  $\mathbb{A}$ .

##### EXERCICE 2.3.8

Soient  $\mathbb{A}$  et  $\mathbb{B}$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$  triangulaires inférieures. Montrer que la matrice  $\mathbb{A}\mathbb{B}$  est aussi triangulaire inférieure et que l'on a

$$(\mathbb{A}\mathbb{B})_{i,i} = A_{i,i}B_{i,i}, \quad \forall i \in \llbracket 1, n \rrbracket.$$

#### Correction

**Première démonstration (directe)** On note  $\mathbb{F} = \mathbb{A}\mathbb{B} \in \mathcal{M}_n(\mathbb{K})$ .

- On va démontrer que  $\mathbb{F}$  est triangulaire inférieure, c'est à dire  $F_{ij} = 0, \forall i < j$ .  
 Soit  $(i, j) \in \llbracket 1, n \rrbracket^2$  tel que  $i < j$ . On a

$$F_{ij} = \sum_{k=1}^n A_{ik}B_{kj} = \sum_{k=1}^i A_{ik} \underbrace{B_{kj}}_{=0 \text{ car } k < j} + \sum_{k=i+1}^n \underbrace{A_{ik}}_{=0 \text{ car } i < k} B_{kj} = 0$$

- Soit  $i \in \llbracket 1, n \rrbracket$ . Montrons que  $F_{ii} = A_{ii}B_{ii}$ . On a

$$F_{ii} = \sum_{k=1}^n A_{ik}B_{ki} = \sum_{k=1}^{i-1} A_{ik} \underbrace{B_{ki}}_{=0 \text{ car } k < i} + A_{ii}B_{ii} + \sum_{k=i+1}^n \underbrace{A_{ik}}_{=0 \text{ car } i < k} B_{ki} = A_{ii}B_{ii}$$

**Deuxième démonstration (récurrence sur l'ordre  $n \geq 2$ )** On veut démontrer, par récurrence faible, la proposition suivante pour  $n \geq 2$

$(\mathcal{P}_n)$ 

$\forall A \in \mathcal{M}_n(\mathbb{K}), \forall B \in \mathcal{M}_n(\mathbb{K}), A$  et  $B$  triangulaires inférieures alors  $AB$  triangulaire inférieure et

$$\forall i \in \llbracket 1, n \rrbracket, (AB)_{ii} = A_{ii}B_{ii}.$$

**Initialisation :**  $n = 2$ . On a

$$A = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}, B = \begin{pmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{pmatrix}$$

et le produit matricielle donne

$$AB = \begin{pmatrix} A_{11}B_{11} & 0 \\ \bullet & A_{22}B_{22} \end{pmatrix}.$$

La proposition  $(\mathcal{P}_2)$  est donc vérifiée.

**Hérédité :** Soit  $n > 2$ . On suppose que  $(\mathcal{P}_n)$  est vraie. Soient  $A \in \mathcal{M}_{n+1}(\mathbb{K})$  et  $B \in \mathcal{M}_{n+1}(\mathbb{K})$  triangulaires inférieures. On a en décomposition bloc diagonal carré

$$A = \begin{pmatrix} \xrightarrow{n} & & \\ \underline{A} & \mathbf{0}_{n,1} & \\ \hline \mathbf{e}^* & f & \end{pmatrix} \text{ et } B = \begin{pmatrix} \xrightarrow{n} & & \\ \underline{B} & \mathbf{0}_{n,1} & \\ \hline \mathbf{g}^* & h & \end{pmatrix}$$

où  $\underline{A}$  et  $\underline{B} \in \mathcal{M}_n(\mathbb{K})$  sont tri. inf.,  $\mathbf{e} \in \mathbb{K}^n, \mathbf{g} \in \mathbb{K}^n, f \in \mathbb{K}$  et  $h \in \mathbb{K}$ . On a alors

$$A = \begin{pmatrix} & & \\ \underline{AB} & \mathbf{0}_{n,1} & \\ \hline \mathbf{e}^*\underline{B} + f\mathbf{g}^* & fh & \end{pmatrix}.$$

Comme  $\underline{A}$  et  $\underline{B}$  sont tri. inf. d'ordre  $n$ , par hypothèse de récurrence  $(\mathcal{P}_n)$ , on a  $\underline{AB}$  tri. inf. et,

$$\forall i \in \llbracket 1, n \rrbracket, (\underline{AB})_{ii} = (\underline{A})_{ii}(\underline{B})_{ii} = A_{ii}B_{ii}.$$

De plus, on a

$$(\underline{AB})_{n+1,n+1} = fh = A_{n+1,n+1}B_{n+1,n+1}.$$

On en déduit que la proposition  $(\mathcal{P}_{n+1})$  est vérifiée.

**Conclusion :** On a démontré par récurrence faible que  $(\mathcal{P}_n)$  est vraie  $\forall n \geq 2$ .

~~~~~ Fin correction ~~~~~

### EXERCICE 2.3.9

Soient  $A$  et  $B$  deux matrices triangulaires supérieures de  $\mathcal{M}_n$ . Soient  $E$  et  $F$  deux matrices triangulaires inférieures de  $\mathcal{M}_n$ .

Q. 1

- Que peut-on dire des matrices  $A^*$  et  $(A^*)^*$ ?
- Montrer que  $C = AB$  est triangulaire supérieure et que  $C_{i,i} = A_{i,i}B_{i,i}, \forall i \in \llbracket 1, n \rrbracket$ .
- Montrer que  $G = EF$  est triangulaire inférieure et que  $G_{i,i} = E_{i,i}F_{i,i}, \forall i \in \llbracket 1, n \rrbracket$ .
- Que peut-on dire des matrices  $AE$  et  $EA$ ?

Q. 2

- Calculer  $\det(A)$ .
- Déterminer les valeurs propres de  $A$ .
- Que peut-on dire si les éléments diagonaux de  $A$  sont tous distincts?

Q. 3

Soit  $D$  la matrice définie par

$$D = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}.$$

- La matrice  $D$  est-elle inversible? Si oui calculer son inverse.
- Pour chacune des valeurs propres, déterminer l'espace propre associé.
- La matrice  $D$  est-elle diagonalisable? Justifier.

**EXERCICE 2.3.10**

Q. 1

Montrer par récurrence sur l'ordre des matrices, que si une matrice est normale et triangulaire supérieure alors elle est diagonale.

On rappelle le théorème de décomposition de Schur:

**Théorème**

Soit  $A \in \mathcal{M}_n(\mathbb{C})$ , il existe une matrice unitaire  $U \in \mathcal{M}_n(\mathbb{C})$  et une matrice triangulaire inférieure  $T \in \mathcal{M}_n(\mathbb{C})$  telles que

$$A = UTU^*. \quad (\text{B.1})$$

Q. 2

Montrer que  $A \in \mathcal{M}_n(\mathbb{C})$  est une matrice normale si et seulement si il existe  $U \in \mathcal{M}_n(\mathbb{C})$  unitaire et  $D \in \mathcal{M}_n(\mathbb{C})$  diagonale telle que  $A = UDU^*$ .

Q. 3

En déduire qu'une matrice normale est diagonalisable et que ses vecteurs propres sont orthogonaux.

**Correction**

R. 1

On va démontrer, par récurrence, que la propriété suivante est vérifiée pour tout  $n \geq 2$ :

$$(P_n) : \forall T \in \mathcal{M}_n(\mathbb{C}) \text{ normale et triangulaire supérieure} \Rightarrow T \text{ diagonale}$$

**Initialisation :**  $n = 2$ . Soit  $T \in \mathcal{M}_2(\mathbb{C})$  une matrice normale et triangulaire supérieure. Comme  $T$  est normale, on a  $TT^* = T^*T$ , ce qui entraîne

$$(TT^*)_{i,j} = (T^*T)_{i,j}, \quad \forall (i,j) \in \llbracket 1, 2 \rrbracket^2.$$

ou encore

$$\begin{pmatrix} T_{1,1} & T_{1,2} \\ 0 & T_{2,2} \end{pmatrix} \begin{pmatrix} \overline{T_{1,1}} & 0 \\ \overline{T_{1,2}} & \overline{T_{2,2}} \end{pmatrix} = \begin{pmatrix} \overline{T_{1,1}} & 0 \\ \overline{T_{1,2}} & \overline{T_{2,2}} \end{pmatrix} \begin{pmatrix} T_{1,1} & T_{1,2} \\ 0 & T_{2,2} \end{pmatrix}.$$

On a donc

$$\begin{aligned} |T_{1,1}|^2 + |T_{1,2}|^2 &= |T_{1,1}|^2, \\ T_{1,2}\overline{T_{2,2}} &= \overline{T_{1,1}}T_{1,2}, \\ T_{2,2}\overline{T_{1,2}} &= \overline{T_{1,2}}T_{1,1}, \\ |T_{2,2}|^2 &= |T_{1,2}|^2 + |T_{2,2}|^2. \end{aligned}$$

De la première équation ou de la dernière, on obtient  $T_{1,2} = 0$ . Les deux autres équations reviennent à  $0 = 0$ , ce qui laisse libre les paramètres  $T_{1,1}$  et  $T_{2,2}$ .

La matrice  $\mathbb{T}$  est donc diagonale. La propriété  $(\mathcal{P}_2)$  est vérifiée.

**Hérédité :** Soit  $n > 2$ . On suppose que  $(\mathcal{P}_{n-1})$  est vraie. Soit  $\mathbb{T} \in \mathcal{M}_n(\mathbb{C})$  une matrice normale et triangulaire supérieure. On décompose la matrice  $\mathbb{T}$  en une matrice bloc  $2 \times 2$  dont le deuxième bloc diagonal, noté  $\mathbb{U}$  est dans  $\mathcal{M}_{n-1}(\mathbb{C})$

$$\mathbb{T} = \left( \begin{array}{c|c} T_{1,1} & \mathbf{c}^* \\ \hline 0 & \mathbb{U} \\ \vdots & \\ 0 & \end{array} \right)$$

avec  $\mathbf{c}^* = (T_{1,2}, \dots, T_{1,n}) \in \mathbb{C}^{n-1}$ . On peut noter que  $\mathbb{U}$  est triangulaire supérieure. On a alors  $\mathbb{T}\mathbb{T}^* = \mathbb{T}^*\mathbb{T}$  qui s'écrit sous forme bloc :

$$\left( \begin{array}{c|c} T_{1,1} & \mathbf{c}^* \\ \hline 0 & \mathbb{U} \\ \vdots & \\ 0 & \end{array} \right) \left( \begin{array}{c|c} \overline{T_{1,1}} & 0 \dots 0 \\ \hline \mathbf{c} & \mathbb{U}^* \end{array} \right) = \left( \begin{array}{c|c} \overline{T_{1,1}} & 0 \dots 0 \\ \hline \mathbf{c} & \mathbb{U}^* \end{array} \right) \left( \begin{array}{c|c} T_{1,1} & \mathbf{c}^* \\ \hline 0 & \mathbb{U} \\ \vdots & \\ 0 & \end{array} \right)$$

c'est à dire

$$\left( \begin{array}{c|c} |T_{1,1}|^2 + \mathbf{c}^*\mathbf{c} & \mathbf{c}^*\mathbb{U}^* \\ \hline \mathbb{U}\mathbf{c} & \mathbb{U}\mathbb{U}^* \end{array} \right) = \left( \begin{array}{c|c} |T_{1,1}|^2 & \overline{T_{1,1}}\mathbf{c}^* \\ \hline T_{1,1}\mathbf{c} & \mathbf{c}\mathbf{c}^* + \mathbb{U}^*\mathbb{U} \end{array} \right).$$

En identifiant bloc par bloc, on obtient

$$\begin{aligned} |T_{1,1}|^2 + \|\mathbf{c}\|_2^2 &= |T_{1,1}|^2 \\ \mathbf{c}^*\mathbb{U}^* &= \overline{T_{1,1}}\mathbf{c}^* \\ \mathbb{U}\mathbf{c} &= T_{1,1}\mathbf{c} \\ \mathbb{U}\mathbb{U}^* &= \mathbf{c}\mathbf{c}^* + \mathbb{U}^*\mathbb{U} \end{aligned}$$

De la première équation, bloc (1, 1), on obtient

$$\mathbf{c} = \mathbf{0}.$$

De la dernière équation, le bloc (2, 2), on déduit  $\mathbb{U}$  normale. Les deux autres équations s'*auto-détruisent*. Comme  $\mathbb{U} \in \mathcal{M}_{n-1}(\mathbb{C})$  est normale et triangulaire supérieure, on obtient, par hypothèse de récurrence, que  $\mathbb{U}$  est diagonale.

**R. 2**

On rappelle le théorème

**Théorème B.3.1.** Soit  $A \in \mathcal{M}_n(\mathbb{C})$ , il existe une matrice unitaire  $U \in \mathcal{M}_n(\mathbb{C})$  et une matrice triangulaire inférieure  $T \in \mathcal{M}_n(\mathbb{C})$  telles que

$$A = UTU^*. \quad (\text{R10.18})$$

$\Leftarrow$  Montrons que si  $A$  est une matrice normale alors il existe une matrice unitaire  $U \in \mathcal{M}_n(\mathbb{C})$  et une matrice diagonale  $D \in \mathcal{M}_n(\mathbb{C})$  telles que  $A = UDU^*$ .

En effet, d'après le théorème B.3.1, il existe une matrice unitaire  $U \in \mathcal{M}_n(\mathbb{C})$  et une matrice triangulaire inférieure  $T \in \mathcal{M}_n(\mathbb{C})$  telles que  $A = UTU^*$ . Il suffit, grâce à la première question, de démontrer que la matrice  $T$  est normale. On a

$$\begin{aligned} A \text{ normale} &\Leftrightarrow A^*A = AA^* \\ &\Leftrightarrow (UT^*U^*)(UTU^*) = (UTU^*)(UT^*U^*) \end{aligned}$$

Or la matrice  $U$  est unitaire,  $U^* = U^{-1}$ , on a donc

$$A \text{ normale} \Leftrightarrow UT^*TU^* = UTT^*U^*$$

En multipliant cette équation par  $U^*$  à gauche et par  $U$  à droite, on obtient

$$\begin{aligned} A \text{ normale} &\Leftrightarrow T^*T = TT^* \\ &\Leftrightarrow T \text{ normale.} \end{aligned}$$

D'après la question 1, la matrice  $T$  est donc diagonale.

$\Rightarrow$  Montrons que si il existe une matrice unitaire  $U \in \mathcal{M}_n(\mathbb{C})$  et une matrice diagonale  $D \in \mathcal{M}_n(\mathbb{C})$  telles que  $A = UDU^*$  alors  $A$  est une matrice normale.

En effet, on a

$$\begin{aligned} A^*A &= (UD^*U^*)(UDU^*) \\ &= UD^*DU^* \text{ car } U \text{ unitaire} \end{aligned}$$

Comme la matrice  $D$  est diagonale, elle est normale :  $DD^* = D^*D$ . On obtient alors

$$\begin{aligned} A^*A &= UDD^*U^* \\ &= (UDU^*)(UD^*U^*) \text{ car } U \text{ unitaire} \\ &= AA^*. \end{aligned}$$

La matrice  $A$  est donc normale.

**R. 3**

On a

$$A = UDU^* = UDU^{-1}.$$

La matrice  $A$  est donc diagonalisable.

Notons  $(\lambda_1, \dots, \lambda_n)$  la diagonale de  $D$  et  $\mathbf{u}_i$  le  $i$ -ème vecteur colonne de la matrice  $U$ .

Comme  $U$  est unitaire,  $U^*U = \mathbb{I}$ , et que

$$U^*U = \begin{pmatrix} \langle \mathbf{u}_1, \mathbf{u}_1 \rangle & \dots & \langle \mathbf{u}_1, \mathbf{u}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{u}_n, \mathbf{u}_1 \rangle & \dots & \langle \mathbf{u}_n, \mathbf{u}_n \rangle \end{pmatrix}$$

on obtient

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \delta_{i,j}, \quad \forall (i, j) \in \llbracket 1, n \rrbracket^2$$

De plus, on a

$$U^*\mathbf{u}_j = \begin{pmatrix} \langle \mathbf{u}_1, \mathbf{u}_j \rangle \\ \vdots \\ \langle \mathbf{u}_n, \mathbf{u}_j \rangle \end{pmatrix} = \mathbf{e}_j,$$

où  $\mathbf{e}_j$  est le  $j$ -ème vecteur de la base canonique.

On a donc

$$\mathbb{A}\mathbf{u}_j = \mathbb{U}\mathbb{D}\mathbb{U}^*\mathbf{u}_j = \mathbb{U}\mathbb{D}\mathbf{e}_j = \mathbb{U}(\lambda_j\mathbf{e}_j) = \lambda_j\mathbf{u}_j.$$

On en déduit que  $\forall j \in \llbracket 1, n \rrbracket$ ,  $(\lambda_j, \mathbf{u}_j)$  est un élément propre de  $\mathbb{A}$  et les vecteurs propres de  $\mathbb{A}$  sont orthogonaux.

Fin correction

### EXERCICE 2.3.11

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice hermitienne

Q. 1 Montrer que

$$\langle \mathbb{A}\mathbf{u}, \mathbf{u} \rangle \in \mathbb{R}, \quad \forall \mathbf{u} \in \mathbb{C}^n. \quad (\text{B.1})$$

On suppose de plus que la matrice  $\mathbb{A}$  est définie positive.

Q. 2

- Montrer que les éléments diagonaux de  $\mathbb{A}$  sont strictement positifs.
- Montrer que les sous matrices principales de  $\mathbb{A}$  sont elles aussi hermitiennes et définies positives.

### EXERCICE 2.3.12 : Décomposition de Schur

Soit  $\mathbb{A} \in \mathcal{M}_{n,n}(\mathbb{C})$  une matrice et  $(\lambda, \mathbf{u})$  un élément propre de  $\mathbb{A}$  avec  $\|\mathbf{u}\|_2 = 1$ .

Q. 1

- En s'aidant de la base canonique  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ , construire une base orthonormée  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  telle que  $\mathbf{x}_1 = \mathbf{u}$ .
- Ecrire la fonction algorithmique `BaseOrtho` permettant de construire la base orthonormée  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  à partir d'un vecteur  $\mathbf{u}$  donné. Les vecteurs  $\mathbf{x}_k$  seront stockés dans une matrice de  $\mathcal{M}_{n,n}(\mathbb{C})$ , le vecteur  $\mathbf{x}_k$  étant en colonne  $k$  de la matrice. On pourra pour cela utiliser les fonctions prédéfinies  $s \leftarrow \text{dot}(\mathbf{u}, \mathbf{v})$  qui retourne le produit scalaire de deux vecteurs,  $s \leftarrow \text{abs}(x)$  qui retourne le module d'un nombre complexe, ...

Notons par  $\mathbb{P}$  la matrice de  $\mathcal{M}_{n,n}(\mathbb{C})$  définie par

$$\mathbb{P} = \left( \begin{array}{c|c|c} & & \\ \mathbf{x}_1 & \dots & \mathbf{x}_n \\ & & \end{array} \right)$$

et par  $\mathbb{B}$  la matrice définie par  $\mathbb{B} = \mathbb{P}^*\mathbb{A}\mathbb{P}$ .

Q. 2

- Calculer la matrice  $\mathbb{P}^*\mathbb{P}$ . Que peut-on en conclure?
- Exprimer les coefficients de la matrice  $\mathbb{B}$  en fonction de la matrice  $\mathbb{A}$  et des vecteurs  $\mathbf{x}_i$ ,  $i \in \llbracket 1, n \rrbracket$ .

$$\mathbb{B} = \mathbb{P}^*\mathbb{A}\mathbb{P}.$$

- En déduire que la première colonne de  $\mathbb{B}$  est  $(\lambda, 0, \dots, 0)^t$ .

Q. 3

Montrer par récurrence sur l'ordre de la matrice que la matrice  $\mathbb{A}$  s'écrit

$$\mathbb{A} = \mathbb{U}\mathbb{T}\mathbb{U}^*$$

où  $\mathbb{U}$  est une matrice unitaire et  $\mathbb{T}$  une matrice triangulaire supérieure.

Q. 4

On dispose des fonctions  $\mathbf{x} \leftarrow \text{ResTriSup}(\mathbb{S}, \mathbf{b})$  et  $\mathbf{x} \leftarrow \text{ResTriInf}(\mathbb{L}, \mathbf{b})$  permettant de résoudre respectivement le système triangulaire supérieur  $\mathbb{S}\mathbf{x} = \mathbf{b}$  et le système triangulaire inférieur  $\mathbb{L}\mathbf{x} = \mathbf{b}$ . En supposant  $\mathbb{A}$  inversible et la décomposition  $\mathbb{A} = \mathbb{U}\mathbb{T}\mathbb{U}^*$  connue, expliquez comment résoudre "simplement" le système linéaire  $\mathbb{A}\mathbf{x} = \mathbf{b}$ .

## Correction

R. 1

- a. La première chose à faire est de construire une base contenant  $\mathbf{u}$  à partir de la base canonique  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ . Comme le vecteur propre  $\mathbf{u}$  est non nul, il existe  $j \in \llbracket 1, n \rrbracket$  tel que  $\langle \mathbf{u}, \mathbf{e}_j \rangle \neq 0$ . La famille  $\{\mathbf{u}, \mathbf{e}_1, \dots, \mathbf{e}_{j-1}, \mathbf{e}_{j+1}, \dots, \mathbf{e}_n\}$  forme alors une base de  $\mathbb{C}^n$  car  $\mathbf{u}$  n'est pas combinaison linéaire des  $\{\mathbf{e}_1, \dots, \mathbf{e}_{j-1}, \mathbf{e}_{j+1}, \dots, \mathbf{e}_n\}$ .

On note  $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$  la base dont le premier élément est  $\mathbf{z}_1 = \mathbf{u}$  :

$$\{\mathbf{z}_1, \dots, \mathbf{z}_n\} = \{\mathbf{u}, \mathbf{e}_1, \dots, \mathbf{e}_{j-1}, \mathbf{e}_{j+1}, \dots, \mathbf{e}_n\}.$$

On peut ensuite utiliser le **procédé de Gram-Schmidt**, rappelé en Proposition B.2.1, pour construire une base orthonormée à partir de cette base.

On calcule successivement les vecteurs  $\mathbf{x}_i$  à partir de la base  $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$  en construisant un vecteur  $\mathbf{w}_i$  orthogonal aux vecteurs  $\mathbf{x}_1, \dots, \mathbf{x}_{i-1}$ .

$$\mathbf{w}_i = \mathbf{z}_i - \sum_{k=1}^{i-1} \langle \mathbf{z}_i, \mathbf{x}_k \rangle \mathbf{x}_k$$

puis on obtient le vecteur  $\mathbf{x}_i$  en normalisant

$$\mathbf{x}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}$$

- b. Nous allons tout d'abord rechercher un indice  $i_{\max}$  tel que  $|u_{i_{\max}}| = \max_{i \in \llbracket 1, n \rrbracket} |u_i|$ . Ensuite, nous construisons la matrice  $\mathbb{Z} \in \mathcal{M}_{n,n}(\mathbb{C})$  telle que la colonne  $i$  de  $\mathbb{Z}$  contienne le vecteur  $\mathbf{z}_i$  avec

$$\{\mathbf{z}_1, \dots, \mathbf{z}_n\} = \{\mathbf{u}, \mathbf{e}_1, \dots, \mathbf{e}_{i_{\max}-1}, \mathbf{e}_{i_{\max}+1}, \dots, \mathbf{e}_n\}.$$

Ensuite, à l'aide des formules précédentes nous calculons la matrice  $\mathbb{P} \in \mathcal{M}_{n,n}(\mathbb{C})$  telle que la colonne  $i$  de  $\mathbb{Z}$  contienne le vecteur  $\mathbf{x}_i$ . Voici donc l'algorithme:

**Algorithme B.27**

**Données :**  $\mathbf{u}$  : un vecteur de  $\mathbb{C}^n$  tel que  $\|\mathbf{u}\|_2 = 1$ .

**Résultat :**  $\mathbb{P}$  : matrice  $n \times n$  dont les vecteurs colonnes forment une base orthonormée de  $\mathbb{C}^n$  et telle que la première colonne de  $\mathbb{P}$  soit  $\mathbf{u}$ .

```

1: Fonction  $\mathbb{P} \leftarrow \text{BaseOrtho}(\mathbf{u})$ 
2:    $umax \leftarrow 0, i_{max} \leftarrow 0$  ▷ On calcule  $i_{max}$ 
3:   Pour  $i \leftarrow 1$  à  $n$  faire
4:     Si  $umax \leq \text{abs}(\mathbf{u}(i))$  alors
5:        $umax \leftarrow \text{abs}(\mathbf{u}(i)), i_{max} \leftarrow i$ 
6:     Fin
7:   Fin
8:    $\mathbb{Z} \leftarrow \text{zeros}(n, n)$  ▷ On calcule  $\mathbb{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$ 
9:    $\mathbb{Z}(:, 1) \leftarrow \mathbf{u}$ 
10:   $i \leftarrow 2$ 
11:  Pour  $k \leftarrow 1$  à  $i_{max} - 1$  faire
12:     $\mathbb{Z}(k, i) \leftarrow 1$ 
13:     $i \leftarrow i + 1$ 
14:  Fin
15:  Pour  $k \leftarrow i_{max} + 1$  à  $n$  faire
16:     $\mathbb{Z}(k, i) \leftarrow 1$ 
17:     $i \leftarrow i + 1$ 
18:  Fin
19:   $\mathbb{P} \leftarrow \text{zeros}(n, n)$  ▷ On calcule  $\mathbb{P} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ 
20:  Pour  $i \leftarrow 1$  à  $n$  faire
21:     $\mathbb{P}(:, i) \leftarrow \mathbb{Z}(:, i)$ 
22:    Pour  $k \leftarrow 1$  à  $i - 1$  faire
23:       $\mathbb{P}(:, i) \leftarrow \mathbb{P}(:, i) - \text{dot}(\mathbb{P}(:, k), \mathbb{Z}(:, i)) * \mathbb{P}(:, k)$ 
24:    Fin
25:     $\mathbb{P}(:, i) \leftarrow \mathbb{P}(:, i) / \text{sqrt}(\text{dot}(\mathbb{P}(:, i), \mathbb{P}(:, i)))$ 
26:  Fin
27: Fin

```

**R. 2**

a. Notons  $\mathbb{S} = \mathbb{P}^* \mathbb{P}$  la matrice de  $\mathcal{M}_{n,n}(\mathbb{C})$ . On a pour tout  $(i, j) \in \llbracket 1, n \rrbracket^2$

$$S_{i,j} = \langle \mathbb{P}(:, i), \mathbb{P}(:, j) \rangle = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \delta_{i,j}.$$

On a donc

$$\mathbb{P}^* \mathbb{P} = \mathbb{I}$$

et la matrice  $\mathbb{P}$  est unitaire.

b. En conservant l'écriture colonne de la matrice  $\mathbb{P}$  on obtient

$$\mathbb{B} = \begin{pmatrix} \mathbf{x}_1^* \\ \mathbf{x}_2^* \\ \vdots \\ \mathbf{x}_n^* \end{pmatrix} \wedge \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^* \\ \mathbf{x}_2^* \\ \vdots \\ \mathbf{x}_n^* \end{pmatrix} \wedge \begin{pmatrix} \wedge \mathbf{x}_1 & \wedge \mathbf{x}_2 & \dots & \wedge \mathbf{x}_n \end{pmatrix}$$

Ce qui donne

$$\mathbb{B} = \begin{pmatrix} \mathbf{x}_1^* \wedge \mathbf{x}_1 & \mathbf{x}_1^* \wedge \mathbf{x}_2 & \dots & \mathbf{x}_1^* \wedge \mathbf{x}_n \\ \mathbf{x}_2^* \wedge \mathbf{x}_1 & \mathbf{x}_2^* \wedge \mathbf{x}_2 & \dots & \mathbf{x}_2^* \wedge \mathbf{x}_n \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_n^* \wedge \mathbf{x}_1 & \mathbf{x}_n^* \wedge \mathbf{x}_2 & \dots & \mathbf{x}_n^* \wedge \mathbf{x}_n \end{pmatrix}$$

On a donc

$$B_{i,j} = \mathbf{x}_i^* \mathbb{A} \mathbf{x}_j, \quad \forall (i,j) \in \llbracket 1, n \rrbracket^2$$

c. On a  $\mathbb{A} \mathbf{u} = \lambda \mathbf{u}$ ,  $\|\mathbf{u}\| = 1$ , la base  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  est orthonormée et  $\mathbf{x}_1 = \mathbf{u}$ . on obtient alors

$$\mathbb{B} = \begin{pmatrix} \lambda \mathbf{u}^* \mathbf{u} & \mathbf{u}^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{u}^* \mathbb{A} \mathbf{x}_n \\ \lambda \mathbf{x}_2^* \mathbf{u} & \mathbf{x}_2^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{x}_2^* \mathbb{A} \mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda \mathbf{x}_n^* \mathbf{u} & \mathbf{x}_n^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{x}_n^* \mathbb{A} \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \lambda & \mathbf{u}^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{u}^* \mathbb{A} \mathbf{x}_n \\ 0 & \mathbf{x}_2^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{x}_2^* \mathbb{A} \mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathbf{x}_n^* \mathbb{A} \mathbf{x}_2 & \dots & \mathbf{x}_n^* \mathbb{A} \mathbf{x}_n \end{pmatrix}$$

**R. 3**

On veut démontrer, par récurrence faible, la proposition suivante pour  $n \geq 2$

$(\mathcal{P}_n)$   $\forall \mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ ,  $\exists \mathbb{U} \in \mathcal{M}_n(\mathbb{C})$  unitaire,  $\exists \mathbb{T} \in \mathcal{M}_n(\mathbb{C})$  triangulaire supérieure, telles que  $\mathbb{A} = \mathbb{U} \mathbb{T} \mathbb{U}^*$ .

**Initialisation :** Montrons que  $(\mathcal{P}_2)$  est vérifié.

Soit  $\mathbb{A}_2 \in \mathcal{M}_2(\mathbb{C})$ . Elle admet au moins un élément propre  $(\lambda, \mathbf{u})$  (voir Proposition B.2.8 par ex.) avec  $\|\mathbf{u}\| = 1$ . On peut donc appliquer le résultat de la question précédente : il existe une matrice unitaire  $\mathbb{P}_2 \in \mathcal{M}_2(\mathbb{C})$  telle que la matrice  $\mathbb{B}_2 = \mathbb{P}_2 \mathbb{A}_2 \mathbb{P}_2^*$  ait comme premier vecteur colonne  $(\lambda, 0)^t$ . La matrice  $\mathbb{B}_2$  est donc triangulaire supérieure et comme  $\mathbb{P}_2$  est unitaire on en déduit

$$\mathbb{A}_2 = \mathbb{P}_2^* \mathbb{B}_2 \mathbb{P}_2.$$

On pose  $\mathbb{U}_2 = \mathbb{P}_2^*$  matrice unitaire et  $\mathbb{T}_2 = \mathbb{B}_2$  matrice triangulaire supérieure pour conclure que la proposition  $(\mathcal{P}_2)$  est vraie.

**Hérédité :** Supposons que  $(\mathcal{P}_{n-1})$  soit vérifiée. Montrons que  $(\mathcal{P}_n)$  est vraie.

Soit  $\mathbb{A}_n \in \mathcal{M}_n(\mathbb{C})$ . Elle admet au moins un élément propre  $(\lambda, \mathbf{u})$  (voir Proposition B.2.8 par ex.) avec  $\|\mathbf{u}\| = 1$ . On peut donc appliquer le résultat de la question précédente : il existe une matrice unitaire  $\mathbb{P}_n \in \mathcal{M}_n(\mathbb{C})$  telle que la matrice  $\mathbb{B}_n = \mathbb{P}_n \mathbb{A}_n \mathbb{P}_n^*$  s'écrive

$$\mathbb{B}_n = \begin{pmatrix} \lambda & \mathbf{c}_{n-1}^* \\ 0 & \mathbb{A}_{n-1} \\ \vdots & \\ 0 & \end{pmatrix}$$

où  $\mathbf{c}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$  et  $\mathbb{A}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$ . Par hypothèse de récurrence,  $\exists \mathbb{U}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$  unitaire et  $\mathbb{T}_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C})$  triangulaire supérieure telles que

$$\mathbb{A}_{n-1} = \mathbb{U}_{n-1} \mathbb{T}_{n-1} \mathbb{U}_{n-1}^*$$

ou encore

$$\mathbb{T}_{n-1} = \mathbb{U}_{n-1}^* \mathbb{A}_{n-1} \mathbb{U}_{n-1}.$$

Soit  $\mathbb{Q}_n \in \mathcal{M}_n(\mathbb{C})$  la matrice définie par

$$\mathbb{Q}_n = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \mathbb{U}_{n-1} & \\ 0 & & & \end{pmatrix}.$$

La matrice  $\mathbb{Q}_n$  est unitaire. En effet on a

$$\mathbb{Q}_n \mathbb{Q}_n^* = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \mathbb{U}_{n-1} & \\ 0 & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \mathbb{U}_{n-1}^* & \\ 0 & & & \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \underbrace{\mathbb{U}_{n-1} \mathbb{U}_{n-1}^*}_{=\mathbb{I}_{n-1}} & \\ 0 & & & \end{pmatrix} = \mathbb{I}_n.$$

On note  $\mathbb{T}_n$  la matrice définie par  $\mathbb{T}_n = \mathbb{Q}_n^* \mathbb{B}_n \mathbb{Q}_n$ . On a alors

$$\begin{aligned} \mathbb{T}_n &= \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & \mathbb{U}_{n-1}^* & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} \lambda & & & \mathbf{c}_{n-1}^* \\ 0 & & & \\ \vdots & \mathbb{A}_{n-1} & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \mathbb{U}_{n-1} \\ 0 & & & \end{pmatrix} \\ &= \begin{pmatrix} \lambda & & & \mathbf{c}_{n-1}^* \\ 0 & & & \\ \vdots & \mathbb{U}_{n-1}^* \mathbb{A}_{n-1} & & \\ 0 & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \mathbb{U}_{n-1} \\ 0 & & & \end{pmatrix} = \begin{pmatrix} \lambda & & & \mathbf{c}_{n-1}^* \mathbb{U}_{n-1}^* \\ 0 & & & \\ \vdots & \underbrace{\mathbb{U}_{n-1}^* \mathbb{A}_{n-1} \mathbb{U}_{n-1}}_{=\mathbb{T}_{n-1}} & & \\ 0 & & & \end{pmatrix} \end{aligned}$$

La matrice  $\mathbb{T}_n$  est donc triangulaire supérieure et on a par définition de  $\mathbb{B}_n$

$$\mathbb{T}_n = \mathbb{Q}_n^* \mathbb{P}_n \mathbb{A}_n \mathbb{P}_n^* \mathbb{Q}_n.$$

On note  $\mathbb{U}_n = \mathbb{P}_n^* \mathbb{Q}_n$ . Cette matrice est unitaire car les matrices  $\mathbb{Q}_n$  et  $\mathbb{P}_n$  le sont. En effet, on a

$$\mathbb{U}_n \mathbb{U}_n^* = \mathbb{P}_n^* \mathbb{Q}_n (\mathbb{P}_n^* \mathbb{Q}_n)^* = \mathbb{P}_n^* \underbrace{\mathbb{Q}_n \mathbb{Q}_n^*}_{=\mathbb{I}_n} \mathbb{P}_n = \mathbb{P}_n^* \mathbb{P}_n = \mathbb{I}_n.$$

On a  $\mathbb{T}_n = \mathbb{U}_n^* \mathbb{A}_n \mathbb{U}_n$  et en multipliant cette équation à gauche par  $\mathbb{U}_n$  et à droite par  $\mathbb{U}_n^*$  on obtient l'équation équivalente  $\mathbb{A}_n = \mathbb{U}_n \mathbb{T}_n \mathbb{U}_n^*$ . La propriété  $(\mathcal{P}_n)$  est donc vérifiée. Ce qui achève la démonstration.

**R. 4**

Résoudre  $\mathbb{A}\mathbf{x} = \mathbf{b}$  est équivalent à résoudre

$$\mathbb{U}\mathbb{T}\mathbb{U}^*\mathbf{x} = \mathbf{b}. \tag{R12.19}$$

Comme  $\mathbb{U}$  est unitaire, on a  $\mathbb{U}\mathbb{U}^* = \mathbb{I}$  et  $\mathbb{U}^*$  inversible. Donc en multipliant (R12.19) par  $\mathbb{U}^*$  on obtient le système équivalent

$$\underbrace{\mathbb{U}^*\mathbb{U}}_{=\mathbb{I}} \mathbb{T}\mathbb{U}^*\mathbf{x} = \mathbb{U}^*\mathbf{b} \iff \mathbb{T}\mathbb{U}^*\mathbf{x} = \mathbb{U}^*\mathbf{b}.$$

On pose  $\mathbf{y} = \mathbb{U}^*\mathbf{x}$ . Le système précédant se résout en deux étapes

- a. on cherche  $\mathbf{y}$  solution de  $\mathbb{T}\mathbf{y} = \mathbb{U}^*\mathbf{b}$ . Comme  $\mathbb{U}$  est unitaire on a  $\det(\mathbb{U}) \det(\mathbb{U}^*) = \det(\mathbb{I}) = 1$  et donc

$$\begin{aligned} \det(\mathbb{A}) &= \det(\mathbb{U}\mathbb{T}\mathbb{U}^*) = \det(\mathbb{U}) \det(\mathbb{T}) \det(\mathbb{U}^*) \\ &= \det(\mathbb{T}) \end{aligned}$$

Or  $\mathbb{A}$  inversible équivaut à  $\det(\mathbb{A}) \neq 0$  et donc la matrice  $\mathbb{T}$  est inversible. La matrice  $\mathbb{T}$  étant triangulaire inférieure on peut résoudre facilement le système par la *méthode de remontée*.

- b. une fois  $\mathbf{y}$  déterminé, on résout  $\mathbb{U}^*\mathbf{x} = \mathbf{y}$ . Comme  $\mathbb{U}$  est unitaire, on obtient directement  $\mathbf{x} = \mathbb{U}\mathbf{y}$ .

~~~~~ Fin correction ~~~~~

**EXERCICE 2.3.13 : Procédé de Gram-Schmidt**

Soit  $\{\mathbf{v}_i\}_{i \in [1, n]}$  une base de  $\mathbb{K}^n$ . On construit successivement les vecteurs  $\mathbf{u}_i$

$$\mathbf{u}_i = \mathbf{v}_i - \sum_{k=1}^{i-1} \frac{\langle \mathbf{v}_i, \mathbf{u}_k \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \mathbf{u}_k, \quad \forall i \in [1, n].$$

Montrer qu'ils forment une **base orthogonale** de  $\mathbb{K}^n$  et que  $\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) = \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_i)$ ,

$$\forall i \in \llbracket 1, n \rrbracket.$$

**Correction** Montrons par récurrence sur  $i$  que

$$(\mathcal{H})_i : \text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) \text{ est une famille orthogonale et } \text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) = \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_i)$$

**Initialisation :** Pour  $i = 1$ , on a  $\mathbf{u}_1 = \mathbf{v}_1$  et  $(\mathcal{H})_1$  est vérifiée.

**Hérédité :** Soit  $i < n$ . Supposons  $(\mathcal{H})_i$  vérifiée. Montrons alors que  $(\mathcal{H})_{i+1}$  est vraie.

On a

$$\mathbf{u}_{i+1} = \mathbf{v}_{i+1} - \sum_{k=1}^i \frac{\langle \mathbf{u}_k, \mathbf{v}_{i+1} \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \mathbf{u}_k. \quad (\text{B.54})$$

- En effectuant le produit scalaire de (B.54) par  $\mathbf{u}_j$  avec  $j \in \llbracket 1, i \rrbracket$  on obtient

$$\langle \mathbf{u}_j, \mathbf{u}_{i+1} \rangle = \langle \mathbf{u}_j, \mathbf{v}_{i+1} \rangle - \sum_{k=1}^i \frac{\langle \mathbf{u}_k, \mathbf{v}_{i+1} \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \langle \mathbf{u}_j, \mathbf{u}_k \rangle.$$

Par hypothèse de récurrence, la famille  $\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i)$  est orthogonale, c'est à dire  $\forall (r, s) \in \llbracket 1, i \rrbracket^2$ ,  $\langle \mathbf{u}_r, \mathbf{u}_s \rangle = 0$  si  $r \neq s$  et  $\mathbf{u}_r \neq 0$ . On obtient donc

$$\langle \mathbf{u}_j, \mathbf{u}_{i+1} \rangle = \langle \mathbf{u}_j, \mathbf{v}_{i+1} \rangle - \frac{\langle \mathbf{u}_j, \mathbf{v}_{i+1} \rangle}{\langle \mathbf{u}_j, \mathbf{u}_j \rangle} \langle \mathbf{u}_j, \mathbf{u}_j \rangle = 0, \quad \forall j \in \llbracket 1, i \rrbracket.$$

- On montre maintenant par l'absurde que  $\mathbf{u}_{i+1} \neq 0$ .  
Supposons  $\mathbf{u}_{i+1} = 0$ . Alors de (B.54), on obtient

$$\mathbf{v}_{i+1} = \sum_{k=1}^i \frac{\langle \mathbf{u}_k, \mathbf{v}_{i+1} \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \mathbf{u}_k$$

et donc  $\mathbf{v}_{i+1} \in \text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) \stackrel{(\mathcal{H})_i}{=} \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_i)$ . Ceci entre en contradiction avec  $\text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_n)$  base de  $\mathbb{K}^n$ .

- On déduit de (B.54) que  $\mathbf{u}_{i+1} \in \text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i, \mathbf{v}_{i+1})$ . Par hypothèse de récurrence,  $\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) = \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_i)$ , ce qui donne  $\mathbf{u}_{i+1} \in \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_{i+1})$  et donc

$$\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_{i+1}) = \text{Vect}(\mathbf{v}_1, \dots, \mathbf{v}_{i+1}).$$

~~~~~ Fin correction ~~~~~

#### EXERCICE 2.3.14 : factorisation $\mathbb{QR}$

Soit  $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$  une matrice inversible. Pour tout  $i \in \llbracket 1, n \rrbracket$ , on note  $\mathbf{a}_i = \mathbb{A}_{:,i}$  ses  $n$  vecteurs colonnes. En utilisant le procédé de Gram-Schmidt sur la base  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  montrer qu'il existe une matrice  $\mathbb{Q}$  unitaire et une matrice triangulaire supérieure  $\mathbb{R}$  à coefficients diagonaux strictement positifs tel que  $\mathbb{A} = \mathbb{QR}$ .

**Correction** On utilise le procédé d'orthonormalisation de Gram-Schmidt (voir Proposition B.2.1, page 220) pour obtenir la base orthogonale  $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$  en calculant successivement

$$\mathbf{u}_i = \mathbf{a}_i - \sum_{k=1}^{i-1} \frac{\langle \mathbf{u}_k, \mathbf{a}_i \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \mathbf{u}_k, \quad \forall i \in \llbracket 1, n \rrbracket. \quad (\text{B.55})$$

De plus on a  $\text{Vect}(\mathbf{u}_1, \dots, \mathbf{u}_i) = \text{Vect}(\mathbf{a}_1, \dots, \mathbf{a}_i), \forall i \in \llbracket 1, n \rrbracket$ . On normalise la base orthogonale  $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$  pour obtenir la base orthonormée  $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$  :

$$\mathbf{q}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}, \forall i \in \llbracket 1, n \rrbracket$$

et l'on a aussi  $\text{Vect}(\mathbf{q}_1, \dots, \mathbf{q}_i) = \text{Vect}(\mathbf{a}_1, \dots, \mathbf{a}_i), \forall i \in \llbracket 1, n \rrbracket$ .

On note  $\mathbb{Q} \in \mathcal{M}_n(\mathbb{K})$  la matrice définie par

$$\mathbb{Q} = \begin{pmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ | & & | \end{pmatrix}$$

Cette matrice est clairement unitaire puisque la base  $\{\mathbf{q}_1, \dots, \mathbf{q}_n\}$  est orthonormée. Montrons que  $\mathbb{Q}^* \mathbb{A}$  est triangulaire supérieure. On a

$$\mathbb{Q}^* \mathbb{A} = \begin{pmatrix} \mathbf{q}_1^* \\ \vdots \\ \mathbf{q}_n^* \end{pmatrix} \begin{pmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_n \\ | & & | \end{pmatrix} = \begin{pmatrix} \langle \mathbf{q}_1, \mathbf{a}_1 \rangle & \cdots & \langle \mathbf{q}_1, \mathbf{a}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{q}_n, \mathbf{a}_1 \rangle & \cdots & \langle \mathbf{q}_n, \mathbf{a}_n \rangle \end{pmatrix}$$

c'est à dire  $(\mathbb{Q}^* \mathbb{A})_{i,j} = \langle \mathbf{q}_i, \mathbf{a}_j \rangle, \forall (i,j) \in \llbracket 1, n \rrbracket$ . Par définition cette matrice est triangulaire supérieure si  $(\mathbb{Q}^* \mathbb{A})_{i,j} = 0$  pour tout  $i > j$ . Soit  $i \in \llbracket 1, n-1 \rrbracket$ . La base  $\mathcal{Q}$  étant orthonormée, on a  $\mathbf{q}_i \perp \text{Vect}(\mathbf{q}_1, \dots, \mathbf{q}_{i-1})$ . Comme  $\text{Vect}(\mathbf{q}_1, \dots, \mathbf{q}_{i-1}) = \text{Vect}(\mathbf{a}_1, \dots, \mathbf{a}_{i-1})$ , on en déduit que

$$\langle \mathbf{q}_i, \mathbf{a}_j \rangle = 0, \forall j \in \llbracket 1, i-1 \rrbracket.$$

La matrice  $\mathbb{Q}^* \mathbb{A}$  est donc triangulaire supérieure.

De plus, on a

$$(\mathbb{Q}^* \mathbb{A})_{i,i} = \langle \mathbf{q}_i, \mathbf{a}_i \rangle = \frac{\langle \mathbf{u}_i, \mathbf{a}_i \rangle}{\|\mathbf{u}_i\|_2}$$

En prenant le produit scalaire de (B.55) avec  $\mathbf{u}_i$  on obtient

$$\begin{aligned} \langle \mathbf{u}_i, \mathbf{u}_i \rangle &= \langle \mathbf{u}_i, \mathbf{a}_i \rangle - \sum_{k=1}^{i-1} \frac{\langle \mathbf{u}_k, \mathbf{a}_i \rangle}{\langle \mathbf{u}_k, \mathbf{u}_k \rangle} \langle \mathbf{u}_i, \mathbf{u}_k \rangle \\ &= \langle \mathbf{u}_i, \mathbf{a}_i \rangle \text{ car } \langle \mathbf{u}_i, \mathbf{u}_k \rangle = 0, \forall k \neq i \text{ (base orthogonale)} \end{aligned}$$

Comme  $\mathbf{u}_i \neq 0$ , on obtient  $(\mathbb{Q}^* \mathbb{A})_{i,i} > 0$ .

On note  $\mathbb{R} = \mathbb{Q}^* \mathbb{A}$  cette matrice triangulaire supérieure avec  $R_{i,i} > 0, \forall i \in \llbracket 1, n \rrbracket$ . La matrice  $\mathbb{Q}$  étant unitaire (i.e.  $\mathbb{Q} \mathbb{Q}^* = \mathbb{I}$ ) alors  $\mathbb{A} = \mathbb{Q} \mathbb{R}$ .

~~~~~ Fin correction ~~~~~

## Inverse d'une matrice

### EXERCICE 2.3.15

Q. 1

Soit  $\mathbb{A}$  une matrice inversible et symétrique, montrer que  $\mathbb{A}^{-1}$  est symétrique.

Q. 2

Soit  $\mathbb{A}$  une matrice carrée telle que  $\mathbb{I} - \mathbb{A}$  est inversible. Montrer que

$$\mathbb{A}(\mathbb{I} - \mathbb{A})^{-1} = (\mathbb{I} - \mathbb{A})^{-1} \mathbb{A}.$$

Q. 3

Soient  $\mathbb{A}, \mathbb{B}$  des matrices carrées inversibles de même dimension telle que  $\mathbb{A} + \mathbb{B}$  soit inversible. Montrer que

$$\mathbb{A}(\mathbb{A} + \mathbb{B})^{-1} \mathbb{B} = \mathbb{B}(\mathbb{A} + \mathbb{B})^{-1} \mathbb{A} = (\mathbb{A}^{-1} + \mathbb{B}^{-1})^{-1}$$

**EXERCICE 2.3.16**

Soient  $\mathbb{A}$  et  $\mathbb{B}$ , deux matrices de  $\mathcal{M}_n(\mathbb{K})$ .

**Q. 1**

Montrer que

$$\mathbb{A}\mathbb{B} = \mathbb{I} \Rightarrow \mathbb{B}\mathbb{A} = \mathbb{I} \quad (\text{B.1})$$

Conclure.

**EXERCICE 2.3.17**

Soit  $\mathbb{A} \in \mathcal{M}_3(\mathbb{R})$  définie par

$$\mathbb{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

**Q. 1**

Calculer le déterminant de la matrice  $\mathbb{A}$ . Que peut-on en conclure?

**Q. 2**

Calculer si possible l'inverse de la matrice  $\mathbb{A}$  en utilisant la technique de la matrice augmentée.

**EXERCICE 2.3.18**

Soit  $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$  une matrice triangulaire inférieure.

**Q. 1**

A quelle(s) condition(s) la matrice  $\mathbb{L}$  est-elle inversible?

On suppose  $\mathbb{L}$  inversible et on note  $\mathbb{X} = \mathbb{L}^{-1}$ .

**Q. 2**

Montrer que  $\mathbb{X}$  est une matrice triangulaire inférieure avec

$$X_{i,i} = \frac{1}{L_{i,i}}, \quad \forall i \in \llbracket 1, n \rrbracket.$$

**Correction****R. 1**

La matrice  $\mathbb{L}$  est inversible si et seulement si son déterminant est non nul. Or le déterminant d'une matrice triangulaire est égal au produit de ses éléments diagonaux. Pour avoir  $\mathbb{L}$ , matrice triangulaire, inversible, il faut et il suffit que

$$L_{ii} \neq 0, \quad \forall i \in \llbracket 1, n \rrbracket.$$

**R. 2**

La matrice  $\mathbb{M}$  est la matrice inverse de  $\mathbb{L}$  donc

$$\mathbb{L}\mathbb{M} = \mathbb{I}$$

c'est à dire

$$\sum_{k=1}^n L_{i,k} M_{k,j} = \delta_{i,j}, \quad \forall (i, j) \in \llbracket 1, n \rrbracket. \quad (\text{R18.20})$$

Montrons par **récurrence forte finie** que,  $\forall i \in \llbracket 1, n \rrbracket$ , la proposition suivante est vérifiée

$$\mathcal{P}(i) : M_{i,j} = 0, \quad \forall j \in \llbracket i+1, n \rrbracket \text{ et } M_{i,i} = \frac{1}{L_{i,i}}.$$

- Montrons que  $\mathcal{P}(1)$  est vraie.  
Soit  $j \in \llbracket 2, n \rrbracket$ . On a, d'après (R18.20) en  $i = 1$ ,

$$0 = (\mathbb{L}\mathbb{M})_{1,j} = \sum_{k=1}^n L_{1,k} M_{k,j} = L_{1,1} M_{1,j} + \sum_{k=2}^n L_{1,k} M_{k,j}.$$

Comme la matrice  $\mathbb{L}$  est triangulaire inférieure, on a

$$\forall k \in \llbracket 2, n \rrbracket, L_{1,k} = 0$$

et donc

$$L_{1,1}M_{1,j} = 0.$$

Or  $L_{1,1} \neq 0$  d'après **Q.1**, on obtient alors

$$M_{1,j} = 0.$$

D'après (R18.20) en  $i = j = 1$ , on a

$$1 = \sum_{k=1}^n L_{1,k}M_{k,1} = L_{1,1}M_{1,1} + \sum_{k=2}^n L_{1,k}M_{k,1}.$$

Or  $L_{1,k} = 0, \forall k \in \llbracket 2, n \rrbracket$ , car  $\mathbb{L}$  triangulaire inférieure. On en déduit  $L_{1,1}M_{1,1} = 1$  et donc, comme  $L_{1,1} \neq 0$

$$M_{1,1} = \frac{1}{L_{1,1}}.$$

L'assertion  $\mathcal{P}(1)$  est donc vraie.

- Soit  $i \in \llbracket 1, n-1 \rrbracket$ . On suppose que  $\mathcal{P}(k)$  est vérifiée  $\forall k \in \llbracket 1, i \rrbracket$ . Montrons que  $\mathcal{P}(i+1)$  est vérifiée.

Soit  $j \in \llbracket i+2, n \rrbracket$ . On a, d'après (R18.20) en  $(i+1, j)$

$$0 = \sum_{k=1}^n L_{i+1,k}M_{k,j} = \sum_{k=1}^i L_{i+1,k}M_{k,j} + L_{i+1,i+1}M_{i+1,j} + \sum_{k=i+2}^n L_{i+1,k}M_{k,j}.$$

Or  $L_{i+1,k} = 0, \forall k \in \llbracket i+2, n \rrbracket$ , car  $\mathbb{L}$  triangulaire inférieure et  $M_{k,j} = 0 \forall k \in \llbracket 1, i \rrbracket$  par hypothèse de récurrence. On a alors

$$L_{i+1,i+1}M_{i+1,j} = 0.$$

Comme  $L_{i+1,i+1} \neq 0$  d'après **Q.1**, on a

$$M_{i+1,j} = 0.$$

D'après (R18.20) en  $(i+1, i+1)$  on a

$$1 = \sum_{k=1}^n L_{i+1,k}M_{k,i+1} = \sum_{k=1}^i L_{i+1,k}M_{k,i+1} + L_{i+1,i+1}M_{i+1,i+1} + \sum_{k=i+2}^n L_{i+1,k}M_{k,i+1}.$$

Or  $L_{i+1,k} = 0, \forall k \in \llbracket i+2, n \rrbracket$ , car  $\mathbb{L}$  triangulaire inférieure et  $M_{k,i+1} = 0 \forall k \in \llbracket 1, i \rrbracket$  par hypothèse de récurrence. On a alors

$$L_{i+1,i+1}M_{i+1,i+1} = 1.$$

Comme  $L_{i+1,i+1} \neq 0$  d'après **Q.1**, on a

$$M_{i+1,i+1} = \frac{1}{L_{i+1,i+1}}.$$

L'assertion  $\mathcal{P}(i+1)$  est donc vraie.

On a donc démontré que  $\mathcal{P}(i)$  est vraie  $\forall i \in \llbracket 1, n \rrbracket$ . La matrice  $\mathbb{M}$  est donc triangulaire inférieure et

$$M_{i,i} = \frac{1}{L_{i,i}}, \forall i \in \llbracket 1, n \rrbracket.$$

~~~~~ Fin correction ~~~~~

**EXERCICE 2.3.19**

Soit  $A \in \mathcal{M}_{n,n}(\mathbb{K})$  et  $U, B, V$  trois matrices rectangulaires.

**Q. 1** Sous quelles hypothèses les plus générales possibles peut-on définir la matrice  $G$  suivante

$$G = A^{-1} - A^{-1}U (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1}?$$

**Q. 2** Montrer que  $(A + UB)V G = \mathbb{I}$ . Conclure.

**Q. 3** Soit  $\beta \in \mathbb{R}$  et  $u, v \in \mathbb{C}^n$ . Calculer  $(A + \beta uv^t)^{-1}$  en fonction de l'inverse de  $A$ .

**Correction****R. 1**

Pour que la matrice  $G$  soit bien définie il est nécessaire et suffisant que

- la matrice  $A$  soit inversible,
- la matrice  $\mathbb{I} + BVA^{-1}U$  soit inversible (et donc carrée).
- les sommes et produits matriciels dans la définition de  $G$  soient compatibles.

Précisons ce dernier point. Les matrices  $U, V$  et  $B$  sont rectangulaires : on a  $U \in \mathcal{M}_{l,m}(\mathbb{K})$ ,  $B \in \mathcal{M}_{p,q}(\mathbb{K})$  et  $V \in \mathcal{M}_{r,s}(\mathbb{K})$ . Pour définir  $\mathbb{I} + BVA^{-1}U$ , qui est une matrice carrée, il faut avoir compatibilité des produits :

$$BV \Rightarrow q = r, VA^{-1} \Rightarrow s = n, A^{-1}U \Rightarrow l = n.$$

Ensuite pour que la matrice soit carrée il faut que  $BVA^{-1}U$  le soit : c'est à dire  $p = m$ . On a donc

$$U \in \mathcal{M}_{n,m}(\mathbb{K}), B \in \mathcal{M}_{m,q}(\mathbb{K}), V \in \mathcal{M}_{q,n}(\mathbb{K})$$

et alors  $\mathbb{I} + BVA^{-1}U \in \mathcal{M}_{m,m}(\mathbb{K})$ . On en déduit

$$A^{-1}U \in \mathcal{M}_{n,m}(\mathbb{K}) \text{ et } BVA^{-1} \in \mathcal{M}_{m,n}(\mathbb{K})$$

ce qui donne

$$A^{-1}U (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1} \in \mathcal{M}_{n,n}(\mathbb{K}).$$

Pour résumer, la matrice  $G$  est bien définie si et seulement si

- la matrice  $A$  est inversible,
- $U \in \mathcal{M}_{n,m}(\mathbb{K}), B \in \mathcal{M}_{m,q}(\mathbb{K}), V \in \mathcal{M}_{q,n}(\mathbb{K})$
- la matrice  $\mathbb{I} + BVA^{-1}U$  est inversible.

**R. 2**

hypotheses précédentes, on a

$$\begin{aligned} (A + UB)V G &= (A + UB)V \left( A^{-1} - A^{-1}U (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1} \right) \\ &= \mathbb{I} + UBVA^{-1} - AA^{-1}U (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1} - UBVA^{-1}U (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1} \\ &= \mathbb{I} + UBVA^{-1} - U (\mathbb{I} + BVA^{-1}U) (\mathbb{I} + BVA^{-1}U)^{-1} BVA^{-1} \\ &= \mathbb{I} + UBVA^{-1} - UBVA^{-1} \\ &= \mathbb{I}. \end{aligned}$$

On a donc

$$(\mathbb{A} + \mathbb{U}\mathbb{B}\mathbb{V})^{-1} = \mathbb{G}. \quad (\text{R19.21})$$

**R. 3**

On prend  $m = q = 1$ ,  $\mathbb{B} = \beta$ ,  $\mathbb{U} = \mathbf{u}$  et  $\mathbb{V} = \mathbf{v}^*$ . On a alors

$$(\mathbb{A} + \beta \mathbf{u}\mathbf{v}^*)^{-1} = \mathbb{A}^{-1} - \mathbb{A}^{-1} \mathbf{u} (\mathbb{I} + \beta \mathbf{v}^* \mathbb{A}^{-1} \mathbf{u})^{-1} \beta \mathbf{v}^* \mathbb{A}^{-1}$$

Or  $\mathbb{I} + \beta \mathbf{v}^* \mathbb{A}^{-1} \mathbf{u} \in \mathbb{K}$ , donc

$$(\mathbb{A} + \beta \mathbf{u}\mathbf{v}^*)^{-1} = \mathbb{A}^{-1} - \frac{\beta}{1 + \beta \mathbf{v}^* \mathbb{A}^{-1} \mathbf{u}} \mathbb{A}^{-1} \mathbf{u}\mathbf{v}^* \mathbb{A}^{-1}.$$

~~~~~ Fin correction ~~~~~

### EXERCICE 2.3.20

Etant donnée une matrice  $\mathbb{D} \in \mathcal{M}_{n,n}(\mathbb{C})$ , on pose

$$\mathbb{D} = \mathbb{A} + \imath \mathbb{B} \text{ avec } \mathbb{A}, \mathbb{B} \in \mathcal{M}_{n,n}(\mathbb{R})$$

Sous certaines hypothèses à préciser, établir la relation

$$\mathbb{D}^{-1} = (\mathbb{A} + \mathbb{B}\mathbb{A}^{-1}\mathbb{B})^{-1} - \imath \mathbb{A}^{-1} \mathbb{B} (\mathbb{A} + \mathbb{B}\mathbb{A}^{-1}\mathbb{B})^{-1}.$$

### Correction

~~~~~ Fin correction ~~~~~

### EXERCICE 2.3.21 : Matrice de Vandermonde

Soient  $(z_i)_{i=0}^n$   $n + 1$  points distincts 2 à 2 de  $\mathbb{C}$ . Soit  $\mathbb{V} \in \mathcal{M}_{n+1}(\mathbb{C})$  la matrice définie par

$$\mathbb{V}_{i,j} = z_{i-1}^{j-1}, \quad \forall (i, j) \in \llbracket 1, n+1 \rrbracket.$$

**Q. 1**

Ecrire la matrice  $\mathbb{V}$ .

Soient  $\mathbf{w} = (w_i)_{i=1}^{n+1}$  un vecteur de  $\mathbb{C}^{n+1}$ . On note  $P_{\mathbf{w}} \in \mathbb{C}_n[X]$ , le polynôme défini par

$$P_{\mathbf{w}}(z) = \sum_{i=0}^n w_{i+1} z^i$$

**Q. 2**

Exprimer  $\mathbf{v} = \mathbb{V}\mathbf{w}$  en fonction de  $P_{\mathbf{w}}$ .

**Q. 3**

En déduire que  $\mathbb{V}$  est inversible.

### Correction

R. 1

On a

$$\mathbb{V} = \begin{pmatrix} 1 & z_0 & \cdots & z_0^n \\ 1 & z_1 & \cdots & z_1^n \\ \vdots & \vdots & & \vdots \\ 1 & z_n & \cdots & z_n^n \end{pmatrix}$$

R. 2

On a  $\mathbf{v} \in \mathbb{C}^{n+1}$  et, pour tout  $i \in \llbracket 1, n+1 \rrbracket$ ,

$$\begin{aligned} v_i &= \sum_{j=1}^{n+1} \mathbb{V}_{i,j} w_j \\ &= \sum_{j=1}^{n+1} z_{i-1}^{j-1} w_j \\ &= \sum_{j=0}^n w_{j+1} z_{i-1}^j = P_{\mathbf{w}}(z_{i-1}). \end{aligned}$$

c'est à dire

$$\mathbf{v} = \begin{pmatrix} P_{\mathbf{w}}(z_0) \\ \vdots \\ P_{\mathbf{w}}(z_n) \end{pmatrix}.$$

R. 3

La matrice  $\mathbb{V}$  est inversible si et seulement si son noyau est réduit à l'élément nul, c'est à dire

$$\ker(\mathbb{V}) = \{\mathbf{0}\}.$$

Soit  $\mathbf{u} = (u_1, \dots, u_{n+1})^* \in \mathbb{C}^{n+1}$ , tel que  $\mathbb{V}\mathbf{u} = \mathbf{0}$ , montrons qu'alors  $\mathbf{u} = \mathbf{0}$ .

On a

$$\mathbb{V}\mathbf{u} = \mathbb{V} \begin{pmatrix} u_1 \\ \vdots \\ u_{n+1} \end{pmatrix} = \begin{pmatrix} P_{\mathbf{u}}(z_0) \\ \vdots \\ P_{\mathbf{u}}(z_n) \end{pmatrix} = \mathbf{0}.$$

Les  $n+1$  points  $(z_i)_{i=0}^n$  sont distincts 2 à 2, donc le polynôme  $P_{\mathbf{u}}$  admet  $n+1$  racines distinctes hors  $P_{\mathbf{u}} \in \mathbb{C}_n[X]$ , c'est donc le polynôme nul, c'est à dire  $u_i = 0, \forall i \in \llbracket 1, n+1 \rrbracket$ . On a donc  $\mathbf{u} = \mathbf{0}$ .La matrice  $\mathbb{V}$  est donc inversible.

~~~~~ Fin correction ~~~~~

## Matrices blocs

## EXERCICE 2.3.22

On considère les matrices blocs suivantes

$$\mathbb{A} = \left( \begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right) = \left( \begin{array}{c|c} \mathbb{C} & \mathbb{I} \\ \hline \mathbb{I} & \mathbb{0} \end{array} \right) \quad \text{et} \quad \mathbb{B} = \left( \begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \end{array} \right) = \left( \begin{array}{c|c} \mathbb{I} & \mathbb{0} \\ \hline \mathbb{C} & \mathbb{C} \end{array} \right)$$

avec par identification

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad \mathbb{C} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Q. 1

Calculer les matrices  $\mathbb{A}\mathbb{B}$  et  $\mathbb{B}\mathbb{A}$  en utilisant l'écriture bloc.

Q. 2

Exprimer les matrices  $A(A+B)$  et  $(2B-A)(B+A)$  en fonction des matrices  $C$  et  $I$ .

**EXERCICE 2.3.23**

Soient  $A \in \mathcal{M}_{n,k}(\mathbb{K})$  et  $B \in \mathcal{M}_{k,n}(\mathbb{K})$ . On note  $L$  la matrice

$$L = \left( \begin{array}{c|c} I - BA & B \\ \hline 2A - ABA & AB - I \end{array} \right).$$

Q. 1

Montrer que la matrice  $L$  est bien définie et spécifier les dimensions des blocs.

Q. 2

Calculer  $L^2$ . Que peut-on en conclure?

**EXERCICE 2.3.24**

Soient  $A \in \mathcal{M}_m(\mathbb{C})$ ,  $B \in \mathcal{M}_n(\mathbb{C})$  et  $D \in \mathcal{M}_{m,n}(\mathbb{C})$ .

Q. 1

Calculer, en fonction des déterminant de  $A$  et  $B$ , le déterminant des matrices

$$E = \left( \begin{array}{c|c} A & 0 \\ \hline 0 & I_n \end{array} \right), \quad F = \left( \begin{array}{c|c} I_m & 0 \\ \hline 0 & B \end{array} \right), \quad \text{et } G = \left( \begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array} \right).$$

Q. 2

Soit  $H = \left( \begin{array}{c|c} A & D \\ \hline 0 & B \end{array} \right)$ . En utilisant les factorisations QR des matrices  $A$  et  $B$ , montrer que

$$\det(H) = \det(A) \det(B). \quad (\text{B.1})$$

Q. 3

En déduire qu'une matrice triangulaire supérieure par blocs est inversible si et seulement si ses matrices blocs diagonales sont inversibles.

Q. 4

En déduire qu'une matrice triangulaire inférieure par blocs est inversible si et seulement si ses matrices blocs diagonales sont inversibles.

**Correction**

R. 1

On a

$$\det E = \det \left( \begin{array}{c|c} A & 0 \\ \hline 0 & I_n \end{array} \right) = \det A$$

et

$$\det F = \det \left( \begin{array}{c|c} I_m & 0 \\ \hline 0 & B \end{array} \right) = \det B.$$

De plus, on a

$$G = \left( \begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array} \right) = \left( \begin{array}{c|c} A & 0 \\ \hline 0 & I_n \end{array} \right) \left( \begin{array}{c|c} I_m & 0 \\ \hline 0 & B \end{array} \right)$$

et donc

$$\det G = \det(EF) \underset{\text{prop. det}}{=} \det(E) \det(F) = \det A \times \det B.$$

R. 2

D'après le théorème de factorisation QR, on a

- $\exists Q_A$  unitaire et  $R_A$  triangulaire supérieure telles que  $A = Q_A R_A$ ,
- $\exists Q_B$  unitaire et  $R_B$  triangulaire supérieure telles que  $B = Q_B R_B$

Comme  $Q_A$  est unitaire (i.e.  $Q_A Q_A^* = \mathbb{I}$ ), on peut noter alors que

$$\left( \begin{array}{c|c} Q_A & 0 \\ \hline 0 & Q_B \end{array} \right) \left( \begin{array}{c|c} R_A & Q_A^* D \\ \hline 0 & R_B \end{array} \right) = \left( \begin{array}{c|c} A & D \\ \hline 0 & B \end{array} \right) = \mathbb{H}.$$

Les matrices  $R_A$  et  $R_B$  étant triangulaires supérieures, on en déduit que la matrice  $\left( \begin{array}{c|c} R_A & Q_A^* D \\ \hline 0 & R_B \end{array} \right)$  est aussi triangulaire supérieure et son déterminant est le produit de ses éléments diagonaux:

$$\det \left( \begin{array}{c|c} R_A & Q_A^* D \\ \hline 0 & R_B \end{array} \right) = \prod_{i=1}^m (R_A)_{ii} \prod_{j=1}^n (R_B)_{jj} = \det(R_A) \det(R_B).$$

D'après la Q.1, on a

$$\det \left( \begin{array}{c|c} Q_A & 0 \\ \hline 0 & Q_B \end{array} \right) = \det(Q_A) \det(Q_B).$$

Comme les matrices  $Q_A$  et  $Q_B$  sont unitaires, on a  $\det(Q_A) = \det(Q_B) = 1$  d'où

$$\det \left( \begin{array}{c|c} Q_A & 0 \\ \hline 0 & Q_B \end{array} \right) = 1.$$

On obtient en déduit alors

$$\det(\mathbb{H}) = \det \left( \begin{array}{c|c} Q_A & 0 \\ \hline 0 & Q_B \end{array} \right) \det \left( \begin{array}{c|c} R_A & Q_A^* D \\ \hline 0 & R_B \end{array} \right) = \det(R_A) \det(R_B).$$

De plus, par propriété du déterminant et comme les matrices unitaires ont pour déterminant 1, on obtient  $\det A = \det(Q_A) \det(R_A) = \det(R_A)$  et  $\det B = \det(Q_B) \det(R_B) = \det(R_B)$ , ce qui donne

$$\det(\mathbb{H}) = \det(A) \det(B).$$

~~~~~Fin correction~~~~~

### B.3.3 Normes

#### Normes vectorielles

##### EXERCICE 2.3.25

Soient  $\mathbf{x}$  et  $\mathbf{y}$  deux vecteurs de  $\mathbb{C}^n$ .

**Q. 1** Trouver  $\alpha \in \mathbb{C}$  tel que  $\langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{x} \rangle = 0$ .

**Q. 2** En calculant  $\|\alpha \mathbf{x} - \mathbf{y}\|_2^2$ , montrer que

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2. \quad (\text{B.1})$$

**Q. 3** Soit  $\mathbf{x} \neq 0$ . Montrer alors que l'inégalité (B.1) est une égalité si et seulement si  $\mathbf{y} = \alpha \mathbf{x}$ .

#### Correction

**R. 1**

- Si  $\mathbf{x} = 0$ , alors  $\alpha$  quelconque.
- Si  $\mathbf{x} \neq 0$ , alors

$$\langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{x} \rangle = 0 \iff \bar{\alpha} \langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{y}, \mathbf{x} \rangle = 0$$

Or  $\mathbf{x} \neq 0$ , ce qui donne

$$\bar{\alpha} = \frac{\langle \mathbf{y}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}.$$

et, comme  $\langle \mathbf{x}, \mathbf{x} \rangle \in \mathbb{R}$  et  $\overline{\langle \mathbf{y}, \mathbf{x} \rangle} = \langle \mathbf{x}, \mathbf{y} \rangle$ , on obtient

$$\alpha = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}. \quad (\text{R25.22})$$

**R. 2**

On a

$$\begin{aligned} \|\alpha \mathbf{x} - \mathbf{y}\|_2^2 &= \langle \alpha \mathbf{x} - \mathbf{y}, \alpha \mathbf{x} - \mathbf{y} \rangle \\ &= \alpha \langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{x} \rangle - \langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{y} \rangle \\ &= -\langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{y} \rangle, \quad \text{car } \langle \alpha \mathbf{x} - \mathbf{y}, \mathbf{x} \rangle = 0 \\ &= -\bar{\alpha} \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle \end{aligned}$$

En utilisant (R25.22), on obtient alors

$$\begin{aligned} \|\alpha \mathbf{x} - \mathbf{y}\|_2^2 &= -\frac{\langle \mathbf{y}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle \\ &= \frac{-\langle \mathbf{y}, \mathbf{x} \rangle \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle \langle \mathbf{x}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \end{aligned}$$

Comme  $\langle \mathbf{y}, \mathbf{x} \rangle = \overline{\langle \mathbf{x}, \mathbf{y} \rangle}$ , on a  $\langle \mathbf{y}, \mathbf{x} \rangle \langle \mathbf{x}, \mathbf{y} \rangle = |\langle \mathbf{x}, \mathbf{y} \rangle|^2$  et donc

$$\begin{aligned} \|\alpha \mathbf{x} - \mathbf{y}\|_2^2 &= \frac{1}{\langle \mathbf{x}, \mathbf{x} \rangle} \left( -|\langle \mathbf{x}, \mathbf{y} \rangle|^2 + \|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2 \right) \\ &\geq 0. \end{aligned} \quad (\text{R25.23})$$

On a alors

$$|\langle \mathbf{x}, \mathbf{y} \rangle|^2 \leq \|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2$$

La fonction  $x \mapsto \sqrt{x}$  étant croissante sur  $[0; +\infty[$ , on obtient (B.1).

**R. 3**

Soit  $\mathbf{x} \neq 0$ . On veut montrer que

$$|\langle \mathbf{x}, \mathbf{y} \rangle| = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \iff \mathbf{y} = \alpha \mathbf{x}$$

$\Leftarrow$  On suppose  $\mathbf{y} = \alpha \mathbf{x}$ . On a alors

$$\langle \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{x} \rangle = \alpha \|\mathbf{x}\|_2^2 \implies |\langle \mathbf{x}, \mathbf{y} \rangle| = |\alpha| \|\mathbf{x}\|_2^2.$$

Comme  $\|\mathbf{y}\|_2 = |\alpha| \|\mathbf{x}\|_2$ , on a aussi

$$\|\mathbf{x}\|_2 \|\mathbf{y}\|_2 = |\alpha| \|\mathbf{x}\|_2^2.$$

On en déduit alors

$$|\langle \mathbf{x}, \mathbf{y} \rangle| = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2.$$

$\Rightarrow$  On suppose  $|\langle \mathbf{x}, \mathbf{y} \rangle| = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$ . Avec cette hypothèse, l'équation (R25.23) devient

$$\|\alpha \mathbf{x} - \mathbf{y}\|_2^2 = 0$$

et donc  $\alpha \mathbf{x} - \mathbf{y} = 0$ , c'est à dire  $\mathbf{y} = \alpha \mathbf{x}$ .

~~~~~ Fin correction ~~~~~

**EXERCICE 2.3.26**

Soient  $\mathbf{x}$  et  $\mathbf{y}$  deux vecteurs de  $\mathbb{C}^n$ .

**Q. 1** Démontrer l'inégalité triangulaire

$$\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2. \quad (\text{B.1})$$

**Q. 2** Si  $\mathbf{x}$  et  $\mathbf{y}$  sont non nuls, prouver que l'inégalité (B.1) est une égalité si et seulement si  $\mathbf{y} = \alpha \mathbf{x}$  avec  $\alpha$  un réel strictement positif.

**Q. 3** Dédurre de (B.1) l'inégalité suivante :

$$|\|\mathbf{x}\|_2 - \|\mathbf{y}\|_2| \leq \|\mathbf{x} - \mathbf{y}\|_2. \quad (\text{B.2})$$

**Q. 4** Soient  $\mathbf{x}_1, \dots, \mathbf{x}_p$ ,  $p$  vecteurs de  $\mathbb{C}^n$ . Montrer que

$$\left\| \sum_{i=1}^p \mathbf{x}_i \right\|_2 \leq \sum_{i=1}^p \|\mathbf{x}_i\|_2. \quad (\text{B.3})$$

**Correction****R. 1**

On rappelle que  $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle$ . On obtient, en utilisant les propriétés du produit scalaire,

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\|_2^2 &= \langle \mathbf{x} + \mathbf{y}, \mathbf{x} + \mathbf{y} \rangle \\ &= \langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{x} \rangle + \langle \mathbf{x}, \mathbf{x} \rangle \\ &= \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 + \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{x} \rangle. \end{aligned}$$

Pour tout nombre complexe  $z$ , on a  $z + \bar{z} = 2 \operatorname{Re}(z)$ , et  $|z| \geq \operatorname{Re}(z)$ .<sup>a</sup> Comme  $\langle \mathbf{y}, \mathbf{x} \rangle = \overline{\langle \mathbf{x}, \mathbf{y} \rangle}$ , on a

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\|_2^2 &= \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 + 2 \operatorname{Re}(\langle \mathbf{x}, \mathbf{y} \rangle) \\ &\leq \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 + 2|\langle \mathbf{x}, \mathbf{y} \rangle|. \end{aligned} \quad (\text{R26.24})$$

L'inégalité de Cauchy-Schwarz donne

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$$

et donc

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\|_2^2 &\leq \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 + 2 \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \\ &= (\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2)^2. \end{aligned}$$

La fonction  $x \mapsto \sqrt{x}$  étant croissante sur  $[0; +\infty[$ , on obtient

$$\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2.$$

<sup>a</sup>En effet,  $z = a + ib$  et  $\bar{z} = a - ib$  d'où  $z + \bar{z} = 2a$ . De plus,  $|z|^2 = a^2 + b^2 \geq a^2$  ce qui donne  $|z| \geq |a| \geq a$ .

**R. 2**

Soient  $\mathbf{x}$  et  $\mathbf{y}$  deux vecteurs de  $\mathbb{C}^n$  non nuls. On veut démontrer que

$$\|\mathbf{x} + \mathbf{y}\|_2 = \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2 \iff \mathbf{y} = \alpha \mathbf{x}, \alpha > 0.$$

$\Leftarrow$  On suppose  $\mathbf{y} = \alpha \mathbf{x}$  avec  $\alpha > 0$ .

On a alors

$$\begin{aligned}\|\mathbf{x} + \mathbf{y}\|_2 &= \|\mathbf{x} + \alpha\mathbf{x}\|_2 \\ &= \|(1 + \alpha)\mathbf{x}\|_2 \\ &= |1 + \alpha| \|\mathbf{x}\|_2.\end{aligned}$$

Comme  $\alpha > 0$ , on a  $|1 + \alpha| = 1 + \alpha$  et donc

$$\|\mathbf{x} + \mathbf{y}\|_2 = \|\mathbf{x}\|_2 + \alpha \|\mathbf{x}\|_2. \quad (\text{R26.25})$$

De plus, on a

$$\begin{aligned}\|\mathbf{y}\|_2 &= \|\alpha\mathbf{x}\|_2 \\ &= |\alpha| \|\mathbf{x}\|_2 \\ &= \alpha \|\mathbf{x}\|_2, \text{ car } \alpha > 0.\end{aligned}$$

D'après (R26.25), on en déduit

$$\|\mathbf{x} + \mathbf{y}\|_2 = \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2.$$

⇒ On suppose que

$$\|\mathbf{x} + \mathbf{y}\|_2 = \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2. \quad (\text{R26.26})$$

Ce qui donne

$$\|\mathbf{x} + \mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 + 2\|\mathbf{x}\|_2\|\mathbf{y}\|_2 \quad (\text{R26.27})$$

On déduit alors de l'égalité (R26.24) que

$$\operatorname{Re}(\langle \mathbf{x}, \mathbf{y} \rangle) = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2. \quad (\text{R26.28})$$

Or, on a,  $\forall z \in \mathbb{C}$ ,  $\operatorname{Re}(z) \leq |z|$  et donc, en utilisant l'inégalité de Cauchy-Schwarz

$$\|\mathbf{x}\|_2 \|\mathbf{y}\|_2 = \operatorname{Re}(\langle \mathbf{x}, \mathbf{y} \rangle) \leq |\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$$

ce qui impose

$$|\langle \mathbf{x}, \mathbf{y} \rangle| = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2.$$

D'après l'exercice précédent, cette égalité est vérifiée si et seulement si  $\mathbf{y} = \alpha\mathbf{x}$  avec  $\alpha = \frac{\langle \mathbf{y}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}$ .

Il nous reste à vérifier que  $\alpha > 0$ .

L'hypothèse (R26.26) avec  $\mathbf{y} = \alpha\mathbf{x}$  devient

$$\|\mathbf{x} + \mathbf{y}\|_2 = |1 + \alpha| \|\mathbf{x}\|_2 = \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2 = 1 + |\alpha| \|\mathbf{x}\|_2. \quad (\text{R26.29})$$

On a donc

$$\begin{aligned}|1 + \alpha| = 1 + |\alpha| &\iff |1 + \alpha|^2 = (1 + |\alpha|)^2 \\ &\iff (1 + \alpha)(1 + \bar{\alpha}) = 1 + 2|\alpha| + |\alpha|^2 \\ &\iff (1 + \alpha)(1 + \bar{\alpha}) = 1 + 2|\alpha| + |\alpha|^2 \\ &\iff 1 + \alpha\bar{\alpha} + \alpha + \bar{\alpha} = 1 + 2|\alpha| + |\alpha|^2 \\ &\iff \operatorname{Re}(\alpha) = |\alpha|\end{aligned}$$

Donc  $\alpha$  est un réel et  $\alpha = \operatorname{Re}(\alpha) = |\alpha| \geq 0$ .

Comme  $\mathbf{y} = \alpha\mathbf{x}$  et  $\mathbf{y} \neq 0$ , on a  $\alpha \neq 0$  et donc  $\alpha > 0$ .

R. 3

On a  $\mathbf{x} = (\mathbf{x} - \mathbf{y}) + \mathbf{y}$ , et par application de l'inégalité triangulaire (B.1) on obtient

$$\|\mathbf{x}\|_2 = \|(\mathbf{x} - \mathbf{y}) + \mathbf{y}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2 + \|\mathbf{y}\|_2 \implies \|\mathbf{x}\|_2 - \|\mathbf{y}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2.$$

De même, avec  $\mathbf{y} = (\mathbf{y} - \mathbf{x}) + \mathbf{x}$ , et par application de l'inégalité triangulaire (B.1) on a

$$\|\mathbf{y}\|_2 = \|(\mathbf{y} - \mathbf{x}) + \mathbf{x}\|_2 \leq \|\mathbf{y} - \mathbf{x}\|_2 + \|\mathbf{x}\|_2 \implies \|\mathbf{y}\|_2 - \|\mathbf{x}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2.$$

En combinant ces deux inégalités, on obtient alors

$$|\|\mathbf{x}\|_2 - \|\mathbf{y}\|_2| \leq \|\mathbf{x} - \mathbf{y}\|_2.$$

R. 4

On effectue une démonstration par récurrence.

Soit  $n \in \mathbb{N}$ ,  $n > 1$ . On définit la propriété  $\mathcal{P}(n)$  par

$$\mathcal{P}(n) : \left\| \sum_{i=1}^n \mathbf{x}_i \right\|_2 \leq \sum_{i=1}^n \|\mathbf{x}_i\|_2. \quad (\text{R26.30})$$

- On a démontré, en Q.1, que la propriété  $\mathcal{P}(2)$  est vraie.
- Soit  $n > 2$ , on suppose que  $\mathcal{P}(n)$  est vérifiée (hypothèse de récurrence). On veut alors montrer que  $\mathcal{P}(n+1)$  est vraie.  
On a

$$\begin{aligned} \left\| \sum_{i=1}^{n+1} \mathbf{x}_i \right\|_2 &= \left\| \sum_{i=1}^n \mathbf{x}_i + \mathbf{x}_{n+1} \right\|_2 \\ &\leq \left\| \sum_{i=1}^n \mathbf{x}_i \right\|_2 + \|\mathbf{x}_{n+1}\|_2, \text{ d'après (B.1)} \\ &\leq \sum_{i=1}^n \|\mathbf{x}_i\|_2 + \|\mathbf{x}_{n+1}\|_2, \text{ car } \mathcal{P}(n) \text{ est vérifiée} \\ &= \sum_{i=1}^{n+1} \|\mathbf{x}_i\|_2. \end{aligned}$$

La propriété  $\mathcal{P}(n+1)$  est donc vérifiée.

On a donc démontré par récurrence que la propriété  $\mathcal{P}(n)$  est vraie pour tout  $n \geq 2$ .

Fin correction

### EXERCICE 2.3.27

Q. 1

Soit la fonction  $f(t) = (1 - \lambda) + \lambda t - t^\lambda$  avec  $0 < \lambda < 1$ . Montrer que pour tous  $\alpha \geq 0$  et  $\beta \geq 0$  on a

$$\alpha^\lambda \beta^{1-\lambda} \leq \lambda \alpha + (1 - \lambda) \beta. \quad (\text{B.1})$$

Soient  $\mathbf{x}$  et  $\mathbf{y}$  deux vecteurs non nuls de  $\mathbb{C}^n$ . Soient  $p > 1$  et  $q > 1$  vérifiant  $\frac{1}{p} + \frac{1}{q} = 1$ .

Q. 2

On pose  $\mathbf{u} = \frac{\mathbf{x}}{\|\mathbf{x}\|_p}$  et  $\mathbf{v} = \frac{\mathbf{y}}{\|\mathbf{y}\|_q}$ . En utilisant l'inégalité (B.1), montrer que l'on a l'inégalité

$$\sum_{i=1}^n |u_i v_i| \leq \frac{1}{p} \sum_{i=1}^n |u_i|^p + \frac{1}{q} \sum_{i=1}^n |v_i|^q = 1. \quad (\text{B.2})$$

**Q. 3** En déduire l'inégalité de Holder suivante

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \sum_{i=1}^n |x_i y_i| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q. \quad (\text{B.3})$$

Quel est le lien entre l'inégalité de Hölder et l'inégalité de Cauchy-Schwarz?

### Correction

**R. 1**

L'inégalité (B.1) est vérifiée si  $\alpha = 0$  ou  $\beta = 0$ . Il nous reste donc à la vérifier pour  $\alpha > 0$  et  $\beta > 0$ . Dans ce cas (B.1) s'écrit

$$\left(\frac{\alpha}{\beta}\right)^\lambda \leq \lambda \frac{\alpha}{\beta} + (1 - \lambda)$$

c'est à dire

$$f\left(\frac{\alpha}{\beta}\right) \geq 0.$$

Montrons que  $f(t) \geq 0, \forall t \in ]0, +\infty[$ .

On a  $f'(t) = \lambda(1 - t^{\lambda-1})$  et

$$f'(t) = 0 \Leftrightarrow 1 - t^{\lambda-1} = 0, \text{ car } \lambda \neq 0$$

De plus, on a  $t^{\lambda-1} = e^{(\lambda-1)\ln(t)}$  et comme  $\lambda - 1 \neq 0$ , on obtient

$$f'(t) = 0 \Leftrightarrow t = 1.$$

- Etudions la fonction sur  $]0, 1[$ . On a pour  $t \in ]0, 1[$ ,  $\ln(t) < 0$  et donc  $(\lambda - 1)\ln(t) > 0$ . Comme la fonction exp est croissante, on en déduit  $\exp((\lambda - 1)\ln(t)) > 1$  et alors  $f'(t) < 0$ .
- Etudions la fonction sur  $]1, +\infty[$ . On a pour  $t \in ]1, +\infty[$ ,  $\ln(t) > 0$  et donc  $(\lambda - 1)\ln(t) < 0$ . Comme la fonction exp est croissante, on en déduit  $0 < \exp((\lambda - 1)\ln(t)) < 1$  et alors  $f'(t) > 0$ .

Le minimum de  $f$  est donc atteint en  $t = 1$  et on a

$$\forall t \in ]0, +\infty[, f(t) \geq f(1) = 0.$$

L'inégalité (B.1) est donc vérifiée  $\forall \alpha \geq 0, \forall \beta \geq 0$  et  $\forall \lambda \in ]0, 1[$ .

**R. 2**

On pose  $\lambda = \frac{1}{p} \in ]0, 1[$ . on a alors  $1 - \lambda = \frac{1}{q}$ . On pose

$$\alpha = |u_i|^p \geq 0, \quad \beta = |v_i|^q \geq 0.$$

En utilisant (B.1), on obtient directement

$$|u_i| |v_i| \leq \frac{1}{p} |u_i|^p + \frac{1}{q} |v_i|^q, \quad \forall i \in \llbracket 1, n \rrbracket.$$

En sommant sur  $i$  on obtient:

$$\sum_{i=1}^n |u_i v_i| \leq \frac{1}{p} \sum_{i=1}^n |u_i|^p + \frac{1}{q} \sum_{i=1}^n |v_i|^q = \frac{1}{p} \|\mathbf{u}\|_p^p + \frac{1}{q} \|\mathbf{v}\|_q^q$$

Comme par construction  $\|\mathbf{u}\|_p = \|\mathbf{v}\|_q = 1$ , on obtient

$$\sum_{i=1}^n |u_i v_i| \leq \frac{1}{p} + \frac{1}{q} = 1.$$

R. 3

Par construction, on a

$$\sum_{i=1}^n |u_i v_i| = \frac{1}{\|\mathbf{x}\|_p \|\mathbf{y}\|_q} \sum_{i=1}^n |x_i y_i|$$

et donc en utilisant l'inégalité (B.3) on obtient

$$\sum_{i=1}^n |x_i y_i| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q.$$

De plus

$$|\langle \mathbf{x}, \mathbf{y} \rangle| = \left| \sum_{i=1}^n \bar{x}_i y_i \right| \leq \sum_{i=1}^n |\bar{x}_i y_i| = \sum_{i=1}^n |x_i y_i| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q.$$

Pour  $p = q = 2$ , l'inégalité de Hölder entraîne l'inégalité de Cauchy-Schwarz.

~~~~~ Fin correction ~~~~~

### EXERCICE 2.3.28

Soit  $p > 1$  et  $q$  le nombre tel que  $\frac{1}{q} = 1 - \frac{1}{p}$ .

Q. 1

Vérifier que  $\forall (\alpha, \beta) \in \mathbb{C}^2$  on a

$$|\alpha + \beta|^p \leq |\alpha| |\alpha + \beta|^{p/q} + |\beta| |\alpha + \beta|^{p/q}. \quad (\text{B.1})$$

Q. 2

En utilisant l'inégalité de Hölder et (B.1), démontrer l'inégalité de Minkowski :

$$\|\mathbf{x} + \mathbf{y}\|_p \leq \|\mathbf{x}\|_p + \|\mathbf{y}\|_p, \quad \forall \mathbf{x} \in \mathbb{C}^n, \quad \forall \mathbf{y} \in \mathbb{C}^n, \quad p \geq 1. \quad (\text{B.2})$$

## Normes matricielles

### EXERCICE 2.3.29 : Norme de Frobenius

Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . On définit l'application  $\|\bullet\|_F$  par

$$\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2. \quad (\text{B.1})$$

Q. 1

On note, respectivement,  $\mathbf{A}_{i,:}$ ,  $\forall i \in \llbracket 1, n \rrbracket$  et  $\mathbf{A}_{:,j}$ ,  $\forall j \in \llbracket 1, n \rrbracket$  les vecteurs lignes et colonnes de  $A$ . Montrer que

$$\|A\|_F^2 = \sum_{i=1}^n \|\mathbf{A}_{i,:}\|_2^2 = \sum_{j=1}^n \|\mathbf{A}_{:,j}\|_2^2 = \text{tr } A^* A. \quad (\text{B.2})$$

Q. 2

Montrer que

$$\|A\mathbf{x}\|_2 \leq \|A\|_F \|\mathbf{x}\|_2, \quad \forall \mathbf{x} \in \mathbb{C}^n. \quad (\text{B.3})$$

Q. 3

Montrer que cette application est une norme matricielle (nommée norme de Frobenius).

Q. 4

Calculer  $\|A^*\|_F$  et  $\|\mathbb{1}_n\|_F$  où  $\mathbb{1}_n$  est la matrice identité de  $\mathcal{M}_n(\mathbb{C})$ .

### EXERCICE 2.3.30

Soit  $A \in \mathcal{M}_{m,n}(\mathbb{C})$ . Montrer les propriétés suivantes

- a.  $\|A\|_2 = \max_{\substack{\|x\|_2=1 \\ \|y\|_2=1}} |\langle Ax, y \rangle|.$
- b.  $\|A\|_2 = \|A^*\|_2.$
- c.  $\|A^*A\|_2 = \|A\|_2^2.$
- d.  $\|U^*AV\|_2 = \|A\|_2$  quand  $UU^* = I$  et  $VV^* = I$

**EXERCICE 2.3.31**

Etant donné une norme vectorielle  $\|\bullet\|$  sur  $\mathbb{K}^n$ , on définit l'application  $\|\bullet\|_s : \mathcal{M}_n(\mathbb{K}) \rightarrow \mathbb{R}^+$  par

$$\|A\|_s \stackrel{\text{def}}{=} \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|Av\|}{\|v\|} \quad (\text{B.1})$$

**Q. 1**

Montrer que  $\|I\|_s = 1.$

On note  $\mathcal{B} = \{v \in \mathbb{K}^n ; \|v\| \leq 1\}$  la boule unité de  $\mathbb{K}^n$  et  $\mathcal{S} = \{v \in \mathbb{K}^n ; \|v\| = 1\}$  la sphère unité de  $\mathbb{K}^n$ .

**Q. 2**

- a. Montrer que  $\mathcal{B}$  et  $\mathcal{S}$  sont des compacts.
- b. Montrer que

$$\|A\|_s = \sup_{v \in \mathcal{S}} \|Av\| \quad (\text{B.2})$$

- c. En déduire que

$$\|A\|_s = \sup_{v \in \mathcal{B}} \|Av\| \quad (\text{B.3})$$

- d. En déduire que l'application  $\|\bullet\|_s$  est bien définie sur  $\mathcal{M}_n(\mathbb{K})$  i.e.  $\forall A \in \mathcal{M}_n(\mathbb{K}), \|A\|_s < +\infty.$

**Q. 3**

- a. Montrer

$$\|A\|_s \leq \inf \{ \alpha \in \mathbb{R} : \|Av\| \leq \alpha \|v\|, \forall v \in \mathbb{K}^n \}.$$

- b. Montrer qu'il existe  $w \in \mathcal{S}$  tel que  $\|A\|_s = \|Aw\|.$

- c. En déduire que

$$\|A\|_s = \inf \{ \alpha \in \mathbb{R} : \|Av\| \leq \alpha \|v\|, \forall v \in \mathbb{K}^n \}. \quad (\text{B.4})$$

**Q. 4**

- a. Montrer que  $\forall v \in \mathbb{K}^n, \|Av\| \leq \|A\|_s \|v\|.$

- b. Soit  $\lambda \in \mathbb{K}^*$ . Montrer qu'il existe  $u \in \mathbb{K}^n$  tel que  $\|u\| = |\lambda|$  vérifiant

$$\|Au\| = \|A\|_s \|u\|.$$

**Q. 5**

Montrer que  $\|\bullet\|_s$  est une norme matricielle.

**Correction****R. 1**

On a immédiatement

$$\|I\|_s = \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|v\|}{\|v\|} = \sup_{\substack{v \in \mathbb{K}^n \\ v \neq 0}} \frac{\|v\|}{\|v\|} = 1.$$

R. 2

a. Les ensembles  $\mathcal{B}$  et  $\mathcal{S}$  sont des compacts car image réciproque de l'application continue  $\mathbf{v} \mapsto \|\mathbf{v}\|$  par le fermé borné  $[0, 1]$  (pour la boule) et le singleton  $\{1\}$  (pour la sphère).

b. On a

$$\|\mathbb{A}\|_s = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \left\| \mathbb{A} \frac{\mathbf{v}}{\|\mathbf{v}\|} \right\| = \sup_{\mathbf{u} \in \mathcal{S}} \|\mathbb{A}\mathbf{u}\|$$

c. Comme  $\mathcal{S} \subset \mathcal{B}$  on a aussi

$$\sup_{\mathbf{v} \in \mathcal{B}} \|\mathbb{A}\mathbf{v}\| \geq \sup_{\mathbf{v} \in \mathcal{S}} \|\mathbb{A}\mathbf{v}\|. \quad (\text{R31.31})$$

On peut aussi remarquer que

$$\sup_{\mathbf{v} \in \mathcal{B}} \|\mathbb{A}\mathbf{v}\| = \sup_{\substack{\mathbf{v} \in \mathcal{B} \\ \mathbf{v} \neq 0}} \|\mathbb{A}\mathbf{v}\| \quad (\text{R31.32})$$

De plus,  $\forall \mathbf{w} \in \mathcal{B} \setminus \{0\}$ , en posant  $\mathbf{u} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \in \mathcal{S}$ , on a  $\mathbf{w} = \|\mathbf{w}\| \mathbf{u}$  et

$$\|\mathbb{A}\mathbf{w}\| = \|\mathbf{w}\| \|\mathbb{A}\mathbf{u}\| \leq \|\mathbb{A}\mathbf{u}\| \text{ car } \|\mathbf{w}\| \leq 1.$$

Or on a

$$\|\mathbb{A}\mathbf{u}\| \leq \sup_{\mathbf{v} \in \mathcal{S}} \|\mathbb{A}\mathbf{v}\|.$$

et on obtient alors

$$\sup_{\substack{\mathbf{w} \in \mathcal{B} \\ \mathbf{w} \neq 0}} \|\mathbb{A}\mathbf{w}\| \leq \sup_{\mathbf{v} \in \mathcal{S}} \|\mathbb{A}\mathbf{v}\|.$$

En utilisant (R31.31) et (R31.32), on en déduit

$$\sup_{\mathbf{w} \in \mathcal{B}} \|\mathbb{A}\mathbf{w}\| = \sup_{\mathbf{u} \in \mathcal{S}} \|\mathbb{A}\mathbf{u}\|.$$

d. L'application  $\mathbf{v} \mapsto \|\mathbb{A}\mathbf{v}\|$  est continue donc son sup sur la sphère unité qui est compacte est atteint.

R. 3

a. Comme  $\|\bullet\|_s$  est bien définie il existe  $\alpha \in \mathbb{R}^+$  tel que  $\|\mathbb{A}\|_s \leq \alpha$ . Soit  $\alpha \in \mathbb{R}^+$  tel que  $\|\mathbb{A}\|_s \leq \alpha$ . On a

$$\begin{aligned} \|\mathbb{A}\|_s \leq \alpha &\Leftrightarrow \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq 0}} \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} \leq \alpha \\ &\Leftrightarrow \frac{\|\mathbb{A}\mathbf{v}\|}{\|\mathbf{v}\|} \leq \alpha, \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \\ &\Leftrightarrow \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \\ &\Leftrightarrow \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n \end{aligned}$$

On en déduit que

$$\|\mathbb{A}\|_s \leq \inf\{\alpha \in \mathbb{R} : \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n\} \quad (\text{R31.33})$$

b. Comme  $\mathcal{S}$  est compact et l'application  $\mathbf{v} \mapsto \|\mathbb{A}\mathbf{v}\|$  est continue, il existe  $\mathbf{w} \in \mathcal{S}$  tel que

$$\|\mathbb{A}\|_s = \sup_{\mathbf{v} \in \mathcal{S}} \|\mathbb{A}\mathbf{v}\| = \|\mathbb{A}\mathbf{w}\|.$$

c. On en déduit  $\|\mathbb{A}\|_s \|\mathbf{w}\| = \|\mathbb{A}\mathbf{w}\|$  car  $\|\mathbf{w}\| = 1$ . On a alors

$$\|\mathbb{A}\|_s \in \{\alpha \in \mathbb{R} : \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n\}$$

et donc

$$\inf\{\alpha \in \mathbb{R} : \|\mathbb{A}\mathbf{v}\| \leq \alpha \|\mathbf{v}\|, \forall \mathbf{v} \in \mathbb{K}^n\} \leq \|\mathbb{A}\|_s.$$

On conclut en utilisant (R31.33).

R. 4

a. On a par définition du sup

$$\|A\|_s = \sup_{\substack{\mathbf{u} \in \mathbb{K}^n \\ \mathbf{u} \neq \mathbf{0}}} \frac{\|A\mathbf{u}\|}{\|\mathbf{u}\|} \geq \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|}, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\}.$$

et donc

$$\|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\|, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\}.$$

qui est équivalent à

$$\|A\mathbf{v}\| \leq \|A\|_s \|\mathbf{v}\|, \quad \forall \mathbf{v} \in \mathbb{K}^n.$$

b. D'après la Q. 3 b., il existe  $\mathbf{w} \in \mathcal{S}$  tel que  $\|A\|_s = \|A\mathbf{w}\|$ . Soit  $\lambda \in \mathbb{K}^*$  et  $\mathbf{u} = \lambda\mathbf{w} \neq \mathbf{0}$ . On a  $\|\mathbf{u}\| = |\lambda|$  et

$$\|A\|_s = \|A\mathbf{w}\| = \left\| A \frac{\mathbf{u}}{\|\mathbf{u}\|} \right\| = \frac{1}{\|\mathbf{u}\|} \|A\mathbf{u}\| \Leftrightarrow \|A\|_s \|\mathbf{u}\| = \|A\mathbf{u}\|$$

R. 5

- $\|A\|_s = 0 \Leftrightarrow A_s = \mathbf{0}$  ?

$\Leftarrow$  trivial.

$\Rightarrow$  Soit  $A \in \mathcal{M}_n(\mathbb{K})$ .

$$\begin{aligned} \|A\|_s = 0 &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} \implies \|A\mathbf{v}\| = 0, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \\ &\implies A\mathbf{v} = \mathbf{0}, \quad \forall \mathbf{v} \in \mathbb{K}^n \setminus \{0\} \end{aligned}$$

Soit  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$  la base canonique de  $\mathbb{K}^n$ . On a alors  $\forall j \in \llbracket 1, n \rrbracket, A\mathbf{e}_j = \mathbf{0}$  et on en déduit que

$$A_{i,j} = \langle \mathbf{e}_i, A\mathbf{e}_j \rangle = 0, \quad \forall (i, j) \in \llbracket 1, n \rrbracket.$$

et donc  $A = \mathbf{0}$ .

- Montrons que  $\|\alpha A\|_s = |\alpha| \|A\|_s, \forall \alpha \in \mathbb{K}, \forall A \in \mathcal{M}_n(\mathbb{K})$ .

Soient  $\alpha \in \mathbb{K}$  et  $A \in \mathcal{M}_n(\mathbb{K})$ . On a  $\alpha A \in \mathcal{M}_n(\mathbb{K})$  (car  $\mathcal{M}_n(\mathbb{K})$  est un espace vectoriel) et

$$\begin{aligned} \|\alpha A\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\alpha A\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{|\alpha| \|A\mathbf{v}\|}{\|\mathbf{v}\|} \quad \text{car } \|\alpha \mathbf{u}\| = |\alpha| \|\mathbf{u}\| \\ &= |\alpha| \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} = |\alpha| \|A\|_s. \end{aligned}$$

- Montrons que  $\|A + B\|_s \leq \|A\|_s + \|B\|_s, \forall (A, B) \in \mathcal{M}_n(\mathbb{K})^2$

Soient  $A$  et  $B$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$ . On a  $A + B \in \mathcal{M}_n(\mathbb{K})$  car  $\mathcal{M}_n(\mathbb{K})$  est un espace vectoriel et

$$\begin{aligned} \|A + B\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|(A + B)\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\mathbf{v} + B\mathbf{v}\|}{\|\mathbf{v}\|} \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\mathbf{v}\| + \|B\mathbf{v}\|}{\|\mathbf{v}\|} \quad \text{par inégalité triangulaire dans } \mathbb{K}^n \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} + \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|B\mathbf{v}\|}{\|\mathbf{v}\|} = \|A\|_s + \|B\|_s. \end{aligned}$$

- Montrons que  $\|AB\|_s \leq \|A\|_s \|B\|_s, \forall (A, B) \in \mathcal{M}_n(\mathbb{K})^2$ .

Soient  $A$  et  $B$  deux matrices de  $\mathcal{M}_n(\mathbb{K})$ . On a  $AB \in \mathcal{M}_n(\mathbb{K})$  par définition du produit matriciel

et

$$\begin{aligned} \|A\mathbb{B}\|_s &= \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|(\mathbb{A}\mathbb{B})\mathbf{v}\|}{\|\mathbf{v}\|} = \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A(\mathbb{B}\mathbf{v})\|}{\|\mathbf{v}\|} \\ &\leq \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|A\|_s \|\mathbb{B}\mathbf{v}\|}{\|\mathbf{v}\|} \quad \text{car } \|A\mathbf{u}\| \leq \|A\|_s \|\mathbf{u}\| \quad \forall \mathbf{u} \in \mathbb{K}^n \\ &\leq \|A\|_s \sup_{\substack{\mathbf{v} \in \mathbb{K}^n \\ \mathbf{v} \neq \mathbf{0}}} \frac{\|\mathbb{B}\mathbf{v}\|}{\|\mathbf{v}\|} = \|A\|_s \|\mathbb{B}\|_s. \end{aligned}$$

Fin correction

### EXERCICE 2.3.32

Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . On note

$$\|A\|_1 \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \mathbf{x} \neq \mathbf{0}}} \frac{\|A\mathbf{x}\|_1}{\|\mathbf{x}\|_1} = \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_1=1}} \|A\mathbf{x}\|_1$$

la norme matricielle subordonnée à la norme vectorielle  $\|\bullet\|_1$ .

**Q. 1** Montrer que

$$\sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_1=1}} \|A\mathbf{x}\|_1 \leq \max_{j \in [1, n]} \sum_{i=1}^n |A_{i,j}|.$$

**Q. 2** a. Déterminer un  $\mathbf{y} \in \mathbb{C}^n$ ,  $\|\mathbf{y}\|_1 = 1$  tel que

$$\|A\mathbf{y}\|_1 = \max_{j \in [1, n]} \sum_{i=1}^n |A_{i,j}|.$$

b. Conclure.

**Q. 3** [Algo] Ecrire une fonction algorithmique, nommée `Norm1`, calculant  $\|A\|_1$ .

### Correction

**R. 1** Soit  $\mathbf{x} \in \mathbb{C}^n$  tel que  $\|\mathbf{x}\|_1 = 1$ . On a

$$\begin{aligned} \|A\mathbf{x}\|_1 &= \sum_{i=1}^n |(A\mathbf{x})_i| = \sum_{i=1}^n \left| \sum_{j=1}^n A_{i,j} x_j \right| \\ &\leq \sum_{i=1}^n \sum_{j=1}^n |A_{i,j} x_j| = \sum_{j=1}^n \left( |x_j| \sum_{i=1}^n |A_{i,j}| \right) \\ &\leq \left( \max_{j \in [1, n]} \sum_{i=1}^n |A_{i,j}| \right) \sum_{j=1}^n |x_j| = \max_{j \in [1, n]} \sum_{i=1}^n |A_{i,j}| \quad \text{car } \|\mathbf{x}\|_1 = \sum_{j=1}^n |x_j| = 1. \end{aligned}$$

On obtient donc

$$\sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_1=1}} \|A\mathbf{x}\|_1 \leq \max_{j \in [1, n]} \sum_{i=1}^n |A_{i,j}|.$$

**R. 2**a. Soit  $k \in \llbracket 1, n \rrbracket$  tel que

$$\sum_{i=1}^n |A_{i,k}| = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |A_{i,j}|.$$

On a

$$\|A\mathbf{y}\|_1 = \sum_{i=1}^n |(A\mathbf{y})_i| = \sum_{i=1}^n \left| \sum_{j=1}^n A_{i,j} y_j \right|.$$

Pour obtenir

$$\sum_{i=1}^n |A_{i,k}| = \sum_{i=1}^n \left| \sum_{j=1}^n A_{i,j} y_j \right|$$

on prend  $y_j = \delta_{k,j}$ ,  $\forall j \in \llbracket 1, n \rrbracket$ , c'est à dire  $\mathbf{y} = \mathbf{e}_k$  le  $k^{\text{ème}}$  vecteur de la base canonique. Dans ce cas on a  $\|\mathbf{y}\|_1 = 1$  et

$$\begin{aligned} \|A\mathbf{y}\|_1 &= \|A\mathbf{e}_k\|_1 = \|A_{:,k}\|_1 \\ &= \sum_{i=1}^n |A_{i,k}| = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |A_{i,j}|. \end{aligned}$$

b. D'après la proposition/définition des normes matricielles subordonnées, on a

$$\|A\|_1 = \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_1 = 1}} \|A\mathbf{x}\|_1.$$

En utilisant les résultats de **Q.1** et **Q.2**, on obtient

$$\|A\|_1 = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |A_{i,j}|.$$

**R. 3**

Voici une possibilité de fonction:

**Algorithme B.28** Fonction `Norm1` permettant de calculer  $\|A\|_1$ **Données :**  $A$  : matrice de  $\mathcal{M}_n(\mathbb{C})$ .**Résultat :**  $r$  : le réel  $r = \max_{j \in \llbracket 1, n \rrbracket} \sum_{i=1}^n |a_{i,j}|$ 

```

1: Fonction  $r \leftarrow \text{Norm1}(A)$ 
2:    $n \leftarrow$  nb de lignes de  $A$ 
3:    $r \leftarrow 0$ 
4:   Pour  $j \leftarrow 1$  à  $n$  faire
5:      $S \leftarrow 0$ 
6:     Pour  $i \leftarrow 1$  à  $n$  faire
7:        $S \leftarrow S + |A(i,j)|$ 
8:     Fin
9:     Si  $r < S$  alors
10:       $r \leftarrow S$ 
11:   Fin
12: Fin
13: Fin

```

~~~~~Fin correction~~~~~

**EXERCICE 2.3.33**

Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . On note

$$\|A\|_\infty \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \mathbf{x} \neq 0}} \frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}$$

la norme matricielle subordonnée à la norme vectorielle  $\|\bullet\|_\infty$ .

**Q. 1** Montrer que

$$\sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_\infty = 1}} \|A\mathbf{x}\|_\infty \leq \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}|.$$

**Q. 2** a. Déterminer un  $\mathbf{y} \in \mathbb{C}^n$ ,  $\|\mathbf{y}\|_\infty = 1$  tel que

$$\|A\mathbf{y}\|_\infty = \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}|.$$

b. Conclure.

**Q. 3** [Algo] Ecrire une fonction algorithmique, nommée *NormInf*, calculant  $\|A\|_\infty$ .

**Correction**

**R. 1** Soit  $\mathbf{x} \in \mathbb{C}^n$  tel que  $\|\mathbf{x}\|_\infty = 1$ . On a

$$\begin{aligned} \|A\mathbf{x}\|_\infty &= \max_{i \in [1, n]} |(A\mathbf{x})_i| = \max_{i \in [1, n]} \left| \sum_{j=1}^n A_{i,j} x_j \right| \\ &\leq \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}| |x_j| \\ &\leq \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}| \quad \text{car } |x_j| \leq \max_{i \in [1, n]} |x_i| = \|\mathbf{x}\|_\infty = 1. \end{aligned}$$

On obtient donc

$$\sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_\infty = 1}} \|A\mathbf{x}\|_\infty \leq \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}|.$$

**R. 2** a. Soit  $k \in [1, n]$  tel que

$$\sum_{j=1}^n |A_{k,j}| = \max_{i \in [1, n]} \sum_{j=1}^n |A_{i,j}|.$$

On a, pour tout  $\mathbf{y} \in \mathbb{C}^n$ ,

$$\|A\mathbf{y}\|_\infty = \max_{i \in [1, n]} |(A\mathbf{y})_i| = \max_{i \in [1, n]} \left| \sum_{j=1}^n A_{i,j} y_j \right|.$$

On va construire un vecteur  $\mathbf{y} \in \mathbb{C}^n$ ,  $\|\mathbf{y}\|_\infty = 1$ , tel que

$$\max_{i \in [1, n]} \left| \sum_{j=1}^n A_{i,j} y_j \right| = \sum_{j=1}^n |A_{k,j}|.$$

On sait déjà que, si  $\|\mathbf{y}\|_\infty = 1$ ,

$$\forall i \in \llbracket 1, n \rrbracket, \quad \left| \sum_{j=1}^n A_{i,j} y_j \right| \leq \sum_{j=1}^n |A_{k,j}|.$$

On va donc construire  $\mathbf{y} \in \mathbb{C}^n$ ,  $\|\mathbf{y}\|_\infty = 1$ , de telle sorte que

$$\left| \sum_{j=1}^n A_{k,j} y_j \right| = \sum_{j=1}^n |A_{k,j}|.$$

Il suffit pour cela de prendre,

$$\forall j \in \llbracket 1, n \rrbracket, \quad y_j = \begin{cases} \frac{|A_{k,j}|}{A_{k,j}} & \text{si } A_{k,j} \neq 0 \\ 1 & \text{si } A_{k,j} = 0 \end{cases}.$$

et on a bien  $\|\mathbf{y}\|_\infty = 1$ . On a alors

$$\left| \sum_{j=1}^n A_{k,j} y_j \right| = \sum_{j=1}^n |A_{k,j}| \quad \text{et} \quad \forall i \in \llbracket 1, n \rrbracket, \quad \left| \sum_{j=1}^n A_{i,j} y_j \right| \leq \sum_{j=1}^n |A_{k,j}|$$

et donc

$$\|\mathbf{A}\mathbf{y}\|_\infty = \sum_{j=1}^n |A_{k,j}| = \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |A_{i,j}|.$$

b. D'après la proposition/définition des normes matricielles subordonnées, on a

$$\|\mathbf{A}\|_\infty = \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_\infty = 1}} \|\mathbf{A}\mathbf{x}\|_\infty.$$

En utilisant les résultats de **Q.1** et **Q.2**, on obtient

$$\|\mathbf{A}\|_\infty = \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |A_{i,j}|.$$

**R. 3**

Voici une possibilité de fonction:

---

**Algorithme B.29** Fonction NormInf permettant de calculer  $\|\mathbf{A}\|_\infty$

---

**Données :**  $\mathbf{A}$  : matrice de  $\mathcal{M}_n(\mathbb{C})$ .

**Résultat :**  $r$  : le réel  $r = \max_{i \in \llbracket 1, n \rrbracket} \sum_{j=1}^n |a_{i,j}|$

```

1: Fonction  $r \leftarrow \text{NormInf}(\mathbf{A})$ 
2:    $n \leftarrow$  nb de lignes de  $\mathbf{A}$ 
3:    $r \leftarrow 0$ 
4:   Pour  $i \leftarrow 1$  à  $n$  faire
5:      $S \leftarrow 0$ 
6:     Pour  $j \leftarrow 1$  à  $n$  faire
7:        $S \leftarrow S + |A(i,j)|$ 
8:     Fin
9:     Si  $r < S$  alors
10:       $r \leftarrow S$ 
11:   Fin
12: Fin
13: Fin

```

---

Fin correction

**EXERCICE 2.3.34**Soit  $A \in \mathcal{M}_n(\mathbb{C})$ . On note  $B = A^*A$ .**Q. 1**Soit  $(\lambda, \mathbf{u}) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$  un élément propre de  $B$ .

- Montrer que la matrice  $B$  est hermitienne.
- Montrer que les valeurs propres de  $B$  sont réelles.
- En déduire que

$$\lambda = \frac{\|A\mathbf{u}\|_2^2}{\|\mathbf{u}\|_2^2}.$$

La matrice  $B$  étant hermitienne (elle est donc normale), d'après le Théorème de réduction 3.2 page 63, il existe alors une matrice  $U$  unitaire et une matrice  $D$  diagonale telle que

$$B = UDU^*.$$

On note  $(\lambda_i, \mathbf{e}_i)_{i \in [1, n]}$  les éléments propres de  $D$ . Les vecteurs  $\mathbf{e}_i$  sont les vecteurs de la base canonique de  $\mathbb{C}^n$  et  $\lambda_i = D_{ii}$ .**Q. 2**

- Démontrer que les  $(\lambda_i, \mathbf{v}_i)_{i \in [1, n]}$  sont les éléments propres de  $B$  où  $\mathbf{v}_i$  est le  $i$ -ème vecteur colonne de  $U$ .
- En déduire que  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  est une base orthonormée de  $\mathbb{C}^n$ .

Soit  $\mathbf{x} \in \mathbb{C}^n$  tel que  $\|\mathbf{x}\|_2 = 1$  décomposée dans la base  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ :

$$\exists (\alpha_1, \dots, \alpha_n) \in \mathbb{C}^n, \text{ tels que } \mathbf{x} = \sum_{i=1}^n \alpha_i \mathbf{v}_i.$$

**Q. 3**

- Montrer que

$$\langle \mathbf{x}, \mathbf{x} \rangle = \sum_{i=1}^n |\alpha_i|^2 = 1.$$

- Montrer que

$$\sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \|\mathbf{v}\|_2 = 1}} \|A\mathbf{v}\|_2^2 \leq \rho(A^*A).$$

- Déterminer un vecteur  $\mathbf{w} \in \mathbb{C}^n$ , tel que

$$\|A\mathbf{w}\|_2^2 = \rho(A^*A).$$

- En déduire que

$$\|A\|_2 \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \mathbf{x} \neq 0}} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sqrt{\rho(A^*A)}.$$

**Q. 4**

- Montrer que la norme  $\|\bullet\|_2$  est invariante par transformation unitaire :

$$UU^* = I \implies \|A\|_2 = \|UA\|_2 = \|AU\|_2 = \|U^*AU\|_2.$$

- Montrer que si  $A$  est hermitienne alors

$$\|A\|_2 = \rho(A).$$

## Correction

R. 1

a. Il faut montrer que  $\mathbb{B} = \mathbb{B}^*$ . Or on a

$$\mathbb{B}^* = (\mathbb{A}^* \mathbb{A})^* = \mathbb{A}^* (\mathbb{A}^*)^* = \mathbb{A}^* \mathbb{A} = \mathbb{B}.$$

b. Comme  $(\lambda, \mathbf{u})$  est un élément propre de  $\mathbb{B}$ , on a  $\mathbb{B}\mathbf{u} = \lambda\mathbf{u}$ . On en déduit que

$$\langle \mathbb{B}\mathbf{u}, \mathbf{u} \rangle = \langle \lambda\mathbf{u}, \mathbf{u} \rangle = \bar{\lambda} \langle \mathbf{u}, \mathbf{u} \rangle = \bar{\lambda} \|\mathbf{u}\|_2^2.$$

De plus par propriété du produit scalaire, on a

$$\langle \mathbb{B}\mathbf{u}, \mathbf{u} \rangle = \langle \mathbf{u}, \mathbb{B}^*\mathbf{u} \rangle.$$

Comme  $\mathbb{B}$  est hermitienne, on obtient

$$\begin{aligned} \langle \mathbb{B}\mathbf{u}, \mathbf{u} \rangle &= \langle \mathbf{u}, \mathbb{B}\mathbf{u} \rangle \\ &= \langle \mathbf{u}, \lambda\mathbf{u} \rangle = \lambda \langle \mathbf{u}, \mathbf{u} \rangle = \lambda \|\mathbf{u}\|_2^2. \end{aligned}$$

On a donc

$$\lambda \|\mathbf{u}\|_2^2 = \bar{\lambda} \|\mathbf{u}\|_2^2$$

et comme  $\|\mathbf{u}\|_2 \neq 0$  ( $\mathbf{u}$  est un vecteur propre) on obtient  $\lambda = \bar{\lambda}$ , c'est à dire  $\lambda \in \mathbb{R}$ .

c. On a

$$\begin{aligned} \langle \mathbb{B}\mathbf{u}, \mathbf{u} \rangle &= \langle \mathbb{A}^* \mathbb{A}\mathbf{u}, \mathbf{u} \rangle \\ &= \langle \mathbb{A}\mathbf{u}, \mathbb{A}\mathbf{u} \rangle \quad \text{par propriété du produit scalaire} \\ &= \|\mathbb{A}\mathbf{u}\|_2^2. \end{aligned}$$

De plus, on a vu que  $\langle \mathbb{B}\mathbf{u}, \mathbf{u} \rangle = \lambda \|\mathbf{u}\|_2^2$  avec  $\|\mathbf{u}\|_2 > 0$ . On en déduit alors

$$\lambda = \frac{\|\mathbb{A}\mathbf{u}\|_2^2}{\|\mathbf{u}\|_2^2} \geq 0.$$

R. 2

a. On a  $\mathbb{B} = \mathbb{U}\mathbb{D}\mathbb{U}^*$ . Or  $\mathbb{U}$  est unitaire, donc inversible d'inverse  $\mathbb{U}^*$ . En multipliant à gauche par  $\mathbb{U}^*$  et à droite par  $\mathbb{U}$  on obtient

$$\mathbb{U}^* \mathbb{B} \mathbb{U} = \mathbb{U}^* (\mathbb{U}\mathbb{D}\mathbb{U}^*) \mathbb{U} = (\mathbb{U}^* \mathbb{U}) \mathbb{D} (\mathbb{U}^* \mathbb{U}) = \mathbb{D}.$$

On obtient alors

$$\begin{aligned} \mathbb{D}\mathbf{e}_i = \lambda_i \mathbf{e}_i &\iff \mathbb{U}^* \mathbb{B} \mathbb{U} \mathbf{e}_i = \lambda_i \mathbf{e}_i \\ &\iff \mathbb{B} \mathbb{U} \mathbf{e}_i = \lambda_i \mathbb{U} \mathbf{e}_i. \end{aligned}$$

C'est à dire en posant  $\mathbf{v}_i = \mathbb{U}\mathbf{e}_i$  ( $i$ -ème vecteur colonne de  $\mathbb{U}$ ), les éléments propres de  $\mathbb{B}$  sont les  $(\lambda_i, \mathbf{v}_i)_{i \in [1, n]}$ . On peut noter que  $\mathbf{v}_i \neq 0$  car  $\mathbb{U}$  est inversible.

b. On a

$$\mathbb{U} = \begin{pmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{pmatrix} \quad \text{et} \quad \mathbb{U}^* = \begin{pmatrix} \mathbf{v}_1^* \\ \mathbf{v}_2^* \\ \vdots \\ \mathbf{v}_n^* \end{pmatrix}$$

On a donc

$$U^*U = \begin{pmatrix} \mathbf{v}_1^* \mathbf{v}_1 & \mathbf{v}_1^* \mathbf{v}_2 & \dots & \mathbf{v}_1^* \mathbf{v}_n \\ \mathbf{v}_2^* \mathbf{v}_1 & \mathbf{v}_2^* \mathbf{v}_2 & \dots & \mathbf{v}_2^* \mathbf{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_n^* \mathbf{v}_1 & \mathbf{v}_n^* \mathbf{v}_2 & \dots & \mathbf{v}_n^* \mathbf{v}_n \end{pmatrix}$$

et donc

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, (U^*U)_{i,j} = \mathbf{v}_i^* \mathbf{v}_j = \langle \mathbf{v}_i, \mathbf{v}_j \rangle.$$

Comme  $U$  est unitaire, on a  $U^*U = \mathbb{I}$  et donc

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, (U^*U)_{i,j} = \delta_{i,j}.$$

On en déduit alors

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{i,j}.$$

$\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  est donc une base orthonormée de  $\mathbb{C}^n$ .

**R. 3**

a. On peut voir que

$$\begin{aligned} \langle \mathbf{x}, \mathbf{x} \rangle &= \left\langle \sum_{i=1}^n \alpha_i \mathbf{v}_i, \sum_{j=1}^n \alpha_j \mathbf{v}_j \right\rangle \\ &= \sum_{i=1}^n \overline{\alpha_i} \alpha_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \\ &= \sum_{i=1}^n \overline{\alpha_i} \alpha_i \text{ car } \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij} \\ &= \sum_{i=1}^n |\alpha_i|^2. \end{aligned}$$

De plus  $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle = 1$ .

b. On a

$$\begin{aligned} \|\mathbb{A}\mathbf{x}\|_2^2 &= \langle \mathbb{A}\mathbf{x}, \mathbb{A}\mathbf{x} \rangle = \langle \mathbb{A}^* \mathbb{A} \mathbf{x}, \mathbf{x} \rangle = \langle \mathbb{B}\mathbf{x}, \mathbf{x} \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i \mathbb{B}\mathbf{v}_i, \sum_{j=1}^n \alpha_j \mathbf{v}_j \right\rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i \lambda_i \mathbf{v}_i, \sum_{j=1}^n \alpha_j \mathbf{v}_j \right\rangle \\ &= \sum_{i=1}^n \overline{\alpha_i} \lambda_i \sum_{j=1}^n \alpha_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle \\ &= \sum_{i=1}^n \lambda_i |\alpha_i|^2 \text{ car } \lambda_i \in \mathbb{R} \text{ et } \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij} \\ &\leq \left( \max_{i \in \llbracket 1, n \rrbracket} \lambda_i \right) \sum_{i=1}^n |\alpha_i|^2 = \rho(\mathbb{A}^* \mathbb{A}) \text{ car } \lambda_i \geq 0 \text{ et } \sum_{i=1}^n |\alpha_i|^2 = 1. \end{aligned}$$

On en déduit alors

$$\sup_{\substack{\mathbf{v} \in \mathbb{C}^n \\ \|\mathbf{v}\|_2 = 1}} \|\mathbb{A}\mathbf{v}\|_2^2 \leq \rho(\mathbb{A}^* \mathbb{A}).$$

c. Pour démontrer que l'on a en fait égalité il suffit de trouver un vecteur la vérifiant, c'est à dire un vecteur  $\mathbf{w} \in \mathbb{C}^n$ ,  $\|\mathbf{w}\|_2 = 1$ , tel que

$$\|\mathbb{A}\mathbf{w}\|_2^2 = \max_{i \in \llbracket 1, n \rrbracket} \lambda_i$$

où les  $\lambda_i$  sont positifs ou nuls (valeurs propres de  $\mathbb{B}$ . Pour cela on note  $k \in \llbracket 1, n \rrbracket$  l'indice tel que  $\lambda_k = \max_{i \in \llbracket 1, n \rrbracket} \lambda_i$ . En choisissant  $\mathbf{w} = \mathbf{v}_k$  (qui est de norme 1) on obtient alors

$$\|\mathbb{A}\mathbf{v}_k\|_2^2 = \langle \mathbb{A}\mathbf{v}_k, \mathbb{A}\mathbf{v}_k \rangle = \langle \mathbb{A}^* \mathbb{A}\mathbf{v}_k, \mathbf{v}_k \rangle = \langle \lambda_k \mathbf{v}_k, \mathbf{v}_k \rangle = \lambda_k = \rho(\mathbb{A}^* \mathbb{A}).$$

d. D'après la proposition/définition des normes matricielles subordonnées, on a

$$\|\mathbb{A}\|_2 = \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \|\mathbf{x}\|_2=1}} \|\mathbb{A}\mathbf{x}\|_2.$$

En utilisant les résultats de **Q.3**, 2. et 3., on obtient

$$\|\mathbb{A}\|_2 = \sqrt{\rho(\mathbb{A}^* \mathbb{A})}.$$

**R. 4**

a. Soit  $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$  unitaire, i.e.

$$\mathbb{U}\mathbb{U}^* = \mathbb{U}^*\mathbb{U} = \mathbb{I}.$$

- Montrons que  $\|\mathbb{A}\|_2 = \|\mathbb{U}\mathbb{A}\|_2$ .  
On a  $\|\mathbb{A}\|_2 = \sqrt{\rho(\mathbb{A}^* \mathbb{A})}$  et donc

$$\|\mathbb{U}\mathbb{A}\|_2 = \sqrt{\rho((\mathbb{U}\mathbb{A})^* \mathbb{U}\mathbb{A})} = \sqrt{\rho(\mathbb{A}^* (\mathbb{U}^* \mathbb{U}) \mathbb{A})} = \sqrt{\rho(\mathbb{A}^* \mathbb{A})} = \|\mathbb{A}\|_2.$$

- Montrons que  $\|\mathbb{A}\|_2 = \|\mathbb{A}\mathbb{U}\|_2$ .  
On a

$$\|\mathbb{A}\mathbb{U}\|_2 \stackrel{\text{def}}{=} \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \mathbf{x} \neq 0}} \frac{\|\mathbb{A}\mathbb{U}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

En posant  $\mathbf{y} = \mathbb{U}\mathbf{x}$ , on a  $\mathbf{x} = \mathbb{U}^*\mathbf{y}$  car  $\mathbb{U}^{-1} = \mathbb{U}^*$  ( $\mathbb{U}$  étant unitaire). Comme  $\mathbb{U}$  est inversible on a

$$\begin{aligned} \{\mathbb{U}^*\mathbf{y}, \forall \mathbf{y} \in \mathbb{C}^n \setminus \{0\}\} &= \mathbb{C}^n \setminus \{0\} \\ \sup_{\substack{\mathbf{x} \in \mathbb{C}^n \\ \mathbf{x} \neq 0}} \frac{\|\mathbb{A}\mathbb{U}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} &= \sup_{\substack{\mathbf{y} \in \mathbb{C}^n \\ \mathbf{y} \neq 0}} \frac{\|\mathbb{A}\mathbf{y}\|_2}{\|\mathbb{U}^*\mathbf{y}\|_2}. \end{aligned}$$

De plus, on a

$$\|\mathbb{U}^*\mathbf{y}\|_2^2 = \langle \mathbb{U}^*\mathbf{y}, \mathbb{U}^*\mathbf{y} \rangle = \langle \mathbf{y}, \mathbb{U}\mathbb{U}^*\mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{y} \rangle = \|\mathbf{y}\|_2^2$$

et donc  $\|\mathbb{A}\mathbb{U}\|_2 = \|\mathbb{A}\|_2$ .

- Montrons que  $\|\mathbb{A}\|_2 = \|\mathbb{U}^* \mathbb{A}\|_2$ .  
Ceci découle des deux égalités précédentes. En effet,

$$\begin{aligned} \|\mathbb{U}^* \mathbb{A}\|_2 &= \|\mathbb{U}^* (\mathbb{A}\mathbb{U})\|_2 = \|\mathbb{A}\mathbb{U}\|_2 && \text{car } \mathbb{U}^* \text{ unitaire} \\ &= \|\mathbb{A}\|_2 && \text{car } \mathbb{U} \text{ unitaire} \end{aligned}$$

~~~~~ Fin correction ~~~~~

## B.4 Listings

### B.4.1 Codes sur la méthode de dichotomie/bissection

Transcription de l'Algorithme 2.1 (page 18)

```

1 function x=dichotomie1(f,a,b,epsilon)
2   kmin=floor( ...
3     log((b-a)/epsilon)/log(2) );
4   X=zeros(kmin+1,1);
5   A=zeros(kmin+1,1);B=zeros(kmin+1,1);
6   A(1)=a;B(1)=b;X(1)=(a+b)/2;
7   for k=1:kmin
8     if f(X(k))==0
9       A(k+1)=X(k);B(k+1)=X(k);
10      elseif f(B(k))*f(X(k))<0
11        A(k+1)=X(k);B(k+1)=B(k);
12      else
13        A(k+1)=A(k);B(k+1)=X(k);
14      end
15      X(k+1)=(A(k+1)+B(k+1))/2;
16    end
17    x=X(kmin+1);
18  end

```

Listing B.1: fonction dichotomie1 (Matlab)

```

1 clear all
2 close all
3 f=@(x) (x+2)*(x+2)*(x-pi);
4 x=dichotomie1(f,-1,2*pi,1e-8);
5 fprintf('x=%.16f\n',x)
6 x=dichotomie1(@cos,2,pi,1e-8);
7 fprintf('x=%.16f\n',x)

```

Listing B.3: script dichotomie1 (Matlab)

```

double dichotomie1(double (*f)(double),
1   double a, double b, double eps){
2
3   double *A,*B,*X,x;
4   int kmin,k;
5   size_t Size;
6   assert(f(a)*f(b)<0);
7   kmin=(int) floor(log((b-a)/eps)/log(2.));
8   Size=(kmin+1)*sizeof(double);
9   assert( X=(double*)malloc( Size ) );
10  assert( A=(double*)malloc( Size ) );
11  assert( B=(double*)malloc( Size ) );
12  // ou assert( A ∈ B ∈ X );
13  A[0]=a;B[0]=b;X[0]=(a+b)/2.;
14  for (k=0;k<kmin;k++){
15    if ( f(X[k])==0){
16      A[k+1]=X[k];B[k+1]=X[k];
17    } else if ( f(B[k])*f(X[k])<0 ){
18      A[k+1]=X[k];B[k+1]=B[k];
19    } else {
20      A[k+1]=A[k];B[k+1]=X[k];
21    }
22    X[k+1]=(A[k+1]+B[k+1])/2;
23  }
24  x=X[kmin];
25  free(X);free(A);free(B);
26  return x;
27 }

```

Listing B.2: fonction dichotomie1 (C)

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>

double dichotomie1(
1   double (*f)(double),
2   double a, double b, double eps
3 );
4
5
6 double g1(double x){
7   return (x+2)*(x+2)*(x-M_PI);
8 }
9
10
11 int main(){
12
13   double x;
14   x=dichotomie1(g1,-1,2*M_PI,1e-8);
15   printf("x=%.16lf, error=%.6e\n",
16     x, fabs(x-M_PI));
17   x=dichotomie1(cos,-1,M_PI,1e-8);
18   printf("x=%.16lf, error=%.6e\n",
19     x, fabs(x-M_PI_2));
20   return 1;
21 }
22
23 // Definition de dichotomie1 ensuite ...
24

```

Listing B.4: main dichotomie1 (C)

## Transcription de l'Algorithme 2.5 (page 20)

```

1 function x=dichotomie5(f,a,b)
2   assert(f(a)*f(b)<0, ...
3         'test f(a)*f(b)<0 failed');
4   A=a;B=b;x=(A+B)/2;xp=A;
5   while x~xp
6     if f(B)*f(x)<0
7       A=x;
8     else
9       B=x;
10    end
11    xp=x;
12    x=(A+B)/2;
13  end
14 end

```

Listing B.5: fonction dichotomie5 (Matlab)

```

1 clear all
2 close all
3 f=@(x) (x+2)*(x+2)*(x-pi);
4 x=dichotomie5(f,-1,2*pi);
5 fprintf('x=%.16f\n',x)
6 x=dichotomie5(@cos,2,pi);
7 fprintf('x=%.16f\n',x)

```

Listing B.7: script dichotomie5 (Matlab)

```

1 double dichotomie5(double (*f)(double),
2                   double a, double b){
3   double A,B,x,xp;
4   assert(f(a)*f(b)<0);
5   A=a;B=b;x=(A+B)/2.;xp=A;
6   while (x!=xp){
7     if (f(B)*f(x)<0)
8       A=x;
9     else
10      B=x;
11     xp=x;
12     x=(A+B)/2;
13   }
14   return x;
15 }

```

Listing B.6: fonction dichotomie5 (C)

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <assert.h>
4
5 double dichotomie5(double (*f)(double),
6                   double a, double b);
7
8 double g1(double x){
9   return (x+2)*(x+2)*(x-M_PI);
10 }
11
12 int main(){
13   double x;
14   x=dichotomie5(g1,-1,2*M_PI);
15   printf("x=%.161f\n",x);
16   x=dichotomie5(cos,-1,M_PI);
17   printf("x=%.161f\n",x);

```

Listing B.8: main dichotomie5 (C)



## Liste des algorithmes

|      |  |    |
|------|--|----|
| 1.1  | Algorithme de calcul de $\pi$ , version naïve . . . . .  | 2  |
| 1.2  | Algorithme de calcul de $\pi$ , version stable . . . . .   | 9  |
| 2.1  | Méthode de dichotomie : version 1 . . . . .  | 18 |
| 2.2  | Méthode de dichotomie : version 2 . . . . .  | 19 |
| 2.3  | Méthode de dichotomie : version 3 . . . . .  | 19 |
| 2.4  | Méthode de dichotomie : version 4 . . . . .  | 20 |
| 2.5  | Méthode de dichotomie : version 5 . . . . .  | 20 |
| 2.6  | Méthode de point fixe : version <b>Tantque</b> <i>formel</i> . . . . .   | 30 |
| 2.7  | Méthode de point fixe : version <b>Répéter</b> <i>formel</i> . . . . .   | 30 |
| 2.8  | Méthode de point fixe : version <b>Tantque</b> <i>formel</i> avec critères d'arrêt . . . . .   | 31 |
| 2.9  | Méthode de point fixe : version <b>Répéter</b> <i>formel</i> avec critères d'arrêt . . . . .   | 31 |
| 2.10 | Méthode de point fixe : version <b>Tantque</b> avec critères d'arrêt . . . . .   | 31 |
| 2.11 | Méthode de point fixe : version <b>Répéter</b> avec critères d'arrêt . . . . .   | 31 |
| 2.12 | Méthode de la corde . . . . .  | 35 |
| 2.13 | Méthode de la corde utilisant la fonction <b>PtFixe</b> . . . . .  | 35 |
| 2.14 | Méthode de Newton . . . . .  | 41 |
| 2.15 | Méthode de Newton scalaire . . . . .   | 41 |
| 2.16 | Méthode de point fixe vectorielle . . . . .  | 50 |
| 2.17 | Méthode de Newton . . . . .  | 51 |
| 3.1  | Fonction <b>RSLMatDiag</b> permettant de résoudre le système linéaire à matrice diagonale inversible<br>$\mathbb{A}\mathbf{x} = \mathbf{b}.$ . . . . .   | 59 |
| 3.2  | Fonction <b>RSLTriInf</b> permettant de résoudre le système linéaire triangulaire inférieur inversible<br>$\mathbb{A}\mathbf{x} = \mathbf{b}.$ . . . . . | 61 |
| 3.3  | Fonction <b>RSLTriSup</b> permettant de résoudre le système linéaire triangulaire supérieur inversible<br>$\mathbb{A}\mathbf{x} = \mathbf{b}.$ . . . . . | 62 |
| 3.4  | Algorithme de Gauss-Jordan formel pour la résolution de $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .  | 69 |
| 3.5  | Algorithme de Gauss-Jordan avec fonctions pour la résolution de $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .  | 70 |
| 3.6  | Recherche d'un pivot pour l'algorithme de Gauss-Jordan. . . . .  | 70 |
| 3.7  | Permutte deux lignes d'une matrice et d'un vecteur. . . . .  | 70 |
| 3.8  | Combinaison linéaire $\mathcal{L}_i \leftarrow \mathcal{L}_i + \alpha\mathcal{L}_j$ appliqué à une matrice et à un vecteur. . . . .                      | 70 |

|      |  |     |
|------|--|-----|
| 3.9  | Fonction <b>RSLFactLU</b> permettant de résoudre, par une factorisation $\mathbb{L}\mathbb{U}$ , le système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ où $\mathbb{A}$ est une matrice de $\mathcal{M}_n(\mathbb{K})$ , dont toutes les sous-matrices principales sont inversibles, et $\mathbf{b} \in \mathbb{K}^n$ . . . . . | 78  |
| 3.10 | Fonction <b>FactLU</b> permet de calculer les matrices $\mathbb{L}$ et $\mathbb{U}$ dites matrice de factorisation $\mathbb{L}\mathbb{U}$ associée à la matrice $\mathbb{A}$ , telle que $\mathbb{A} = \mathbb{L}\mathbb{U}$   |     |
| 3.11 | Fonction <b>FactLUligU</b> permet de calculer la ligne $i$ de $\mathbb{U}$ à partir de (3.17) . . . . .  | 83  |
| 3.12 | Fonction <b>FactLUcolL</b> permet de calculer la colonne $i$ de $\mathbb{L}$ à partir de (3.18) . . . . .  | 83  |
| 3.13 | Fonction <b>FactLU</b> permet de calculer les matrices $\mathbb{L}$ et $\mathbb{U}$ dites matrice de factorisation $\mathbb{L}\mathbb{U}$ associée à la matrice $\mathbb{A}$ , telle que $\mathbb{A} = \mathbb{L}\mathbb{U}$ en utilisant des fonctions intermédiaires. . . . .  | 83  |
| 3.14 | Algorithme de base permettant de résoudre, par une factorisation de Cholesky positive, le système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ où $\mathbb{A}$ une matrice de $\mathcal{M}_n(\mathbb{C})$ hermitienne définie positive et $\mathbf{b} \in \mathbb{C}^n$ . . . . .  | 86  |
| 3.15 | Fonction <b>RSLCholesky</b> permettant de résoudre, par une factorisation de Cholesky positive, le système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ où $\mathbb{A}$ une matrice hermitienne de $\mathcal{M}_n(\mathbb{C})$ définie positive et $\mathbf{b} \in \mathbb{C}^n$ . . . . .                                       | 86  |
| 3.16 | Fonction <b>Cholesky</b> permettant de calculer la matrice $\mathbb{B}$ , dites matrice de factorisation positive de Cholesky associée à la matrice $\mathbb{A}$ , telle que $\mathbb{A} = \mathbb{B}\mathbb{B}^*$ . . . . .   | 89  |
| 3.17 | Calcul du $\alpha$ et de la matrice de Householder $\mathbb{H}(\mathbf{u})$ telle que $\mathbb{H}(\mathbf{u})\mathbf{a} = \alpha\mathbf{b}$ . . . . .  | 93  |
| 3.18 | Fonction <b>FactQR</b> . . . . .   | 99  |
| 3.19 | Méthode itérative pour la résolution d'un système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .   | 119 |
| 3.20 | Méthode itérative de Jacobi pour la résolution d'un système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .   | 120 |
| 3.21 | Méthode itérative de Gauss-Seidel pour la résolution d'un système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .   | 122 |
| 3.22 | Itération de Jacobi : calcul de $\mathbf{x}$ tel que $x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n A_{ij}y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$ . . . . .   | 123 |
| 3.23 | Itération de Gauss-Seidel : calcul de $\mathbf{x}$ tel que $x_i = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}y_j \right), \quad \forall i \in \llbracket 1, n \rrbracket.$ . . . . .  | 123 |
| 3.24 | Méthode itérative pour la résolution d'un système linéaire $\mathbb{A}\mathbf{x} = \mathbf{b}$ . . . . .   | 124 |
| 3.25 | Itération S.O.R. : calcul de $\mathbf{x}$ tel que $x_i = \frac{w}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=i+1}^n A_{ij}y_j \right) + (1-w)y_i$ . . . . .   | 125 |
| 4.1  | Fonction <b>Lagrange</b> permettant de calculer le polynôme d'interpolation de Lagrange $\mathcal{P}_n(x)$ défini par (4.2) . . . . .  | 139 |
| 4.2  | Fonction <b>Hermite</b> permettant de calculer le polynôme d'interpolation de Lagrange-Hermite $H_n(t)$ défini par (4.17) . . . . .  | 152 |
| 4.3  | Fonction <b>polyA</b> permettant de calculer le polynôme $A_i$ en $t \in \mathbb{R}$ donné par $A_i(t) = (1 - 2L'_i(x_i)(t - x_i))L_i^2(t)$ . . . . .  | 153 |
| 4.4  | Fonction <b>polyB</b> permettant de calculer le polynôme $B_i$ en $t \in \mathbb{R}$ donné par $B_i(t) = (t - x_i)L_i^2(t)$  | 153 |
| 4.5  | Fonction <b>polyL</b> permettant de calculer le polynôme $L_i$ en $t \in \mathbb{R}$ donné par $L_i(t) = \prod_{j=0, j \neq i}^n \frac{t - x_j}{x_i - x_j}$ . . . . .  | 153 |
| 4.6  | Fonction <b>polyLp</b> permettant de calculer $L'_i(x_i) = \sum_{k=0, k \neq i}^n \frac{1}{x_i - x_k}$ . . . . .   | 153 |
| 5.1  | Fonction <b>WeightsFromPoints</b> retournant le tableau des poids $\mathbf{w}$ associé à un tableau de points $\mathbf{x}$ donnés (points 2 à 2 distincts) appartenant à un intervalle $[a, b]$ . . . . .  | 168 |
| 5.2  | Fonction <b>WeightsPointsNC</b> retournant le tableau de points $\mathbf{x}$ donnés correspondant à la discrétisation régulière intervalle $[a, b]$ . et le tableau des poids $\mathbf{w}$ associé à un . . . . .  | 177 |
| 5.3  | Fonction <b>QuadElemGen</b> retourne la valeur de $I = (b - a) \sum_{j=0}^n w_j f(x_j)$ . . . . .  | 178 |
| 5.4  | Fonction <b>QuadElemGen</b> retourne la valeur de $I = (b - a) \sum_{j=0}^n w_j f(x_j)$ où les poids $w_i$ et les points $x_i$ sont ceux définis par la formule de quadrature élémentaire de Newton-Cotes . . . . .  | 178 |
| 5.5  | Fonction <b>GaussLegendre</b> retournant le tableau des points $\mathbf{t}$ et le tableau des poids $\mathbf{w}$ . . . . .   | 187 |
| 5.6  | Fonction <b>QuadElemGaussLegendre</b> retournant une approximation de $\int_a^b f(x)dx$ en utilisant la formule de quadrature de Gauss-Legendre à $n + 1$ points sur l'intervalle $[a, b]$ . . . . .   | 187 |

|      |  |     |
|------|--|-----|
| 5.7  | Fonction <code>QuadSimpson</code> retourne une approximation de l'intégrale d'une fonction $f$ sur l'intervalle $[\alpha, \beta]$ utilisant la méthode de quadrature composée de Simpson en <b>minimisant</b> le nombre d'appels à la fonction $f$ . | 191 |
| A.1  | Exemple de boucle «pour»   | 201 |
| A.2  | Exemple de boucle «tant que»   | 202 |
| A.3  | Exemple de boucle «répéter ...jusqu'à»   | 202 |
| A.4  | Exemple d'instructions conditionnelle «si»   | 202 |
| A.5  | Exemple de fonction : Résolution de l'équation du premier degré $ax + b = 0$ .   | 203 |
| A.6  | Fonction <code>sum</code> permettant de calculer $\sum_{i=0}^n u_i$ .  | 205 |
| A.7  | Fonction <code>sumrows</code>  | 206 |
| A.8  | Calcul de $S = \sum_{k=1}^n k \sin(2kx)$   | 207 |
| A.9  | Calcul de $P = \prod_{n=1}^n \sin(2kz/n)^k$  | 208 |
| A.10 | En-tête de la fonction SFT retournant valeur de la série de Fourier en $t$ tronquée au $n$ premiers termes de l'exercice 1.2.3.  | 208 |
| A.11 | Fonction SFT retournant la valeur de la série de Fourier en $t$ tronquée au $n$ premiers termes de l'exercice 1.2.3.   | 209 |
| B.1  | Fonction <code>dot</code> , produit scalaire entre deux vecteurs.  | 232 |
| B.2  | Fonction <code>norm2</code>  | 232 |
| B.3  | Fonction <code>aUpbV</code>  | 233 |
| B.4  | Fonction <code>VecZeros</code> retournant le vecteur nul de $\mathbb{R}^n$ .   | 234 |
| B.5  | Fonction <code>VecConst</code> retournant le vecteur $\mathbf{v}$ de $\mathbb{R}^n$ dont toutes les composantes valent $\alpha \in \mathbb{R}$ .   | 234 |
| B.6  | Fonction <code>VecPlusConst</code> retournant le vecteur $\mathbf{v}$ de $\mathbb{R}^n$ tel que $\forall i \in \llbracket 1, n \rrbracket, v_i = u_i + \alpha$ avec  | 234 |
| B.7  | Fonction <code>VecRand</code> retournant un vecteur dont toutes les composantes sont aléatoires suivant la loi uniforme sur $[a, b]$ .   | 235 |
| B.8  | Fonction <code>MatZeros</code> la matrice nulle de $\mathcal{M}_{m,n}(\mathbb{R})$ .   | 236 |
| B.9  | Fonction <code>MatConst</code> retournant la matrice de $\mathcal{M}_{m,n}(\mathbb{R})$ dont toutes les composantes valent $\alpha \in \mathbb{R}$ .   | 236 |
| B.10 | Fonction <code>MatPlusConst</code> retournant la matrice $\mathbb{B} \in \mathcal{M}_{m,n}(\mathbb{R})$ telle que $\forall (i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket, B_{i,j} = A_{i,j} + \alpha$ .                    | 236 |
| B.11 | Fonction <code>MatRand</code> retournant une matrice de $\mathcal{M}_{m,n}(\mathbb{R})$ dont toutes les composantes sont aléatoires suivant la loi uniforme sur $[a, b]$ .   | 237 |
| B.12 | Fonction <code>setMatCol</code>  | 238 |
| B.13 | Fonction <code>getMatCol</code>  | 238 |
| B.14 | Fonction <code>setMatRow</code>  | 238 |
| B.15 | Fonction <code>getMatRow</code>  | 239 |
| B.16 | Fonction <code>aApbB</code> retournant la matrice $\mathbb{W} \stackrel{\text{def}}{=} \alpha \mathbb{A} + \beta \mathbb{B}$ .   | 240 |
| B.17 | Fonction <code>ProdMatVec</code> retournant le vecteur $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice $\mathbb{A}$ par le vecteur $\mathbf{u}$ .   | 241 |
| B.18 | Fonction <code>ProSca</code> , produit scalaire entre deux vecteurs.   | 243 |
| B.19 | Fonction <code>ProdMatVec1</code> retournant le vecteur $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice $\mathbb{A}$ par le vecteur $\mathbf{u}$ .  | 244 |
| B.20 | Fonction <code>ProdMatVecFun</code> retournant le vecteur $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice $\mathbb{A}$ par le vecteur $\mathbf{u}$ .  | 244 |
| B.21 | Fonction <code>ProdMatVecFun</code> retournant le vecteur $\mathbf{v} \stackrel{\text{def}}{=} \mathbb{A}\mathbf{u}$ , produit de la matrice $\mathbb{A}$ par le vecteur $\mathbf{u}$ .  | 245 |
| B.22 | Fonction <code>ProdMatMatFun</code> retournant la matrice $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice $\mathbb{A}$ par la matrice $\mathbb{B}$ .  | 246 |
| B.23 | Fonction <code>ProdMatMatFun1</code> retournant la matrice $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice $\mathbb{A}$ par la matrice $\mathbb{B}$ .   | 246 |
| B.24 | Fonction <code>ProdMatMatSim1</code> retournant la matrice $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice $\mathbb{A}$ par la matrice $\mathbb{B}$ .   | 247 |
| B.25 | Fonction <code>ProdMatMatFun2</code> retournant la matrice $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice $\mathbb{A}$ par la matrice $\mathbb{B}$ .   | 247 |
| B.26 | Fonction <code>ProdMatMatSim2</code> retournant la matrice $\mathbb{G} \stackrel{\text{def}}{=} \mathbb{A}\mathbb{B}$ , produit de la matrice $\mathbb{A}$ par la matrice $\mathbb{B}$ .   | 248 |
| B.27 |  | 255 |
| B.28 | Fonction <code>Norm1</code> permettant de calculer $\ \mathbb{A}\ _1$  | 277 |

|  |     |
|--|-----|
| B.29 Fonction <code>NormInf</code> permettant de calculer $\ A\ _\infty$ . . . . . | 279 |
|--|-----|

# Index

- $(E, d)$ , 213
- $\mathbb{K}$ , 217
- $\delta_{i,j}$ , 218
- $\mathcal{O}(g)$ , 212
- $\mathcal{O}(h^p)$ , 212
- $\det(A)$ , 221
- $\text{im}(A)$ , 219
- $\ker(A)$ , 219
- $\text{rank}(A)$ , 219
- $\mathcal{M}_{m,n}$ , 218
- $\text{Sp}(A)$ , 222
- $\rho(A)$ , 222
- $\text{tr}(A)$ , 221
- $\mathbb{1}$  ou  $\mathbb{1}_n$ , 219
- $\mathbb{O}_{m,n}$ , 218
  
- Adhérence, 214
  
- base, 217
- base orthogonale, 218
- Boule
  - fermée, 213
  - ouverte, 213
  
- Compacte, 214
- Convergence
  - cubique, 216
  - linéaire, 216
  - ordre, 216
  - quadratique, 216
  - sous-linéaire, 216
  - super-linéaire, 216
- Convergence:ordre, 216
  
- demi largeur de bande, 223
- diagonale, 223
- diagonale dominante, 223
  
- diagonale strictement dominante, 223
- diagonalisable, 228
- Distance, 212
- Déterminant, 221
  
- Élément propre, 222
- Ensemble
  - adhérence, 214
  - Borné, 213
  - fermé, 213
  - intérieur, 214
- Ensemble:compact, 214
- Ensemble:orthonormal, 218
- Ensemble:ouvert, 213
- Ensemble:voisinage, 213
- Espace de Banach, 216
- Espace métrique, 213
  - complet, 216
- espace vectoriel normé, 99, 225
  
- Fermé, 213
- Fonction
  - continuité, 214
  - continuité uniforme, 215
  - contractante, 215
  - limite, 214
  - lipschitzienne, 215
- Formule
  - Taylor-Lagrange, 212
  - Taylor-Landau, 212
  
- grand  $\mathcal{O}$ , 212
- groupe des permutations, 221
  
- Hermitienne, 220
  
- Identité, 219

- Intérieur, 214
- Inverse, 219
- Inversible, 219
- Inégalité de Cauchy-Schwarz, 100
- Inégalité de Hölder, 100, 226
- Kronecker, 218
- Matrice
  - adjointe, 218
  - définie positive, 220
  - déterminant, 221
  - élément propre, 222
  - hermitienne, 220
  - identité, 219
  - image, 219
  - inverse, 219
  - inversible, 219
  - non inversible, 219
  - normale, 220
  - noyau, 219
  - nulle  $\mathbb{O}_{m,n}$ , 218
  - orthogonale, 220
  - permutation, 221
  - polynôme caractéristique, 222
  - produit de, 219
  - projection orthogonale, 218
  - rang, 219
  - rayon spectrale, 222
  - régulière, 219
  - semi définie positive, 220
  - singulière, 219
  - sous-espace propre, 222
  - spectre, 222
  - symétrique, 220
  - trace, 221
  - transposée, 218
  - unitaire, 220
  - valeur propre, 221
  - vecteur propre, 222
- matrice
  - bande, 223
  - bloc, 224
  - bloc-carrée, 224
  - diagonale, 223
  - diagonale dominante, 223
  - diagonale par blocs, 224
  - diagonale strictement dominante, 223
  - diagonalisable, 228
  - matrice de passage..., 228
  - norme, 100, 226
  - pentadigonale, 223
  - sous matrice, 223
  - sous matrice principale, 223
  - triangulaire, 223
  - triangulaire inférieure, 223
  - triangulaire par blocs, 224
  - triangulaire supérieure, 223
  - tridiagonale, 223
- Normale, 220
- norme
  - invariance par transformation unitaire, 103, 227, 276
  - matricielle non subordonnée, 104, 227
  - matricielle subordonnée, 100, 226
  - vectorielle, 99, 225
- norme matricielle, 100, 226
- normes
  - équivalentes, 100, 226
- Opérateur de projection, 218
- Ordre de convergence, 216
- Orthogonale, 220
- Ouvert, 213
- Partie
  - bornée, 213
- permutations, 221
- Point
  - adhérent, 214
  - intérieur, 214
- Polynôme caractéristique, 222
- Projection orthogonale:opérateur, 218
- Rayon spectral, 222
- Régulière, 219
- Singulière, 219
- sous matrice, 223
- sous matrice principale, 223
- Sous-espace propre, 222
- Spectre, 222
- Suite
  - convergente, 213
  - divergente, 213
- Suite de Cauchy, 215
- Symétrique, 220
- Théorème
  - accroissements finis, 211
  - Bolzano, 211
  - de bijection, 212
  - de Heine, 215
  - des valeurs intermédiaire (TVI), 211
  - Rolle, 211
- Trace d'une matrice, 221
- triangulaire, 223
- triangulaire inférieure, 223
- triangulaire supérieure, 223
- Unitaire, 220
- Valeur propre, 221
  - multiplicité algébrique, 222
  - multiplicité géométrique, 222
- Vecteur
  - adjoint, 217
  - nul  $\mathbf{0}_n$  ou  $\mathbf{0}$ , 218
  - orthogonal à une partie, 217

orthogonaux, 217  
orthonormaux, 218  
produit scalaire, 217  
propre, 222  
transposé, 217  
vecteur  
  colonne, 217  
  convergence, 106, 229  
  ligne, 217  
Voisinage, 213



## Bibliography

- [1] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. DUNOD, 2006.
- [2] M. Crouzeix and A.L. Mignot. *Analyse numérique des équations différentielles*. Mathématiques appliquées pour la maîtrise. Masson, 1992.
- [3] J.-P. Demailly. *Analyse numérique et équations différentielles*. Grenoble Sciences. EDP Sciences, 2006.
- [4] J.P. Demailly. *Analyse Numérique et Equations Différentielles*. PUG, 1994.
- [5] W. Gander, M.J. Gander, and F. Kwok. *Scientific computing : an introduction using Maple and MATLAB*. Springer, Cham, 2014.
- [6] T. Huckle. Collection of software bugs <http://www.zenger.informatik.tu-muenchen.de/persons/huckle/bugse.html>.
- [7] P. Lascaux and R. Théodor. *Analyse numérique matricielle appliquée à l'art de l'ingénieur*. Number vol. 1 et 2 in *Analyse numérique matricielle appliquée à l'art de l'ingénieur*. Dunod, 2004.