

Guillaume Connan

VISA POUR LA PRÉPA 2021-2022

MATHS ET INFORMATIQUE

MPSI - MP2I - PCSI - PTSI
BCPST - ECG

DUNOD

Couverture : création Hokus Pokus, adaptation Studio Dunod

© Dunod, 2021
11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-082980-4

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

1. Savez-vous raisonner ?	1
1.1 Un brin de logique	1
1.2 Raisonnement par récurrence	5
1.3 Ensembles	7
1.4 Ordre dans \mathbb{R}	11
1.5 Exercices	15
Coups de pouce	19
Solutions et commentaires	20
2. Savez-vous calculer ?	25
2.1 De l'importance de savoir calculer	25
2.2 Calcul de sommes et de produits	25
2.3 Exercices	29
Coups de pouce	37
Solutions et commentaires	39
3. Les suites	67
3.1 Convergence d'une suite	67
3.2 Suites adjacentes	77
3.3 Suites récurrentes d'ordre 1	79
3.4 Exercices	84
Coups de pouce	91
Solutions et commentaires	92

4. Savez-vous intégrer ?	103
4.1 Mise en place d'une définition	103
4.2 Quelles sont les fonctions intégrables ?	107
4.3 Propriétés de l'intégrale	109
4.4 Valeur moyenne	110
4.5 Primitive et intégrale	112
4.6 Exercices	114
Coups de pouce	119
Solutions et commentaires	120
5. Savez-vous prévoir ?	127
5.1 Rappels de théorie des ensembles	127
5.2 Une dose d'algèbre générale	128
5.3 Quelques résultats sur les cardinaux	130
5.4 Dénombrément	131
5.5 Triangle de PASCAL – Binôme de NEWTON	133
5.6 Probabilités ?	134
5.7 Avant la formalisation	134
5.8 Espace probabilisable – Espace probabilisé	135
5.9 Probabilités conditionnelles	138
5.10 Variables aléatoires réelles finies	141
5.11 Quelques lois discrètes classiques	146
5.12 Exercices	149
Coups de pouce	157
Solutions et commentaires	159

6. Savez-vous programmer ?

175

6.1	Avant de commencer	175
6.2	Premiers pas	177
6.3	Un peu de vocabulaire	178
6.4	Les types de base	182
6.5	Instructions et expressions conditionnelles	186
6.6	Elif	187
6.7	Survol des structures de données	187
6.8	Les boucles	195
6.9	Les fonctions	199
6.10	Exercices	206
	Coups de pouce	216
	Solutions et commentaires	218

1.1 Un brin de logique

« Contrariwise, continued Tweedledee, if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic. »

Lewis CARROLL – *De l'autre côté du miroir*

Contexte

On appellera **proposition** tout énoncé dont on peut décider s'il est vrai ou faux indépendamment de ses composantes.

On distinguera la logique des propositions de la logique des **prédicats**, qui introduit des variables dans les assertions pouvant les rendre donc parfois vraies et parfois fausses.

Par exemple « *Je suis actuellement en salle 3* » est une proposition susceptible d'être vraie ou fausse dans toute situation alors que la valeur de vérité de « *x est actuellement en salle 3* » dépend de *x*.

En informatique, **Vrai** et **Faux** sont appelés des **booléens**, notés `True` et `False` en **Python** :

```
>>> 2 > 3
False
>>> 2 < 3
True
>>> 2 + 2 == 4
True
>>> len("Papa") == 2 + 2
True
```

Implication

Définition : implication

A et B étant deux propositions, on note $A \Rightarrow B$ (A implique B) la proposition qui est toujours vraie sauf quand A est vraie et B est fausse.

On comprend alors mieux cette citation de Pierre DAC (dans *Y'a du mou dans la corde à nœuds*) : « Avec le mot *SI* on peut faire tout ce qu'on ne peut pas faire. »

On peut résumer la situation dans une **table de vérité** car les propositions A et B ne peuvent prendre que deux valeurs donc il n'y a que quatre possibilités pour les valeurs du couple (A,B) :

A	B	C
F	F	V
F	V	V
V	V	V
V	F	F

Ainsi, le cas où A est fausse ne nous intéresse pas vraiment car on est sûr que l'implication sera vraie quelle que soit la valeur de vérité de B .

Ainsi, $A \Rightarrow B$ étant vraie, on peut dire que :

- SI A (est vraie) ALORS B (est vraie).
- A est vraie SEULEMENT SI B est vraie.
- IL SUFFIT que A soit vraie pour que B soit vraie.
- IL FAUT que B soit vraie pour que A soit vraie.
- A est une **condition suffisante** de B .
- B est une **condition nécessaire** de A .

Exemple

Soit A la proposition : x est un entier naturel. Soit B la proposition : x est positif.

Les propositions suivantes sont vraies :

- SI x est un entier naturel ALORS x est positif.
- IL SUFFIT que x soit un entier naturel pour que x soit positif.
- IL FAUT que x soit positif pour que x soit un entier naturel.
- x est un entier naturel SEULEMENT SI x est positif.
- x entier naturel est une **condition suffisante** de x positif.
- x positif est une **condition nécessaire** de x entier naturel.

On a bien « x est un entier naturel » implique que x est positif ($A \Rightarrow B$ est vraie).

Remarque

$B \Rightarrow A$ est la proposition **réciproque** de $A \Rightarrow B$.

— Contraposée

Reprenons notre exemple : SI x n'est pas positif ALORS x ne peut pas être un entier naturel.

On note $\neg A$ la **négation** de A . L'observation précédente signifie que $\neg B \Rightarrow \neg A$. Est-ce toujours le cas? Dressons la table !

A	B	$A \Rightarrow B$	$\neg B$	$\neg A$	$\neg B \Rightarrow \neg A$
F	F	V	V	V	V
F	V	V	F	V	V

A	B	$A \Rightarrow B$	$\neg B$	$\neg A$	$\neg B \Rightarrow \neg A$
V	V	V	F	F	V
V	F	F	V	F	F

Ainsi, dans tous les cas, $A \Rightarrow B$ et $\neg B \Rightarrow \neg A$ ont la même valeur de vérité.



Ainsi, dans certains cas, pour démontrer une implication, il sera parfois plus simple de démontrer la contraposée.

Remarque

Dans un théorème du cours de mathématiques sous la forme *SI hypothèse ALORS conclusion*, on a en fait une implication « hypothèse \Rightarrow conclusion ».

- Si on a l'hypothèse vérifiée et le théorème démontré alors on peut en déduire que la conclusion est vraie. C'est le *modus ponens*.
- Si on a le contraire de la conclusion vérifié et le théorème démontré, alors on peut en déduire que le contraire de l'hypothèse est vrai. C'est le *modus tollens*.
- Par exemple, en supposant que la proposition « x est un entier naturel implique que x est positif » est un théorème, 3 étant un entier naturel, on en déduit que 3 est positif.
- On peut aussi dire que comme -3 n'est pas positif, on en déduit que -3 n'est pas un entier naturel.

— Contraire (négation) d'une proposition

On voudrait savoir si la proposition « un nombre est positif implique que ce nombre est un entier naturel » est vraie.

On « sent » bien que non. Par exemple $\sqrt{2}$ est bien positif (hypothèse vraie) mais pourtant ce n'est pas un entier naturel (conclusion fausse).

Si on formalise, on a $A \text{ ET } \neg B$. Est-ce que par hasard $A \text{ ET } \neg B$ ne serait pas la négation de $A \Rightarrow B$?

C'est quoi ET ? Pour être en accord avec le *bon sens*, on dira que $A \text{ ET } B$ est vrai seulement quand les deux propositions A et B sont simultanément vraies. On note cet opérateur \wedge .

A	B	$\neg B$	$A \wedge \neg B$	$A \Rightarrow B$
F	F	V	F	V
F	V	F	F	V
V	V	F	F	V
V	F	V	V	F

Bingo ! Les deux propositions sont bien contraires. Or une proposition et son contraire ne peuvent pas être vraies simultanément (c'est ce qu'on appelle le **tiers exclu**).

Puisque $A \wedge \neg B$ est vrai (dans le cas de $\sqrt{2}$), c'est que son contraire, $A \Rightarrow B$, est faux.

Quantificateurs

Un pour tous : \forall

Il y a quelque chose de pas très clair cependant. On a parlé d'un certain x , ou d'un certain entier... C'est un peu flou.

Il vaudrait mieux écrire que notre théorème est en fait :

« Quel que soit le nombre x , si x est un entier naturel alors il est positif. »

Plus formellement, on a l'habitude de noter cela :

$$(\forall x)(x \in \mathbb{N} \Rightarrow x \geq 0)$$

Ce symbole \forall se lit « pour tout » (notation introduite par l'Allemand GENTZEN en 1933 : « pour tout » se dit *für Alle* en allemand, d'où le A à l'envers).

Un pour l'existence : \exists

Nous avons établi qu'il existait **au moins** un nombre positif ($\sqrt{2}$) qui n'était pas un entier naturel. Cela s'écrit formellement :

$$(\exists x)(x \geq 0 \wedge x \notin \mathbb{N})$$

Ce symbole \exists se lit « il existe *au moins* » (on le doit au britannique Bertrand RUSSELL dans ce contexte).

Il existe une variante : $(\exists!)$ qui signifie « il existe un UNIQUE élément ».

Quantificateurs et négation

Quand ce n'est pas vrai pour tous, c'est que c'est faux pour au moins un.

Quand ce n'est pas vrai pour au moins un, c'est que c'est faux pour tous.

Ainsi, si on note $P(x)$ une certaine propriété dépendant de x :

$$\neg((\exists x)(P(x))) \equiv (\forall x)(\neg P(x))$$

et

$$\neg((\forall x)(P(x))) \equiv (\exists x)(\neg P(x))$$

avec \equiv significatif « a la même valeur de vérité que ».

La dernière proposition renvoie à la recherche de **contre-exemple** : pour montrer que quelque chose n'est pas toujours vrai, il faut exhiber un cas où cette chose est fausse.



À retenir Bien repérer les quantificateurs dans les énoncés de propositions permet de donner des pistes de démonstrations.

Remarque

Souvent, on contracte l'écriture $(\forall x)(x \in \mathbb{R})$ en $(\forall x \in \mathbb{R})$.

Réciproque/Équivalence

Lorsqu'on a établi qu'une implication était vraie, on a souvent envie de vérifier l'implication *réciproque* (i.e. avec la « flèche dans l'autre sens »).

Si c'est le cas, on utilise le symbole \Leftrightarrow :

$$(A \Rightarrow B) \wedge (B \Rightarrow A) \equiv (A \Leftrightarrow B)$$

On lit « A est équivalent à B » ou « A **si, et seulement si**, B ».

Exemple

On voudrait vérifier que deux nombres sont égaux **si, et seulement si**, leurs carrés sont égaux.

Nous voudrions donc voir si la proposition suivante est vraie :

$$(\forall x \in \mathbb{R})(\forall y \in \mathbb{R})(x^2 = y^2 \Leftrightarrow x = y)$$

Pour démontrer une équivalence, on procède souvent par **double inclusion**.

- $x = y \Rightarrow x^2 = y^2$. Ce sens est assez évident puisque, pour tous réels x et y : $x^2 = x \times x = y \times y = y^2$ (car $x = y$). Ainsi :

$$(\forall x \in \mathbb{R})(\forall y \in \mathbb{R})(x = y \Rightarrow x^2 = y^2)$$

- $x = y \Leftarrow x^2 = y^2$. On sent qu'il y a un problème en tant qu'élève bien entraîné(e). En effet, $(-2)^2 = 2^2$ et pourtant $-2 \neq 2$. Plus formellement :

$$(\exists x \in \mathbb{R})(\exists y \in \mathbb{R})(x^2 = y^2 \wedge \neg(x = y))$$

ou encore :

$$\neg((\forall x \in \mathbb{R})(\forall y \in \mathbb{R})(x^2 = y^2 \Rightarrow x = y))$$

ce qui traduit que l'implication dans ce sens est fausse, donc que l'équivalence est fausse.

1.2 Raisonnement par récurrence

— Réursion, récurrence, induction

...des mots qui tournent autour du même thème mais qu'il faut savoir distinguer.

- En informatique, une *fonction réursive* (ou un *type réursif*) est une fonction (ou un type) qui fait référence à elle(lui)-même dans sa définition. Ce mécanisme est très puissant et permet de condenser l'écriture d'un programme.

C'est le principe de la mise en abyme... comme dans le dessin de La Vache qui rit®...

Par exemple, pour calculer la somme S_n des entiers de 1 à n , on peut dire que $S_1 = 1$ et que, pour tout entier $n \geq 2$, $S_n = n + S_{n-1}$. Cela donne en **Python** :

```
>>> def somme(n: int) -> int:
    if n == 1:
        return 1
    else:
        return n + somme(n - 1)
>>> somme(5)
15
```

- L'*induction* consiste à conclure à partir d'observations préalables : cela correspond à la démarche expérimentale classique. Sans précautions, cela peut conduire à certaines contre-vérités. Par exemple 3, 5 et 7 sont premiers donc on pourrait en déduire, par induction, que tous les nombres impairs à partir de 3 sont premiers, et pourtant...
- L'*induction mathématique* ou *raisonnement par récurrence* corrige ce défaut. On part toujours d'un résultat observé mais on **démontre** qu'il est vrai pour tous les éléments d'un ensemble donné, éventuellement infini. C'est l'induction expérimentale corrigée par la rigueur du raisonnement mathématique.

Principes de récurrence

C'est l'induction mathématique qui nous intéresse ici.

Théorème : principe de récurrence simple

Soit \mathcal{P}_n une proposition ne dépendant que d'un entier n .

SI :

- il existe un entier n_0 tel que \mathcal{P}_{n_0} soit vraie ;
- $(\forall n \geq n_0)(\mathcal{P}_n \Rightarrow \mathcal{P}_{n+1})$

ALORS :

$$(\forall n \geq n_0)(\mathcal{P}_n)$$

Par exemple, nous allons essayer de prouver que la proposition bien connue suivante est vraie pour tout entier naturel n :

$$\mathcal{P}_n : \ll 0 + 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} \gg$$

- Il est facile de vérifier que $0 = \frac{0(0+1)}{2}$, donc la deuxième étape de notre raisonnement est vérifiée, \mathcal{P}_0 est vraie.
- Supposons qu'une « génération », appelons-la par exemple la k -ième, soit « infectée ». Plus sobrement, on dira : soit k un entier supérieur à 1. Supposons que \mathcal{P}_k soit vraie et essayons alors de montrer que cela implique que la génération suivante, la $k+1$ -ième, sera elle aussi infectée, ce qui impliquera que : $\mathcal{P}_k \Rightarrow \mathcal{P}_{k+1}$.

Il s'agit donc de calculer $1 + 2 + 3 + \dots + k + (k+1)$ sachant que $1 + 2 + 3 + \dots + k = \frac{k(k+1)}{2}$, or

$$\begin{aligned} \mathcal{P}_k \Rightarrow 1 + 2 + 3 + \dots + k + (k+1) &= \frac{k(k+1)}{2} + (k+1) = (k+1) \left(\frac{k}{2} + 1 \right) \\ &= (k+1) \frac{k+2}{2} = \frac{(k+1)(k+1+1)}{2} \end{aligned}$$

Nous en déduisons que $(\forall k \in \mathbb{N})(\mathcal{P}_k \Rightarrow \mathcal{P}_{k+1})$.

- Nous avons vérifié que la proposition était vraie au rang 0 et qu'elle était héréditaire. Nous en déduisons que la proposition sera toujours vraie, quel que soit l'entier naturel n grâce au principe de récurrence simple.

Théorème : principe de récurrence à deux pas

Soit \mathcal{P}_n une proposition ne dépendant que d'un entier n .

SI :

- il existe un entier n_0 tel que \mathcal{P}_{n_0} et \mathcal{P}_{n_0+1} soient vraies ;
- $(\forall n \geq n_0)(\mathcal{P}_n \wedge \mathcal{P}_{n+1} \Rightarrow \mathcal{P}_{n+2})$

ALORS :

$$(\forall n \geq n_0)(\mathcal{P}_n)$$

Par exemple, on pose $F_0 = F_1 = 1$ et $(\forall n \geq 0)(F_{n+2} = F_{n+1} + F_n)$ (c'est la suite dite de FIBONACCI).

On veut démontrer que $(\forall n \geq 0)(0 \leq F_n \leq \left(\frac{5}{3}\right)^n)$:

- Notons, pour tout entier naturel n , \mathcal{P}_n : « $0 \leq F_n \leq \left(\frac{5}{3}\right)^n$ ».

- On vérifie aisément que \mathcal{P}_0 et \mathcal{P}_1 sont vraies.

- $\mathcal{P}_n \wedge \mathcal{P}_{n+1} \Rightarrow 0 + 0 \leq F_{n+2} \leq \left(\frac{5}{3}\right)^n + \left(\frac{5}{3}\right)^{n+1}$

$$\text{Or } \left(\frac{5}{3}\right)^n + \left(\frac{5}{3}\right)^{n+1} = \left(\frac{5}{3}\right)^{n+2} \times \left(\frac{3}{5} + \left(\frac{3}{5}\right)^2\right) = \left(\frac{5}{3}\right)^{n+2} \times \frac{24}{25} \leq \left(\frac{5}{3}\right)^{n+2} \times 1$$

On a donc démontré que $(\forall n \geq 0)(\mathcal{P}_n \wedge \mathcal{P}_{n+1} \Rightarrow \mathcal{P}_{n+2})$.

- Ainsi, d'après le principe de récurrence à deux pas, \mathcal{P}_n est vraie pour tout entier naturel n .

Théorème : principe de récurrence forte

Soit \mathcal{P}_n une proposition ne dépendant que d'un entier n .

SI :

- il existe un entier n_0 tel que \mathcal{P}_{n_0} soit vraie ;
- $(\forall n \geq n_0)(\mathcal{P}_{n_0} \wedge \mathcal{P}_{n_0+1} \wedge \dots \wedge \mathcal{P}_{n-1} \wedge \mathcal{P}_n \Rightarrow \mathcal{P}_{n+1})$

ALORS :

$$(\forall n \geq n_0)(\mathcal{P}_n)$$

On pourra par exemple démontrer que $(\forall n \in \mathbb{N}^*)(a_n = 3n)$ avec $a_1 = 3$ et

$$(\forall n \in \mathbb{N}^*)(a_{n+1} = \frac{2}{n} \sum_{k=1}^n a_k).$$

1.3 Ensembles

Définition

La théorie des ensembles a bouleversé les mathématiques à partir des travaux de Georg CANTOR dès la fin du XIX^e siècle. Cette section n'est qu'une très courte introduction. Les curieux pourront, outre CANTOR, aller découvrir qui était RUSSELL que nous avons déjà évoqué.

Nous nous contenterons de la définition donnée par CANTOR lui-même :

Définition : ensemble

Par ensemble, nous entendons toute collection M d'objets m de notre intuition ou de notre pensée, définis et **distincts**, ces objets étant appelés les **éléments** de M .

On note $m \in M$ qu'on lit « m appartient à M ».

Ou encore $M \ni m$ qu'on lit « M contient m ».

Inclusion

Définition : inclusion

Si $(\forall x)(x \in A \Rightarrow x \in B)$ alors on note cela $A \subseteq B$ ou $B \supseteq A$ et on lit « **A est inclus dans B** » ou « **B contient A** ».



Vous noterez la correspondance entre inclusion des ensembles et implication des appartenances.

Remarques

- On devrait distinguer \subsetneq , l'inclusion stricte, et \subseteq , l'inclusion au sens large. Dans la plupart des textes cependant, on utilise \subset pour représenter les deux cas.
- Pour dire qu'un ensemble **n'est pas** inclus dans un autre, on utilise le symbole $\not\subseteq$ (à ne pas confondre avec \subsetneq).
- Pour démontrer que l'ensemble A n'est pas inclus dans l'ensemble B , on doit nier la définition de l'inclusion : $\neg((\forall x)(x \in A \Rightarrow x \in B)) \equiv (\exists x)(\neg(x \in A \Rightarrow x \in B)) \equiv (\exists x)(x \in A \wedge x \notin B)$. Ceci signifie qu'il existe au moins un élément de A qui n'est pas un élément de B et confirme la signification intuitive du fait que A n'est pas inclus dans B .



Vous veillerez à ne pas utiliser les symboles logiques comme des abréviations dans vos copies : soit tout est symbolique, soit tout est en français.

Égalité

Définition : égalité

On dit que deux ensembles A et B sont égaux et on note $A = B$ si, et seulement si :

$$(\forall x)(x \in A \Leftrightarrow x \in B)$$

Remarques

- On note le lien entre égalité des ensembles et équivalence des propositions.
- Pour démontrer que deux ensembles sont égaux, on procède souvent par double inclusion (à rapprocher de la démonstration par double implication puisqu'il y a une équivalence dans la définition de l'égalité de deux ensembles).

Intersection

Définition : intersection

On appelle **intersection** des deux ensembles A et B et on note $A \cap B$ (A inter B) l'ensemble défini par :

$$x \in A \cap B \Leftrightarrow (x \in A) \wedge (x \in B)$$

Remarque

Cette fois, on voit le lien entre le symbole \cap des ensembles et le symbole \wedge de la logique des propositions.

Réunion

On a parlé en logique de ET mais quid de OU ? En fait, il existe deux OU : l'*inclusif* et l'*exclusif*. Il est intéressant de savoir lequel est utilisé dans la proposition « fromage OU dessert » : ai-je le droit de prendre les deux (cas *inclusif*) ou non (cas *exclusif*) ?

En logique, on note le OU *inclusif* \vee et le OU *exclusif* \oplus .

Définition : réunion

On appelle réunion des deux ensembles A et B et on note $A \cup B$ (A union B) l'ensemble défini par :

$$x \in A \cup B \Leftrightarrow (x \in A) \vee (x \in B)$$

Remarque

On voit le lien entre le symbole \cup des ensembles et le symbole \vee de la logique des propositions.

Différence

Définition : différence

On appelle différence des deux ensembles A et B et on note $A \setminus B$ (A privé de B) l'ensemble défini par :

$$x \in A \setminus B \Leftrightarrow (x \in A) \wedge (x \notin B)$$

Par exemple, $\mathbb{N} \setminus \{0,1\}$ désigne l'ensemble des entiers naturels supérieurs (ou égaux) à 2.

Définitions par extension / par compréhension

Quand on peut nommer tous les éléments d'un ensemble, on dit qu'on donne la définition en extension de cet ensemble. Par exemple, l'ensemble des nombres entiers naturels pairs inférieurs à 7 est $P_7 = \{0, 2, 4, 6\}$.

Parfois il est beaucoup trop long voire impossible de donner une définition par extension. On peut alors donner une propriété qui caractérise cet ensemble. Par exemple :

$$P = \{p \in \mathbb{Z} \mid (\exists k \in \mathbb{Z})(x = 2 \times k)\}$$

caractérise l'ensemble des entiers pairs. On aurait pu aussi écrire :

$$P = \{2 \times k \mid k \in \mathbb{Z}\}$$

En **Python**, il est souvent commode de définir des ensembles par compréhension.

```
>>> {2*k for k in range(4)}
{0, 2, 4, 6}
```

Ensembles de nombres

Rappelons les notations introduites au lycée :

- \mathbb{N} comme entiers Naturels : 0, 1, 2, 3...
- \mathbb{Z} comme les (Z)entiers (relatifs) : ...-3, -2, -1, 0, 1, 2, 3... Il s'agit de la réunion de \mathbb{N} avec l'ensemble de toutes les solutions des équations $x + n = 0$ avec n un entier naturel.
- \mathbb{D} comme Décimal. Il s'agit des nombres dont l'écriture en base 10 est « finie ». Plus rigoureusement :

$$x \in \mathbb{D} \Leftrightarrow (\exists p \in \mathbb{Z})(\exists k \in \mathbb{Z})(x = k \times 10^p)$$

- \mathbb{Q} comme rationnel. Il s'agit de l'ensemble de toutes les solutions des équations $n \times x = p$ avec n un entier et p un entier naturel non nul :

$$x \in \mathbb{Q} \Leftrightarrow (\exists p \in \mathbb{Z})(\exists n \in \mathbb{N}^*)(n \times x = p)$$

On note l'unique solution de $n \times x = p$ sous la forme $\frac{p}{n}$.

- \mathbb{R} comme Réel. Là les choses se compliquent. Une définition rigoureuse de ce qu'est un nombre réel ne sera au programme que des MPSI/MP2I. On se contentera de ce qu'on vous a dit en classe de Seconde : c'est l'ensemble des abscisses des points d'une droite. Il y a bien sûr beaucoup de choses à dire sur les réels. Nous en reparlerons plus tard.
- \mathbb{C} comme Complexe. $\mathbb{C} = \{a + ib \mid a \in \mathbb{R}, b \in \mathbb{R}\}$ avec i l'unique nombre vérifiant $i^2 = -1$. Les nombres complexes ne sont plus au programme du lycée depuis 2020. Vous les découvrirez en prépa.

Remarques

- $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{D} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$
- On note abusivement A^* l'ensemble $A \setminus \{0\}$ avec A l'un des ensembles ci-dessus.

Intervalle de \mathbb{R}

Vous avez découvert les intervalles en Seconde.

Définition : intervalles de \mathbb{R}

Soit I un ensemble de réels. On dit que I est un intervalle de \mathbb{R} quand :

$$(\forall x \in I)(\forall y \in I)(\forall z \in \mathbb{R})(x \leq z \leq y \Leftrightarrow z \in I)$$



Un intervalle de \mathbb{R} est plus prosaïquement un ensemble de réels SANS TROU.

Par exemple, \mathbb{R}^* n'est pas un intervalle de \mathbb{R} . En effet, prenons $x = -1$, $y = 1$ et $z = 0$: $x \leq z \leq y$ mais $z \notin \mathbb{R}^*$.

Intervalle de \mathbb{Z}

De la même manière, on définit des intervalles entiers avec des doubles crochets (avec ses pendants ouverts).

Définition : intervalles de \mathbb{Z}

$$[[n, p]] = \{k \in \mathbb{Z} \mid n \leq k \leq p\}$$

1.4 Ordre dans \mathbb{R}

— Qu'est-ce que comparer deux nombres réels ?

« *Le Calcul infinitésimal, [...], est l'apprentissage du maniement des inégalités bien plus que des égalités, et on pourrait le résumer en trois mots : MAJORER, MINORER, APPROCHER.* »

Jean DIEUDONNÉ – *Calcul Infinitésimal* (1968)

L'une de vos principales occupations pendant les deux ans à venir sera de suivre cet adage de Jean DIEUDONNÉ, un des grands mathématiciens du XX^e siècle, membre fondateur du groupe BOURBAKI.

Le rappel de résultats pour la plupart utilisés depuis le collège de manière plus ou moins magique va nous permettre de mettre en pratique le principe le plus important du travail en prépa : tout commence par l'étude du cours. **L'approfondissement des démonstrations de cours est la condition *sine qua non* pour dompter les exercices.** Il s'agit dans cette section de revoir des notions normalement connues mais sous le nouvel angle des études post-Bac.

Définissons tout d'abord ce qu'on entend par comparer deux nombres réels.

Définition : comparaison de deux nombres réels

Deux nombres a et b vérifient l'inéquation $a < b$ si, et seulement si, $a - b$ est négatif ou $b - a$ est positif.

— Opérations sur les inégalités

Cette définition va permettre d'écrire le premier théorème de ce cours.

Théorème : addition d'un nombre aux deux membres d'une inégalité

Soit a , b et c trois nombres réels quelconques.

$a < b$ si, et seulement si, $a + c < b + c$.

Ceci signifie qu'ajouter un **même nombre** (positif ou négatif) aux deux membres d'une inégalité ne change pas le sens de cette inégalité.

Il s'agit de revenir à la définition : pour comparer $a + c$ et $b + c$, il faut étudier le signe de leur différence.

$$a < b \Leftrightarrow b - a > 0$$

$$a < b \Leftrightarrow b + c - c - a > 0$$

$$a < b \Leftrightarrow (b + c) - (a + c) > 0$$

$$a < b \Leftrightarrow a + c < b + c \text{ (d'après la définition)}$$



Ruse habituelle : on fait souvent « apparaître » un terme (ici c) dont on a besoin puis on le fait disparaître pour garder l'équilibre.

Théorème : multiplication des deux membres d'une inégalité par un même nombre

Soit trois nombres réels a , b et c tels que $a < b$ et $c \neq 0$.

- Si c est **positif** alors $a \cdot c < b \cdot c$.
- Si c est **négatif** alors $a \cdot c > b \cdot c$.

Autrement dit :

- multiplier les deux membres d'une inégalité par un même nombre **positif ne change pas le sens** de cette inégalité ;
- multiplier les deux membres d'une inégalité par un même nombre **négatif change le sens** de cette inégalité.

Pour la démonstration, on opère de la même façon en étudiant le signe de la différence. On traite le cas $c > 0$, l'autre situation étant similaire.

$$\begin{aligned} a < b &\Leftrightarrow b - a > 0 \\ a < b &\Leftrightarrow c \times (b - a) > 0 \text{ (car } c > 0) \\ a < b &\Leftrightarrow c \times b - c \times a > 0 \\ a < b &\Leftrightarrow c \times a < c \times b \text{ (d'après la définition)} \end{aligned}$$

Remarque

Le cas de la division par un même nombre s'en déduit car diviser par c , c'est multiplier par $1/c$.

Théorème : comparaison de carrés

Deux nombres strictement positifs et leurs carrés sont rangés dans le même ordre :

$$a > 0 \wedge b > 0 \Rightarrow (a < b \Leftrightarrow a^2 < b^2)$$

Deux nombres strictement négatifs et leurs carrés sont rangés dans l'ordre inverse :

$$a < 0 \wedge b < 0 \Rightarrow (a < b \Leftrightarrow a^2 > b^2)$$

Démonstration :

Plusieurs pistes mais on peut remarquer que $b^2 - a^2 = (b - a)(b + a)$.

Ainsi, dans le cas où a et b sont strictement positifs, $b + a$ est strictement positif et on déduit de l'égalité précédente que $b^2 - a^2$ et $b - a$ sont de même signe, d'où la conclusion.

Dans le cas où a et b sont strictement négatifs, $b + a$ est strictement négatif et on déduit de l'égalité précédente que $b^2 - a^2$ et $b - a$ sont de signes contraires, d'où la conclusion.

Attention ! Dans le cas où a et b sont de signes différents, on ne peut pas conclure directement. Par exemple $-1 < 2$ et $(-1)^2 < 2^2$ mais $-2 < 1$ et $(-2)^2 > 1^2$.

Théorème : comparaison d'inverses

Deux nombres non nuls **de même signe** et leurs inverses sont rangés dans l'ordre inverse :

$$a \times b > 0 \Rightarrow (a < b \Leftrightarrow \frac{1}{a} > \frac{1}{b})$$

Démonstration :

$$\frac{1}{b} - \frac{1}{a} = \frac{a - b}{ab}$$



Comme $ab > 0$, on a $\frac{1}{b} - \frac{1}{a}$ du signe de $a - b$ i.e. du signe contraire de $b - a$, d'où la conclusion.

Théorème : addition membre à membre

Soit a, b, c et d quatre nombres réels. Alors :

$$\left. \begin{array}{l} a < b \\ c < d \end{array} \right\} \Rightarrow a + c < b + d$$

Ceci signifie qu'en additionnant membre à membre deux inégalités on n'en change pas le sens.

Il s'agit cette fois de démontrer une implication :

$$\begin{aligned} \left. \begin{array}{l} a < b \\ c < d \end{array} \right\} &\Rightarrow b - a > 0 \wedge d - c > 0 \text{ (d'après la définition)} \\ \left. \begin{array}{l} a < b \\ c < d \end{array} \right\} &\Rightarrow b - a + d - c > 0 \text{ (somme de deux nombres positifs)} \\ \left. \begin{array}{l} a < b \\ c < d \end{array} \right\} &\Rightarrow (b + d) - (a + c) > 0 \\ \left. \begin{array}{l} a < b \\ c < d \end{array} \right\} &\Rightarrow a + c < b + d \text{ (d'après la définition)} \end{aligned}$$

Théorème : transitivité de $<$

Soit a, b et c trois nombres réels. Alors :

$$\left. \begin{array}{l} a < b \\ b < c \end{array} \right\} \Rightarrow a < c$$

Démonstration :

$$\begin{aligned} \left. \begin{array}{l} a < b \\ b < c \end{array} \right\} &\Rightarrow b - a > 0 \wedge c - b > 0 \text{ (définition)} \\ \left. \begin{array}{l} a < b \\ b < c \end{array} \right\} &\Rightarrow b - a + c - b > 0 \text{ (somme de deux nombres positifs)} \\ \left. \begin{array}{l} a < b \\ b < c \end{array} \right\} &\Rightarrow c - a > 0 \\ \left. \begin{array}{l} a < b \\ b < c \end{array} \right\} &\Rightarrow a < c \text{ (définition)} \end{aligned}$$

Théorème : multiplication membre à membre

Soit a, b, c et d quatre nombres réels **POSITIFS**. Alors :

$$\left. \begin{array}{l} a < b \\ c < d \end{array} \right\} \Rightarrow a \cdot c < b \cdot d$$

Ceci signifie qu'en multipliant membre à membre deux inégalités entre nombres POSITIFS on n'en change pas le sens.



Démonstration :

$$\left. \begin{array}{l} a < b \\ c < d \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} a \times c < b \times c \text{ car } c > 0 \\ b \times c < b \times d \text{ car } b > 0 \end{array} \right.$$

$$\left. \begin{array}{l} a < b \\ c < d \end{array} \right\} \Rightarrow a \times c < b \times d \text{ par transitivité}$$

Vous aurez noté l'importance du terme POSITIFS. Essayez avec $-2 < 5$ et $-3 < -1$...

Attention ! Il ne faudra surtout pas diviser deux inégalités membre à membre ! Il faudra toujours revenir à une multiplication.

Par exemple, soit $x \in [-2, 3]$ et $y \in [-3, 1]$. Donnons un encadrement de $\frac{x+4}{y+4}$.

$$\text{D'une part : } -3 < y \leq 1 \Leftrightarrow 1 < y+4 \leq 5 \Leftrightarrow \frac{1}{5} \leq \frac{1}{y+4} < 1.$$

$$\text{D'autre part : } -2 < x \leq 3 \Leftrightarrow 2 < x+4 \leq 7.$$

On peut alors multiplier membre à membre ces inégalités ne comportant que des nombres strictement négatifs :

$$\frac{2}{5} < \frac{x+4}{y+4} < 7$$

Théorème : comparaison à 1

Soit a et b deux nombres réels avec $b \neq 0$.

- Si a et b sont POSITIFS, $a < b \Leftrightarrow \frac{a}{b} < 1$.
- Si a et b sont NÉGATIFS, $a < b \Leftrightarrow \frac{a}{b} > 1$.

Il suffit de multiplier membre à membre par $1/b$.

— Ordre et fonctions

Nous rappelons une définition importante du cours de Seconde.

Définition : sens de variation d'une fonction

Une fonction f est dite **croissante** sur un intervalle I si, et seulement si :

$$(\forall x \in I)(\forall y \in I)(x < y \Rightarrow f(x) \leq f(y))$$

Elle est dite **décroissante** sur un intervalle I si, et seulement si :

$$(\forall x \in I)(\forall y \in I)(x < y \Rightarrow f(x) \geq f(y))$$

Autrement dit : une fonction est croissante sur I si, et seulement si, elle conserve l'ordre des éléments de I , et une fonction est décroissante sur I si, et seulement si, elle inverse l'ordre des éléments de I .

On va donc pouvoir composer des inégalités sur des nombres d'un intervalle I par des fonctions monotones sur I (i.e. de sens de variation constant sur I).

1.5 Exercices

Test 1.1 (Condition nécessaire/suffisante - ♥)

Deux nombres de carrés égaux sont-ils égaux ?

Test 1.2 (♥)

On considère les propositions atomiques B, T, V, C représentant les assertions suivantes : « Hélène est belle », « Hélène aime le thé vert », « Hélène porte une robe violette », « Chri-Chri aime Hélène ». Énoncez des phrases simples traduisant les propositions suivantes :

1. (a) $\neg B$ (d) $(B \wedge V) \Rightarrow C$
 (b) $\neg B \wedge \neg T$ (e) $C \Rightarrow (B \Leftrightarrow V)$
 (c) $\neg(B \vee T)$ (f) $((B \vee T) \wedge \neg V) \Rightarrow \neg C$

2. Traduisez par une proposition simple les phrases

- (a) « Chri-Chri aime Hélène seulement si elle porte une robe violette ».
- (b) « Chri-Chri aime Hélène si elle porte une robe violette ».
- (c) « Chri-Chri aime Hélène si, et seulement si, elle porte une robe violette ».
- (d) « Si Chri-Chri aime Hélène alors elle porte une robe violette ».
- (e) « Si Hélène porte une robe violette alors Chri-Chri l'aime ».
- (f) « Il est suffisant qu'Hélène porte une robe violette pour que Chri-Chri l'aime ».
- (g) « Il est nécessaire qu'Hélène porte une robe violette pour que Chri-Chri l'aime ».

Test 1.3 (♥)

Réécrivez les phrases suivantes pour qu'elles soient équivalentes au sens de la logique des propositions en utilisant « suffisant » ou « nécessaire » :

1. Si ses poules ne se lavent pas les dents, il ne vient pas en cours.
2. Si je range ma chambre, Hélène ne m'aimera pas.

Test 1.4 (♥)

Déterminez les contraposées, les réciproques puis les négations des propositions suivantes (vous énoncerez les propositions dans le même langage courant) :

1. Si Max n'étudie pas son cours de maths, les filles le fuient.
2. Héloïse marquera un but seulement si elle lit la vie de Carl GAUSS en latin.

Test 1.5 (♥)

1. Démontrez, à l'aide de tables de vérité :

- (a) la première loi de DE MORGAN de la logique des propositions : $\neg(p \wedge q) \equiv \neg p \vee \neg q$;
- (b) la distributivité de la disjonction \vee sur la conjonction \wedge .

2. Démontrer les mêmes propriétés pour les lois ensemblistes :

- (a) la première loi de DE MORGAN : $\overline{A \cup B} = \overline{A} \cap \overline{B}$;
- (b) la distributivité de la réunion \cup sur l'intersection \cap .

Test 1.6 (♥)

Déterminez une table de vérité du connecteur IFTE défini sur \mathcal{B}_2^3 par $\text{IFTE}(x,y,z) = 1$ si, et seulement si, « SI x est vrai ALORS y est vrai SINON z est vrai » est valide.

Test 1.7 (♥)

L'implication est-elle associative ?

Test 1.8 (♥)

Que signifie : « Il ne m'arrive jamais de n'avoir à me plaindre de rien » ?

Test 1.9 (♥)

Montrez que le produit d'un irrationnel par un rationnel non nul est toujours un irrationnel. Avez-vous raisonné par l'absurde ou par contraposée ?

Test 1.10 (♥♥)

Formulez les phrases suivantes en utilisant uniquement des symboles mathématiques, notamment ceux de la logique des prédicats :

1. L'entier n est le carré d'un entier.
2. Le cercle unité possède un point à coordonnées rationnelles.
3. Les ensembles A et B ne sont pas disjoints.
4. Les ensembles A et B sont distincts.
5. Les éléments de A sont des éléments de B .
6. Tout réel positif est un carré de réel.
7. L'équation $f(x) = 0$ ne possède pas de solution entière.
8. La fonction f s'annule sur tout intervalle réel de longueur 3.
9. Tout intervalle ouvert de \mathbb{R} contient un rationnel.
10. La fonction f ne s'annule pas sur $[a,b]$.
11. Tout intervalle fermé de \mathbb{R} d'amplitude 1 contient un nombre entier.

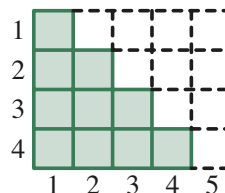
Test 1.11 (♥)

Écrivez par compréhension :

1. $\{\dots, -64, -27, -8, -1, 0, 1, 8, 27, 64, \dots\}$
2. $\{1, 5, 9, 13, 17, 21, 25, \dots\}$

Test 1.12 (Carrés récurifs - ♥)

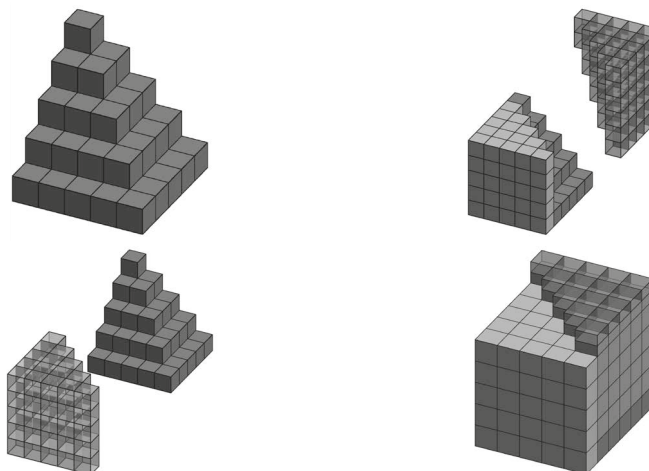
Voici un test de fin d'étude maternelle en Syldavie : prenez un cube, placez en dessous deux autres cubes, et encore en dessous trois cubes, etc.



Combien y a-t-il de cubes verts au total sur le dessin ci-dessus ? On peut encore les compter à la main, mais que faire si je vous demande le nombre de cubes lorsqu'on a placé 100 rangées ? n rangées ?

Test 1.13 (Cubes récurifs - ♥)

Que vous inspirent les petits dessins suivants :



Test 1.14 (Test de primalité - ♥)

Que pensez-vous de la proposition suivante ? :

« Les images des entiers naturels par la fonction $p : n \mapsto n^2 - n + 41$ sont des nombres premiers. »

Test 1.15 (Tout le monde a le même âge - ♥)

On se propose de démontrer que tous les étudiants en prépa ont le même âge et, pour cela, on note $P(n)$ l'affirmation :

« Si on choisit n étudiants en prépa ($n \in \mathbb{N}^*$), il est sûr qu'ils ont tous le même âge. »

Il est clair que $P(1)$ est vraie.

Démontrons que $P(n) \Rightarrow P(n+1)$. Pour cela nous supposons que $P(n)$ est vraie (c'est l'hypothèse de récurrence) et nous choisissons un groupe quelconque de $n+1$ étudiants que nous ordonnons par ordre alphabétique (pourquoi pas ?). D'après l'hypothèse de récurrence, les n premiers de l'ordre alphabétique ont tous le même âge ainsi que les n derniers. Comme ces deux groupes de n étudiants ont au moins un étudiant en commun, on en déduit qu'ils ont tous le même âge.

Nous venons de démontrer que $P(n) \Rightarrow P(n+1)$ pour tout $n \geq 1$ et, comme $P(1)$ est vrai, $P(n)$ est toujours vrai pour tout $n \geq 1$. **Y a-t-il une erreur dans le raisonnement ?**

Test 1.16 (♥)

1. Voici un extrait du rapport d'une épreuve de concours en 2020 : *Un nombre très important de candidats pense que si $a \geq 4$ et $b \geq 4$ alors $\frac{a}{b} \geq \frac{4}{4} = 1$.*

Qu'en pensez-vous ?

2. Sachant que $x \geq 1$, démontrez que $y = \frac{3x+1}{x+3}$ est aussi supérieur à 1.

3. Sachant que $x \geq 1$, déterminez le signe de $y - x$. On lit à ce sujet sur le rapport de jury : *La gestion parfois défailante des règles de calculs algébriques entraîne que seuls 40 % des candidats fournissent une réponse correcte.*

4. Résolvez dans $[1, +\infty[$ l'équation $L = \frac{3L+1}{L+3}$. Toujours dans ce rapport de jury : *Il est surprenant de constater que la résolution de l'équation $L = \frac{3L+1}{L+3}$ pose problème à un nombre très important de candidats.*

Test 1.17 (♥♥)

Démontrez que toute fonction strictement croissante de \mathbb{N} dans \mathbb{N} vérifie :

$$(\forall n \in \mathbb{N})(f(n) \geq n)$$

Test 1.18 (♥♥)

Démontrez par récurrence que $(\forall x \in \mathbb{R}_+)(\forall n \in \mathbb{N})((1+x)^n \geq 1+nx)$ (inégalité de BERNOULLI).

Coups de pouce

Coups de pouce pour le test 1.1

Notons C la proposition « les deux nombres ont leurs carrés égaux » et E « les deux nombres sont égaux ».

Il s'agit donc d'étudier la proposition $C \Rightarrow E$.

Coups de pouce pour le test 1.2

Le langage courant peut paraître ambigu... Vous pouvez tenter de vous replacer dans un contexte sûr avec des propositions du type « Q est un quadrilatère », « Q est un carré », « Q est un parallélogramme », « Q est un rectangle », « Q est un losange »... si vous connaissez bien votre cours de collège ;-)

Coup de pouce pour le test 1.3

$A \Rightarrow B$ peut se lire « Il est suffisant que A soit réalisé pour que B le soit » ou, en passant à la contraposée « Il est nécessaire que B soit réalisé pour que A le soit » (si on n'a pas B alors on n'a pas A).

Coup de pouce pour le test 1.4

Déterminez A et B pour que les propositions soient de la forme $A \Rightarrow B$ puis appliquer les définitions du cours.

Coup de pouce pour le test 1.5

1. Il s'agit de démontrer $\neg(p \wedge q) \equiv \neg p \vee \neg q$ et $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ en dressant les tables de chaque membre de ces équivalences de formules.

2. Utilisez les résultats de la question 1. et se souvenir que $A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$.

Coups de pouce pour le test 1.7

Est-ce que $(p \Rightarrow q) \Rightarrow r \equiv p \Rightarrow (q \Rightarrow r)$ est une tautologie ? On peut dresser une table de vérité ou bien être plus malin en essayant de déterminer un cas qui rende la proposition fausse.

Coups de pouce pour le test 1.8

L'emploi des doubles négations est un classique des entretiens d'embauches : il faut vous en débarrasser en vous souvenant que $\neg(\neg p) \equiv p$.

Coups de pouce pour le test 1.9

Un nombre r est rationnel si, et seulement si, il existe un couple d'entiers $(p, q) \in \mathbb{Z} \times \mathbb{N}^*$ tel que $r = \frac{p}{q}$.

Il s'agit de vérifier la validité de :

$$(x \in \mathbb{Q} \wedge y \notin \mathbb{Q}) \Rightarrow xy \notin \mathbb{Q}$$

Coup de pouce pour le test 1.10

Il s'agit d'utiliser les quantificateurs \exists et \forall . Il y a plusieurs possibilités d'écriture la plupart du temps.

Rappel : le cercle unité est l'ensemble des points à une distance 1 de l'origine du repère.

Pouce de pouce pour le test 1.11

Essayez de reconnaître un motif logique comme dans les tests de QI...

Coups de pouce pour le test 1.12

Le dessin nous donne une idée : si nous complétons la figure pour former un rectangle, il y a deux fois plus de cubes, mais maintenant nous pouvons les compter. Il y en a

$$\text{en effet } \frac{4 \times (4+1)}{2}, \text{ et donc :}$$

$$1 + 2 + 3 + 4 = \frac{4 \times (4+1)}{2}$$

Coups de pouce pour le test 1.13

On doit démontrer par récurrence que $1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(n+1/2)}{3}$.

Coups de pouce pour le test 1.14

Trop beau pour être vrai, pourtant cela part bien : vrai pour 1, 2, 3, 4,...

Coups de pouce pour le test 1.15

Faites bien attention à l'initialisation.

Coup de pouce pour le test 1.16

Revoyez le cours sur la multiplication membre à membre. Rappelez-vous en particulier qu'on ne peut diviser membre à membre deux inégalités : il faut revenir à des multiplications par passage à l'inverse. Mais, pour cela, il faut avoir des nombres de même signe.

Coup de pouce pour le test 1.17

Raisonnez par récurrence sur n .

Coup de pouce pour le test 1.18

Pour l'hérédité, pensez que $(1+x)^{n+1} = (1+x) \times (1+x)^n$.

Solutions et commentaires

Solution du test 1.1

Notons tout d'abord que la proposition $E \Rightarrow C$ est bien sûr vraie : si deux nombres sont égaux, leurs carrés le sont aussi (cette proposition reste vraie si les deux nombres ne sont pas égaux).

La contraposée $\neg C \Rightarrow \neg E$ nous permet alors de dire que si deux nombres n'ont pas le même carré, ils ne sont pas égaux.

Cependant, $(3)^2 = (-3)^2$ mais $3 \neq -3$. Ainsi on peut avoir C vraie mais E fausse : la proposition $C \Rightarrow E$ est donc fausse.

On remarque que cette formulation n'est pas satisfaisante car la proposition est parfois vraie (pour des nombres positifs par exemple) : il faudrait pouvoir être plus précis. Veut-on savoir si la proposition est vraie **pour tout** couple de nombres ou bien *s'il existe* un couple qui la rend fausse ?

Il faut alors passer de la **logique des propositions** à la **logique des prédicats** qui étudie les propositions dont la valeur de vérité dépend de certaines variables.

Vous étudierez ceci plus en détail l'an prochain et travaillerez notamment avec les **quantificateurs** \forall et \exists .

Solution du test 1.2

1. (a) « Hélène n'est pas belle ».
- (b) « Hélène n'est pas belle et n'aime pas le thé vert ».
- (c) Difficile à exprimer en français si ce n'est sous la forme précédente car les deux propositions sont équivalentes d'après les lois de DE MORGAN.
- (d) « Si Hélène est belle et porte une robe violette alors Chri-Chri l'aime ».
- (e) « Si Chri-Chri aime Hélène, alors celle-ci est belle si, et seulement si, elle porte une robe violette ».
- (f) « Si Hélène ne porte pas de robe violette et qu'elle est belle ou aime le thé vert, alors Chri-Chri ne l'aime pas ».

2. (a) $C \Rightarrow V$

(b) $V \Rightarrow C$

(c) $C \Leftrightarrow V$

(d) $C \Rightarrow V$

(e) $V \Rightarrow C$

(f) $V \Rightarrow C$

(g) $C \Rightarrow V$

Solution du test 1.3

1. « Il est suffisant que les poules ne se lavent pas les dents pour qu'il ne vienne pas en cours » ou « Il est nécessaire qu'il ne vienne pas en cours pour que les poules ne se lavent pas les dents ».

2. « Il est suffisant que je range ma chambre pour qu'Hélène ne m'aime pas » ou « Il est nécessaire qu'Hélène ne m'aime pas pour que je range ma chambre ».

Solution du test 1.4

1. Appelons M la proposition : « Max n'étudie pas son cours de maths » et F la proposition « Les filles le fuient ». La proposition originale est donc $M \Rightarrow F$.

La contraposée est donc $\neg F \Rightarrow \neg M$: « Si les filles ne fuient pas Max alors il étudie son cours de maths ».

La réciproque est $F \Rightarrow M$: « Si les filles fuient Max alors il n'étudie pas son cours de maths ».

La négation est $M \wedge \neg F$: « Max n'étudie pas son cours de maths et les filles ne le fuient pas ».

2. Appelons H la proposition « Héloïse marquera un but » et G la proposition « Elle lit la vie de Carl GAUSS en latin ».
La proposition originale est donc $H \Rightarrow G$.

La contraposée est donc $\neg G \Rightarrow \neg H$: « Si Héloïse ne lit pas la vie de Carl GAUSS en latin alors elle ne marquera pas de but ».

La réciproque est $G \Rightarrow H$: « Si Héloïse lit la vie de Carl GAUSS en latin alors elle marquera un but ».

La négation est $H \wedge \neg G$: « Héloïse marque un but et n'a pas lu la vie de Carl GAUSS en latin ».

Solution du test 1.5

1. On vérifie que les colonnes correspondant à $\neg(p \wedge q)$ et à $\neg p \vee \neg q$ sont identiques :

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

On fait de même avec $p \vee (q \wedge r)$ et $(p \vee q) \wedge (p \vee r)$

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

$$\begin{aligned}
 2. (a) \quad x \in \overline{A \cap B} &\Leftrightarrow \neg(x \in A \cap B) \\
 x \in \overline{A \cap B} &\Leftrightarrow \neg((x \in A) \wedge (x \in B)) \\
 x \in \overline{A \cap B} &\Leftrightarrow (x \notin A) \vee (x \notin B) \quad (\text{loi de DE MORGAN}) \\
 x \in \overline{A \cap B} &\Leftrightarrow (x \in \bar{A}) \vee (x \in \bar{B}) \\
 x \in \overline{A \cap B} &\Leftrightarrow x \in \bar{A} \cup \bar{B}
 \end{aligned}$$

On en déduit que $\overline{A \cap B} = \bar{A} \cup \bar{B}$.

$$\begin{aligned}
 (b) \quad x \in A \cup (B \cap C) &\Leftrightarrow (x \in A) \vee ((x \in B) \wedge (x \in C)) \\
 x \in A \cup (B \cap C) &\Leftrightarrow ((x \in A) \vee (x \in B)) \wedge ((x \in A) \vee (x \in C)) \quad (\text{distributivité de la disjonction sur la conjonction}) \\
 x \in A \cup (B \cap C) &\Leftrightarrow (x \in A \cup B) \wedge (x \in A \cup C) \\
 x \in A \cup (B \cap C) &\Leftrightarrow x \in (A \cup B) \cap (A \cup C)
 \end{aligned}$$

On en déduit que $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

Solution du test 1.6

x	y	z	$\text{IFTE}(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Solution du test 1.7

On peut par exemple considérer l'environnement où les trois propositions atomiques sont fausses.

Alors $\text{Val}((p \Rightarrow q) \Rightarrow r, \langle v(p) = v(q) = v(r) = 0 \rangle) = 0$ mais $\text{Val}(p \Rightarrow (q \Rightarrow r), \langle v(p) = v(q) = v(r) = 0 \rangle) = 1$.

L'implication n'est donc pas associative : on ne peut donc pas se passer de parenthèses.

Solution du test 1.8

« In summary, I would like to convey a little positive message to you...
I do not have any... Would two negative messages suit you? »

Woody ALLEN

La négation de « il ne m'arrive jamais » est « il peut m'arriver » et la négation de « n'avoir à me plaindre de rien » est « avoir à me plaindre d'au moins une chose ».

La phrase a donc la même signification que « Il m'arrive parfois de me plaindre ».

Solution du test 1.9

Raisonnons par ...

Soit x un nombre irrationnel et y un nombre rationnel. Il existe donc un couple d'entiers $(p, q) \in \mathbb{Z} \times \mathbb{N}^*$ tel que $y = \frac{p}{q}$.

Supposons de plus que $x \cdot y$ soit rationnel. Alors existe un couple d'entiers $(a, b) \in \mathbb{Z} \times \mathbb{N}^*$ tel que $x \cdot y = x \cdot \frac{p}{q} = \frac{a}{b}$.

On en déduit que $x = \frac{q \cdot a}{p \cdot b}$: le nombre irrationnel $x...$ est donc rationnel. Nous obtenons une contradiction.

L'hypothèse supplémentaire « $x \cdot y$ est rationnel » est donc fausse.

Nous avons donc démontré par l'absurde que le produit d'un rationnel par un irrationnel est irrationnel.

Solution du test 1.10

1. $(\exists k \in \mathbb{Z})(n = k^2)$
2. $(\exists x \in \mathbb{Q})(\exists y \in \mathbb{Q})(x^2 + y^2 = 1)$
3. $(\exists x)(x \in A \wedge x \in B)$ ou $A \cap B \neq \emptyset$
4. $(\exists x)((x \in A) \wedge (x \notin B)) \vee ((x \notin A) \wedge (x \in B))$
5. $A \subseteq B$ ou $(\forall x)(x \in A \Rightarrow x \in B)$
6. $(\forall x \in \mathbb{R}_+)(\exists y \in \mathbb{R})(x = y^2)$
7. $\{x \mid (x \in \mathbb{Z}) \wedge (f(x) = 0)\} = \emptyset$ ou $\{x \mid f(x) = 0\} \cap \mathbb{Z} = \emptyset$
8. $(\forall a \in \mathbb{R})(\exists x \in [a, a + 3])(f(x) = 0)$
9. $(\forall x \in \mathbb{R})(\forall y \in \mathbb{R})(x < y \Rightarrow]x, y[\cap \mathbb{Q} \neq \emptyset)$
10. $\{x \mid x \in [a, b] \wedge f(x) = 0\} = \emptyset$
11. $(\forall x \in \mathbb{R})(\exists n \in \mathbb{N})(n \in [x, x + 1])$ ou $(\forall x \in \mathbb{R})([x, x + 1] \cap \mathbb{N} \neq \emptyset)$

Solution du test 1.11

1. On reconnaît les cubes des premiers entiers : $E_1 = \{x^3 \mid x \in \mathbb{Z}\}$
2. On passe d'un terme à l'autre en ajoutant 4 : $E_2 = \{4k+1 \mid k \in \mathbb{N}\}$

Solution du test 1.12

1. Nous allons essayer de démontrer que la propriété suivante est vraie pour tout entier naturel non nul n :

$$\mathcal{P}(n) : \ll 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \gg$$

2. Il est facile de vérifier que $1 = \frac{1(1+1)}{2}$, donc la deuxième étape de notre raisonnement est vérifiée :

$\mathcal{P}(1)$ est vraie

3. Supposons qu'une « génération », appelons-la par exemple la k^e , soit « infectée ».

Plus sobrement on dira : soit k un entier supérieur à 1. Supposons que $\mathcal{P}(k)$ soit vraie et essayons alors de montrer que cela implique que la génération suivante, la $k+1^e$, sera elle aussi infectée, c'est-à-dire :

$$\mathcal{P}(k) \Rightarrow \mathcal{P}(k+1)$$

Il s'agit donc de calculer $1 + 2 + 3 + \cdots + k + (k+1)$ sachant que $1 + 2 + 3 + \cdots + k = \frac{k(k+1)}{2}$, or :

$$\begin{aligned} 1 + 2 + 3 + \cdots + k + (k+1) &= \underbrace{1 + 2 + 3 + \cdots + k}_{\frac{k(k+1)}{2}} + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= (k+1) \left(\frac{k}{2} + 1 \right) \\ &= (k+1) \left(\frac{k+2}{2} \right) \\ &= \frac{(k+1)(k+1+1)}{2} \end{aligned}$$

Nous en déduisons que $\mathcal{P}(k) \Rightarrow \mathcal{P}(k+1)$.

4. Nous avons vérifié que la proposition était vraie au rang 1 et qu'elle était héréditaire. La proposition est donc vraie pour tout entier $n \geq 1$.

Solution du test 1.13

Soit P_n la proposition : $\ll S_n = \sum_{k=1}^n k^2 = \frac{n(n+1)(n+1/2)}{3} = \frac{n(n+1)(2n+1)}{6} \gg$

P_1 est vraie.

Soit k un entier naturel non nul. Démontrons que $P_k \Rightarrow P_{k+1}$:

$$\begin{aligned} S_{k+1} &= S_k + (k+1)^2 \\ &= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \\ &= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\ &= \frac{(k+1)(k(2k+1) + 6k+6)}{6} \\ &= \frac{(k+1)(2k^2 + 7k+6)}{6} \\ &= \frac{(k+1)(k+2)(2(k+1)+1)}{6} \end{aligned}$$

La propriété est bien héréditaire et vraie pour $n=1$: elle est donc vraie pour tout entier naturel non nul.

Solution du test 1.14

La fonction `isprime` de **Python** teste si un entier est premier.

La commande suivante renvoie le résultat du test de primalité appliqué aux images des quarante premiers entiers naturels par la fonction p :

```
In [12]: [sy.isprime(n**2 - n + 41) for n in range(41)]
```

Quarante et une réponses `True` ! Cela semble donc rudement vrai...et pourtant $p(41) = 41^2$ qui n'est pas vraiment premier donc prudence avec l'induction toute simple : elle a besoin d'être certifiée conforme au label « induction mathématique »...

Solution du test 1.15

Il faut bien sûr commencer l'initialisation à $n = 2$ et alors la propriété n'est pas vraie : rien n'assure que deux étudiants pris au hasard ont le même âge. Pourtant, la proposition est bien héréditaire mais cela ne suffit pas !

Solution du test 1.16

1. Voyez la remarque du cours. On peut prendre $a = 5 > 4$ et $b = 10 > 4$ mais $\frac{a}{b} = \frac{1}{2} < 1$.

2. On remarque que $\frac{3x+1}{x+3} - 1 = \frac{2x-2}{x+3}$. Donc $x \geq 1 \Rightarrow 2x-2 \geq 0 \wedge x+3 > 0 \Rightarrow \frac{2x-2}{x+3} \geq 0 \Rightarrow \frac{3x+1}{x+3} - 1 \geq 0 \Rightarrow \frac{3x+1}{x+3} \geq 1$.

3. On obtient $y - x = \frac{3x+1}{x+3} - \frac{x^2+3x}{x+3} = \frac{1-x^2}{x+3} = \frac{(1-x)(1+x)}{x+3}$.

Donc si $x \geq 1$ alors $1-x \leq 0$, $1+x \geq 0$ et $x+3 \geq 0$. On en déduit que $x \geq 1 \Rightarrow y - x \leq 0$.

4. Sur $[1, +\infty[$, $L = \frac{3L+1}{L+3} \Leftrightarrow L^2 - 1 = 0 \Leftrightarrow L = 1$.

Solution du test 1.17

Soit $\mathcal{P}(n)$ la proposition « $(\forall n \in \mathbb{N})(f(n) \geq n)$ ».

- Comme f est à valeur dans \mathbb{N} , $f(n) \geq 0$ pour tout entier naturel n donc en particulier pour 0 : $f(0) \geq 0$, ce qui assure que $\mathcal{P}(0)$ est vraie.
- Comme f est strictement croissante sur \mathbb{N} , on a, pour tout entier naturel n , $f(n+1) > f(n)$. On en déduit que :

$$(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow f(n+1) > f(n) \geq n)$$

Ainsi $f(n+1) > n$. Or le plus petit entier strictement supérieur à n est le successeur de n , à savoir $n+1$. On en déduit que :

$$(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow f(n+1) > f(n) \geq n)$$

Finalement, on a démontré que :

$$(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1))$$

- On en déduit que, d'après le principe de récurrence simple :

$$(\forall n \in \mathbb{N})(f(n) \geq n)$$

Solution du test 1.18

Soit $\mathcal{P}(n)$ la proposition « $(\forall x \in \mathbb{R}_+)(\forall n \in \mathbb{N})((1+x)^n \geq 1+nx)$ ».

- $(\forall x > 0)((1+x)^0 = 1 \geq 1+0 \times x$ donc $\mathcal{P}(0)$ est vraie.
- $(\forall x > 0)(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow (1+x)^{n+1} = (1+x)(1+x)^n \geq (1+x)(1+nx) = 1+x+nx+nx^2 = 1+(n+1)x+nx^2 \geq 1+(n+1)x)$
car $(\forall x > 0)(\forall n \in \mathbb{N})(nx^2 \geq 0)$. Ainsi $(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1))$.
- On en déduit, d'après le principe de récurrence simple, l'inégalité de BERNOULLI.

2.1 De l'importance de savoir calculer

On dispose certes d'ordinateurs pour effectuer les calculs (et nous verrons comment le faire) mais, auparavant, méditez cette pensée d'Alain CONNES, membre de l'Académie des sciences, professeur au Collège de France, à l'IHES et à l'université Vanderbilt aux États-Unis. Il a de plus reçu la médaille Fields en 1982, le prix Crafoord en 2001 et la Médaille d'or du CNRS en 2004.

« Quand on effectue un long calcul algébrique, la durée nécessaire est souvent très propice à l'élaboration dans le cerveau de la représentation mentale des concepts utilisés. C'est pourquoi l'ordinateur, qui donne le résultat d'un tel calcul en supprimant la durée, n'est pas nécessairement un progrès. On croit gagner du temps, mais le résultat brut d'un calcul sans la représentation mentale de sa signification n'est pas un progrès. »

Alain CONNES – Sciences et imaginaire

2.2 Calcul de sommes et de produits

Sommes

Définition : symbole Σ

Soit $n \in \mathbb{N}^*$ et u_1, \dots, u_n une famille de nombres. On note :

$$u_1 + u_2 + \dots + u_n = \sum_{i=1}^n u_i$$

Plus généralement, pour tous $p \in \mathbb{N}$ et $n \in \mathbb{N}$ tels que $p \leq n$, on note :

$$u_p + u_{p+1} + \dots + u_n = \sum_{i=p}^n u_i = \sum_{i \in \llbracket p, n \rrbracket} u_i = \sum_{p \leq i \leq n} u_i$$

- i s'appelle l'**indice** de la somme ;
- p et n sont les **bornes** de la somme.



Nombre de termes. La somme $\sum_{p \leq i \leq n} u_i$ contient $n - p + 1$ termes !

Remarques

- L'indice d'une somme est dit muet, c'est-à-dire que toute lettre non utilisée convient et on peut écrire : $\sum_{i=p}^n u_i = \sum_{j=p}^n u_j$.
- Il existe une sorte de **relation de CHASLES** pour les sommes :

$$\sum_{p \leq i \leq n} u_i = \sum_{p \leq i \leq r} u_i + \sum_{r+1 \leq i \leq n} u_i$$



Somme des 1. $\sum_{k=0}^n k = n+1$

— Sommes et puissances entières

Théorème : sommes de puissances d'entiers successifs

Soit n un entier naturel quelconque, alors :

$$\begin{aligned} - \sum_{0 \leq k \leq n} k &= \frac{n(n+1)}{2} \\ - \sum_{0 \leq k \leq n} k^2 &= \frac{n(n+1)(2n+1)}{6} \\ - \sum_{0 \leq k \leq n} k^3 &= \left(\frac{n(n+1)}{2} \right)^2 \end{aligned}$$

Les démonstrations se font par récurrence et ont été vues en exercice dans le chapitre 1.

Théorème : somme des puissances successives d'un nombre quelconque

Soit $x \in \mathbb{R}$ et $n \in \mathbb{N}$, alors :

$$\sum_{0 \leq k \leq n} x^k = \begin{cases} \frac{x^{n+1} - 1}{x - 1} & \text{si } x \neq 1 \\ n + 1 & \text{sinon} \end{cases}$$

Il existe de nombreuses démonstrations de ce théorème, dont une légendaire due au petit GAUSS quand il avait 10 ans.

Nous nous contenterons d'enfoncer le clou de la récurrence. Le cas $x = 1$ étant évident, nous considérerons à présent que x est un réel différent de 1.

Démontrons par récurrence simple la proposition $\mathcal{P}(n)$ définie pour tout entier naturel n par

$$\sum_{0 \leq k \leq n} x^k = \frac{x^{n+1} - 1}{x - 1} :$$

$$\bullet \sum_{k=0}^0 x^k = x^0 = 1 = \frac{1-x}{1-x} = \frac{1-x^{0+1}}{1-x}$$

$$\bullet \mathcal{P}(n) \Rightarrow \sum_{k=0}^{n+1} x^k = \sum_{k=0}^n x^k + x^{n+1} = \frac{x^{n+1} - 1}{x - 1} + x^{n+1} = \frac{x^{n+1} - 1 + x^{n+2} - x^{n+1}}{x - 1} = \frac{x^{n+1+1} - 1}{x - 1}$$

Ceci démontre que $(\forall n \in \mathbb{N})(\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1))$.

D'après le principe de récurrence simple, on en déduit que la proposition est vraie sur \mathbb{N} .

Linéarité de la somme

La notion de linéarité va être très importante tout au long de vos années de prépa. Vous l'avez rencontrée depuis votre tendre enfance de manière informelle. Il est temps de lui donner une définition.

Définition : combinaison linéaire

La linéarité désigne un rapport de dépendance entre des variables. On dit qu'une variable y dépend linéairement d'une famille de variables $(x_i)_{0 \leq i \leq n}$ (ou est une combinaison linéaire ou dépend linéairement) si, et seulement si, il existe une famille $(\lambda_i)_{0 \leq i \leq n}$ de scalaires (i.e. des éléments constants de \mathbb{R}) telle que :

$$y = \lambda_0 x_0 + \lambda_1 x_1 + \dots + \lambda_n x_n$$

Une fonction sera alors dite linéaire si elle conserve les combinaisons linéaires, i.e. l'image d'une combinaison linéaire est la combinaison linéaire des images.

Pour la somme cela donne :

Théorème : linéarité de la somme

Soit n et p des entiers naturels avec $p \leq n$ et x_p, x_{p+1}, \dots, x_n et y_p, y_{p+1}, \dots, y_n des nombres. Alors :

$$(\forall \lambda \in \mathbb{R}) \left(\sum_{k=p}^n (x_k + \lambda y_k) = \sum_{k=p}^n x_k + \lambda \sum_{k=p}^n y_k \right)$$

Ici, $x_k + \lambda y_k$ est une combinaison linéaire des x_k et y_k . La somme de la combinaison est la combinaison des sommes.

Vous pouvez démontrer ce théorème à titre d'exercice, par récurrence sur n par exemple.

Exemple

$$\sum_{k=0}^n k(k-2) = \sum_{k=0}^n k^2 - 2 \sum_{k=0}^n k = \frac{n(n+1)(2n+1)}{6} - 2 \frac{n(n+1)}{2} = \frac{n(n+1)(2n+1-6)}{6}$$

Changement d'indice (réindexation)

On veut par exemple calculer $S_n = \sum_{k=1}^n (k-1)^2$. On peut développer $(k-1)^2$ et utiliser la linéarité de la somme.

On peut sinon remarquer que S_n ressemble à la somme des premiers carrés d'entiers successifs. Il suffit de poser $i = k-1$, ce qui équivaut à $k = i+1$. On a $i_{\max} = k_{\max} - 1 = n-1$ et $i_{\min} = k_{\min} - 1 = 1-1 = 0$.

$$\text{Alors } S_n = \sum_{i=0}^{n-1} i^2 = \frac{(n-1)n(2n-1)}{6}.$$

Sommes télescopiques

Théorème : somme télescopique

Soit $(u_i)_{i \in \mathbb{N}}$ une suite de nombres. Alors, pour tout entier naturel n :

$$\sum_{i=0}^n (u_{i+1} - u_i) = u_{n+1} - u_0 \quad (\text{différence des extrêmes})$$

Pour démontrer ce résultat, on peut par exemple utiliser la linéarité de la somme et effectuer une réindexation.

$$S = \sum_{i=0}^n (u_{i+1} - u_i) = \sum_{i=0}^n u_{i+1} - \sum_{i=0}^n u_i$$

Effectuons le changement d'indice $j = i + 1$, $j_{\max} = n + 1$, $j_{\min} = 0 + 1 = 1$.

$$S = \sum_{j=1}^{n+1} u_j - \sum_{i=0}^n u_i = \left(u_{n+1} + \sum_{j=1}^n u_j \right) - \left(\sum_{i=1}^n u_i + u_0 \right) = u_{n+1} + \left(\sum_{j=1}^n u_j - \sum_{i=1}^n u_i \right) - u_0 = u_{n+1} - u_0$$

car les indices i et j sont muets.

— Sommes doubles

Théorème : somme double

Soit a_p, \dots, a_n et b_q, \dots, b_m des éléments de \mathbb{R} . Alors :

$$\sum_{i=p}^n \left(\sum_{j=q}^m u_{ij} \right) = \sum_{j=q}^m \left(\sum_{i=p}^n u_{ij} \right)$$

Exemple

En utilisant la linéarité de la somme :

$$\sum_{i=0}^n \sum_{j=0}^n i \times j = \sum_{i=0}^n \left(\sum_{j=0}^n i \times j \right) = \sum_{i=0}^n i \times \left(\sum_{j=0}^n j \right) = \sum_{i=0}^n i \times \frac{n(n+1)}{2} = \frac{n(n+1)}{2} \sum_{i=0}^n i = \left(\frac{n(n+1)}{2} \right)^2$$

— Produits

Définition : symbole \prod

Soit n un entier naturel et $(x_i)_{i \in \llbracket 0, n \rrbracket}$ une famille d'éléments de \mathbb{R} . On note :

$$x_0 \times x_1 \times \dots \times x_n = \prod_{i \in \llbracket 0, n \rrbracket} x_i$$

Remarque

Il existe une relation de CHASLES pour les produits :

$$\prod_{p \leq i \leq n} u_i = \prod_{p \leq i \leq r} u_i \times \prod_{r+1 \leq i \leq n} u_i$$

Définition : factorielle

Soit n un entier naturel. On appelle factorielle la quantité :

$$n! = 1 \times 2 \times 3 \times \dots \times n = \prod_{i \in \llbracket 1, n \rrbracket} i$$

Par convention, $0! = 1$.

Attention ! Le produit n'est pas linéaire !



Théorème

Soit a_p, \dots, a_n des éléments de \mathbb{R} et $\lambda \in \mathbb{R}$. Alors :

$$\prod_{k=p}^n \lambda a_k = \lambda^{n-p+1} \prod_{k=p}^n a_k$$

Une rapide récurrence permet de démontrer ce théorème.

Théorème : produit de fractions

Soit a_p, \dots, a_n des éléments de \mathbb{R} et b_p, \dots, b_n des éléments de \mathbb{R} {non nuls}. Alors :

$$\prod_{k=p}^n \frac{a_k}{b_k} = \frac{\prod_{k=p}^n a_k}{\prod_{j=p}^n b_j}$$

Une rapide récurrence permet de démontrer ce théorème.

Théorème : produits télescopiques

Soit p et n deux entiers naturels avec $p \leq n$ et (u_k) une suite d'éléments de \mathbb{R} non nuls. Alors :

$$\prod_{k=p}^n \frac{u_{k+1}}{u_k} = \frac{u_{n+1}}{u_p}$$

On utilise le théorème précédent et on effectue un changement d'indice comme vu avec les sommes télescopiques.

Remarque : transformation de produits en sommes

Quand tous les facteurs sont des réels strictement positifs, on peut ramener un produit à une somme en composant par \ln .

Soit par exemple $(u_k)_{1 \leq k \leq n}$ des réels. Alors :

$$\ln \left(\prod_{k=1}^n u_k \right) = \sum_{k=1}^n \ln(u_k)$$

2.3 Exercices

Calculs trigonométriques

Test 2.1 (♥)

1. Démontrer que, pour tout réel x

$$\cos^4 x + \sin^4 x = 1 - \frac{1}{2} \sin^2(2x)$$

2. Démontrer que, pour tout réel x

$$\cos^4 x + \cos^4(x + \pi/4) + \cos^4(x + 2\pi/4) + \cos^4(x + 3\pi/4) = 3/2$$

Test 2.2 (♥♥)

Déterminer une expression algébrique de $a = \cos \pi/8$ et $b = \cos 5\pi/24$.

Test 2.3 (♥♥)Résoudre dans \mathbb{R} :

1. $\sin^4 x - \frac{5}{2} \sin^2 x + 1 = 0$

2. $\sin x + \sqrt{3} \cos x = \sqrt{2}$

Test 2.4 (♥♥)

L'équation de la trajectoire d'un projectile lancé avec une vitesse initiale v_0 suivant une direction faisant un angle α avec l'horizontale est donnée par :

$$y = -\frac{g}{2v_0^2}(1 + \tan^2 \alpha)x^2 + (\tan \alpha)x, \quad x \in [0, X(\alpha)]$$

où g est l'accélération de la pesanteur et $X(\alpha)$ l'abscisse du point du sol où retombe le projectile. On suppose que v_0 est donnée et que le lanceur peut choisir α dans $]0, \pi/2[$.

1. Comment lancer le plus loin possible ?
2. Montrer que tous les points du sol accessibles, en dehors du point optimal étudié à la question précédente, peuvent être atteints de deux manières exactement : par un tir tendu et par un tir en cloche.

— Résolution de systèmes**Test 2.5 (♥)**Résoudre dans \mathbb{R}^3 le système :

$$\begin{cases} 3x - y + z = 5 \\ x + y - z = -2 \\ -x + 2y + z = 3 \end{cases}$$

Test 2.6 (Système complexe - ♥)Déterminer les nombres complexes z_1 et z_2 tels que :

$$\begin{cases} 2z_1 + z_2 = 4 \\ -2iz_1 + z_2 = 0 \end{cases}$$

Test 2.7 (♥)Résoudre dans \mathbb{R}^3 le système :

$$\begin{cases} 2x + y + z = 1 \\ x + y + 2z = 0 \\ x + 2y - z = -1 \end{cases}$$

Test 2.8 (♥♥)Résoudre dans \mathbb{R}^3 le système :

$$\begin{cases} 2x + y - z = 1 \\ 3x + 3y - z = 2 \\ 2x + 4y = 2 \end{cases}$$

Test 2.9 (♥♥)Résoudre dans \mathbb{R}^3 le système :

$$\begin{cases} 2x + y + z = 1 \\ x - y - z = 2 \\ 4x - y - z = 3 \end{cases}$$

Test 2.10 (♥♥)Résoudre dans \mathbb{R}^3 le système :

$$\begin{cases} 3x - y + z = 5 \\ 2x + y - z = 1 \\ x - y + z = 2 \\ 4x + y + z = 3 \end{cases}$$

Test 2.11 (♥♥)Résoudre dans \mathbb{R}^4 le système :

$$\begin{cases} x + y + z - t = 1 \\ x - y - z + t = 2 \\ x - y - z - t = 3 \end{cases}$$

Test 2.12 (♥♥)Résoudre dans \mathbb{R}^5 le système :

$$\begin{cases} x_1 + 2x_2 - x_3 + 3x_4 = 0 \\ x_2 + x_3 - 2x_4 + 2x_5 = 0 \\ 2x_1 + x_2 - 5x_3 - 4x_5 = 0 \end{cases}$$

Test 2.13 (Résolution de systèmes avec paramètres - ♥♥♥♥♥)Résolvez dans \mathbb{R}^3 les systèmes suivants en fonction du paramètre réel m .

$$1. \begin{cases} mx + y + z = m \\ x + my + z = m \\ x + y + mz = m \end{cases}$$

$$3. \begin{cases} x - my + m^2z = 2m \\ mx - m^2y + mz = 2m \\ mx + y - m^2z = 1 - m \end{cases}$$

$$2. \begin{cases} (m+1)x + my = 2m \\ mx + (m+1)y = 1 \end{cases}$$

— Manipulation des symboles Σ et Π **Test 2.14 (♥♥)**1. Écrire sans Σ :

$$(a) \sum_{n=0}^N (-1)^n$$

$$(b) \sum_{p=1}^n p$$

$$(c) \sum_{p=1}^n n$$

$$(d) \sum_{p=1}^n \frac{p}{n}$$

2. Écrire avec Σ :

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \pm \frac{1}{99}$$

3. Les égalités suivantes sont-elles vraies :

$$\sum_{p=1}^n (p \times n) = \sum_{k=1}^n (k \times n) = n \times \sum_{k=1}^n k = k \times \sum_{k=1}^n n$$

4. Soit

$$S_n = \sum_{p=1}^{2n^2} \frac{1}{\sqrt{p}}$$

(a) Combien y a-t-il de termes dans cette somme ?

(b) Écrire la somme S_{2n} . Combien contient-elle de termes ?

(c) Soit S_{2n}^{impair} la somme des termes de S_{2n} d'ordre impair (obtenus pour $p=1, 3, 5, \dots$). On définit de même S_{2n}^{pair} ; on a donc $S_{2n} = S_{2n}^{\text{impair}} + S_{2n}^{\text{pair}}$. Écrire les sommes S_{2n}^{pair} et S_{2n}^{impair} .

5. Calculez les sommes suivantes sachant que

$$\sum_{i=1}^x i = \frac{x(x+1)}{2} \quad \text{et} \quad \sum_{i=1}^x i^2 = \frac{x(x+1)(2x+1)}{6} ;$$

$$(a) \sum_{k=1}^n \sum_{p=0}^k p \quad (b) \sum_{k=1}^n \sum_{p=0}^k k \quad (c) \sum_{k=1}^n \sum_{p=0}^k n$$

Test 2.15 (♥♥)

Calculez la somme des carrés des entiers impairs successifs inférieurs à 100 à partir de 1.

Vérifiez avec **Python**.

Test 2.16 (♥♥)

1. Développez $(k+1)^3 - k^3$.

2. En remarquant que $\sum_{k=1}^n k^3 + (n+1)^3 = \sum_{k=1}^{n+1} k^3$, déterminez une nouvelle démonstration de la formule de la somme des carrés.

Test 2.17 (♥)

- Calculez $P = \prod_{k=2}^n \left(1 - \frac{1}{k}\right)$
- Calculez $S = \sum_{k=2}^n \ln \left(1 - \frac{1}{k}\right)$

Test 2.18 (♥)

Calculez $S_1 = \sum_{k=1}^n 2^{2k-1}$ et $S_2 = \sum_{k=1}^n \frac{(-1)^k}{2^{k-1}}$.

Test 2.19 (♥♥)

Calculez $S = \sum_{1 \leq i \leq j \leq n} ij$. Testez la formule obtenue en créant une fonction **Python** qui calcule S en fonction de n .

Test 2.20 (♥♥)

1. Prouvez que $P_i = \prod_{k=1}^i (2k-1) = \frac{(2i)!}{2^i i!}$
2. Écrivez une fonction **Python** qui calcule le produit des n premiers entiers impairs.
3. Soit $p \in \mathbb{N}^*$. Exprimez $P = \prod_{k=1}^p \frac{2k-1}{2k}$ à l'aide de la notation factorielle.

Test 2.21 (♥♥)

1. Déterminez des constantes réelles a , b , et c telles que :

$$\forall k \geq 3, \frac{1}{k(k^2-4)} = \frac{a}{k-2} + \frac{b}{k} + \frac{c}{k+2}$$

2. Déduisez-en une expression en fonction de n de la somme $S = \sum_{k=3}^n \frac{1}{k(k^2-4)}$ pour $n \geq 3$.

Dérivation**Test 2.22 (Calcul de dérivées - ♥)**

Calculez les dérivées des fonction définies par :

- | | |
|---|--|
| 1. $f_1(x) = (x^2 + x)^3$ | 9. $f_9(x) = \frac{1}{\cos((x^2 + 5x + 1)^2)}$ |
| 2. $f_2(x) = (3x^2 + 4x - 6)^4$ | 10. $f_{10}(x) = \tan(x)$ |
| 3. $f_3(x) = (3x^2 + 4x - 6)^{-4}$ | 11. $f_{11}(x) = \tan(\sqrt{x})$ |
| 4. $f_4(x) = \frac{1}{(3x^2 - 5x + 1)}$ | 12. $f_{12}(x) = \sqrt{x + \sqrt{x^2 + 2x + 1}}$ |
| 5. $f_5(x) = \cos(\sqrt{x})$ | 13. $f_{13}(x) = (\sqrt{x} + 4)(\sqrt{x} - 4)$ |
| 6. $f_6(x) = \sqrt{\cos(x)}$ | 14. $f_{14}(x) = \frac{\sin(2x)}{\cos(3x)}$ |
| 7. $f_7(x) = (\cos(x^2 + 5x + 1))^2$ | 15. $f_{15}(x) = \sqrt{\sqrt{x}}$ |
| 8. $f_8(x) = \cos((x^2 + 5x + 1)^2)$ | 16. $f_{16}(x) = \ln(\ln(\ln(\ln(x))))$ |

Test 2.23 (Vrai ou faux sur le sens de variation - ♥♥)

1. Une fonction strictement croissante tend vers $+\infty$.
2. Une fonction dérivable en a est continue en a .

3. Une fonction continue en a est dérivable en a .
4. Si f est dérivable sur un intervalle I , sa dérivée est continue sur I .
5. Une fonction dont la dérivée tend vers 0 en $+\infty$ est bornée.
6. Une fonction strictement croissante a une dérivée strictement positive.
7. Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ une fonction croissante et x et y deux réels.
 - (a) Si $f(x) \leq f(y)$ alors $x \leq y$.
 - (b) Si $f(x) < f(y)$ alors $x \leq y$.
8. Toute fonction monotone de $[0,1]$ vers \mathbb{R} est bornée.
9. Si une fonction de \mathbb{R} vers \mathbb{R} est dérivable et admet une limite finie en $+\infty$, alors sa dérivée tend vers 0 en $+\infty$.
10. Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ dérivable.
 - (a) Si f est périodique, alors f' est périodique.
 - (b) Si f' est périodique, alors f est périodique.
 - (c) Si f est paire, alors f' est impaire.
 - (d) Si f est impaire, alors f' est paire.
 - (e) Si f' est impaire, alors f est paire.
 - (f) Si f' est paire, alors f est impaire.

— Fonctions logarithme et exponentielle

Test 2.24 (♥)

Développez et réduisez au maximum les expressions suivantes :

- | | | |
|-----------------------------|-----------------------------|---|
| 1. $e^x e^{-x}$ | 6. $\frac{(e^x)^3}{e^{2x}}$ | 11. $\frac{e^{-4x} e}{(e^{-x})^2}$ |
| 2. $e^x e^{-x+1}$ | 7. $e^x (e^x + e^{-x})$ | 12. $(e^x + e^{-x})^2 - (e^x - e^{-x})^2$ |
| 3. $e e^{-x}$ | 8. $(e^x)^5 (e^{-2x})^2$ | 13. $(e^x - e^{-x})^2 - e^{-x} (e^{3x} - e^{-x})$ |
| 4. $(e^{-x})^2$ | 9. $e^{-3x+1} (e^x)^3$ | 14. $(e^x - e^{-x})(e^{2x} + e^x + 1)$ |
| 5. $\frac{e^{2x}}{e^{2-x}}$ | 10. $\sqrt{e^{-2x}}$ | |

Test 2.25 (♥)

Calculez et factorisez les dérivées et les limites, quand elles existent, aux bornes des ensembles de définition des fonctions définies par les expressions suivantes :

- | | | |
|---------------------------------------|---------------------------------------|--|
| 1. $f_1(x) = e^x + x^2 + 1$ | 6. $f_6(x) = \frac{x^2 e^x}{x+1}$ | 11. $f_{11}(x) = e^{4x+1}$ |
| 2. $f_2(x) = 5e^x + 5xe^x$ | 7. $f_7(x) = \frac{e^x}{x}$ | 12. $f_{12}(x) = e^{5x^3+7x+4}$ |
| 3. $f_3(x) = \frac{e^x + 1}{e^x - 1}$ | 8. $f_8(x) = \frac{1}{e^x}$ | 13. $f_{13}(x) = (x+1)e^{-x+1}$ |
| 4. $f_4(x) = \frac{3x+1-e^x}{e^x}$ | 9. $f_9(x) = (e^x)^2 + \frac{1}{e^x}$ | 14. $f_{14}(x) = \frac{e^{2x} - 1}{x}$ |
| 5. $f_5(x) = x^3 e^{-x}$ | 10. $f_{10}(x) = e^{-x}$ | 15. $f_{15}(x) = e^{\cos(x)}$ |

Test 2.26 (♥♥)

La fonction de croissance de VON BERTANLANFFY donne approximativement la masse $W(t)$ (en kg) à l'âge t (en années) des éléphants africains. Son expression est :

$$W(t) = 2\,600 (1 - 0,51e^{-0,075t})^3$$

1. Évaluez la masse et le taux de croissance d'un nouveau-né (le taux de croissance à l'instant t est évidemment $W'(t)$).
2. Calculez et interprétez $\lim_{t \rightarrow +\infty} W(t)$.
3. Montrez que le taux de croissance le plus fort se situe entre 5 et 6 ans.

Test 2.27 (♥♥)

Les molécules d'un gaz enfermé dans un récipient à la température T sont animées d'une vitesse de v cm.s⁻¹. Cet état d'équilibre est caractérisé par la fonction de distribution de vitesse de MAXWELL-BOLTZMANN :

$$f(v) = cv^2 e^{-mv^2/(2kT)}$$

où T est la température (en K), m la masse d'une molécule et c et k des constantes positives. Montrez que la valeur maximale de f a lieu en $v = \sqrt{2kT/m}$.

Test 2.28 (♥♥)

Résoudre dans $]0, +\infty[$ l'équation $x^{(x^x)} = (x^x)^x$.

Test 2.29 (Trigonométrie exponentielle - ♥♥♥♥)

On appelle cosinus hyperbolique la fonction définie sur \mathbb{R} par :

$$\text{ch} : x \mapsto \text{ch}(x) = \frac{e^x + e^{-x}}{2}$$

sinus hyperbolique la fonction définie sur \mathbb{R} par :

$$\text{sh} : x \mapsto \text{sh}(x) = \frac{e^x - e^{-x}}{2}$$

et tangente hyperbolique la fonction définie par :

$$\text{th} : x \mapsto \frac{\text{sh}(x)}{\text{ch}(x)}$$

1. Déterminez les dérivées de ces fonctions en fonction de ch et sh .
2. Étudiez ces fonctions pour montrer que ch est à valeurs dans $[1, +\infty]$ et que th est définie sur \mathbb{R} et à valeurs dans $[-1, 1]$.
3. Calculez $\text{th}(x)$ en fonction de e^x et e^{-x} , puis en fonction de e^{2x} , enfin en fonction de e^{-2x} .
4. Montrez que $\text{ch}^2(x) - \text{sh}^2(x) = 1$.
5. Montrez que $\text{ch}(x+y) = \text{ch}(x)\text{ch}(y) + \text{sh}(x)\text{sh}(y)$ et que $\text{sh}(x+y) = \text{sh}(x)\text{ch}(y) + \text{ch}(x)\text{sh}(y)$.
6. Dédisez-en que $\text{th}(x+y) = \frac{\text{th}(x) + \text{th}(y)}{1 + \text{th}(x)\text{th}(y)}$.

7. On pose $t = \operatorname{th}(x/2)$. Montrez que $\operatorname{ch}(x) = \frac{1+t^2}{1-t^2}$ puis que $\operatorname{sh}(x) = \frac{2t}{1-t^2}$.
8. Résolvez dans \mathbb{R} l'équation $5\operatorname{ch}(x) - 4\operatorname{sh}(x) = 3$. Vous donnerez une valeur approchée de la solution à 10^{-3} près.
9. Pour le plaisir : dérivez la fonction $x \mapsto \frac{2\sin(x)\operatorname{sh}(x)}{(\operatorname{sh}(x) + \sin(x))^2}$
10. Soit $y \in \mathbb{R}$. Comparez $\operatorname{sh}(y)$ et y .
11. Montrez que $\operatorname{sh}(2x) = 2\operatorname{sh}(x)\operatorname{ch}(x)$ puis étudiez la fonction f définie sur $]0, +\infty[$ par

$$f(x) = \frac{\operatorname{th}(x)}{x}$$

Coups de pouce

Coup de pouce pour le test 2.1

1. Exprimer $\sin^2 x$ et $\cos^2 x$ en fonction de $\cos(2x)$.
2. Penser à $\cos(x + \pi/2) = -\sin x$.

Coup de pouce pour le test 2.2

$\cos \pi/8$ est solution d'une équation du second degré si vous pensez à la formule bien connue $\cos 2x = 2\cos^2 x - 1$ et $\frac{5\pi}{24}$ est la moitié de la moitié de $\frac{5\pi}{6}$.

Coup de pouce pour le test 2.3

1. Posez $t = \sin^2 x$.
2. Utilisez une formule du style : $\cos a \cos b + \sin a \sin b = \cos(a - b)$.

Coup de pouce pour le test 2.4

Pour un observateur immobile et sous abris, la trajectoire est une portion de parabole, son équation étant du second degré en x . Donc $X(\alpha)$ est la racine non nulle de ce trinôme. Comme son nom l'indique, sa valeur dépend de α : vous devez donc calculer $X(\alpha)$ et déterminer la (?) valeur de α qui maximise $X(\alpha)$.

Coup de pouce pour le test 2.5

On applique l'algorithme du pivot de GAUSS. La deuxième ligne peut fournir le premier pivot.

Coup de pouce pour le test 2.6

On applique toujours la même méthode, même si les nombres sont complexes.

Coup de pouce pour le test 2.7

Méthode du pivot.

Coup de pouce pour le test 2.8

La dernière ligne peut fournir le premier pivot. S'il n'y a que deux pivots, pensez à prendre une des inconnues comme paramètre.

Coup de pouce pour le test 2.9

Vous pouvez une nouvelle fois adopter la méthode du pivot de GAUSS mais attention à avoir un œil critique sur les résultats obtenus : trois plans ne sont pas forcément sécants...

Coup de pouce pour le test 2.10

Il y a plus d'équations que d'inconnues mais la méthode reste la même : celle du pivot de GAUSS...

Coup de pouce pour le test 2.11

Il y a cette fois quatre inconnues mais... la méthode est toujours la même ;-)

Coup de pouce pour le test 2.12

Toujours plus fort : cinq inconnues ! On utilise encore la même méthode. Le nombre d'inconnues moins le nombre de pivots nous donnera le nombre d'inconnues non principales, c'est-à-dire de paramètres.

Coup de pouce pour le test 2.13

Il faut être très organisé dans les calculs et bien réfléchir à chaque étape et être prudent : si une division est envisagée, est-elle licite ? Si elle ne l'est pas, il faut alors remplacer m par les valeurs « dangereuses » puis ensuite calculer avec des valeurs de m supposées différentes.

Coup de pouce pour le test 2.14

Il faut bien regarder qui varie et qui est fixe dans la somme et utiliser les propriétés de distributivité du produit sur la somme et des factorisation.

Dans $\sum_{i=2}^n i^2$, i varie de 2 à n (n est donc fixe) : il s'agit de la somme des i^2 pour i variant de 2 à n .

Un résultat classique démontré en Première et en Terminale est à connaître :

$$S_n = \sum_{i=1}^n i = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

De nombreuses démonstrations existent dont celle-ci proposée, selon la légende, par GAUSS alors qu'il était âgé de 7 ans...

$$\begin{array}{ccccccccc} S_n & = & 1 & & + & 2 & & + & \dots & + & n-1 & & + & n \\ S_n & = & n & & + & n-1 & & + & \dots & + & 2 & & + & 1 \\ \hline 2S_n & = & (n+1) & & + & (n+1) & & + & \dots & + & (n+1) & & + & (n+1) \end{array}$$

Coup de pouce pour le test 2.15

1, 3, 5, ..., 99 peuvent s'écrire sous la forme $2 \times 0 + 1$, $2 \times 1 + 1$, $2 \times 2 + 1$, ..., $2 \times 49 + 1$.

Coup de pouce pour le test 2.16

Il est utile de connaître le développement remarquable suivant : $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$. Utilisez ensuite la linéarité de la somme comme à l'exercice précédent après une réindexation.

Coup de pouce pour le test 2.17

Pensez à un produit télescopique après réduction au même dénominateur.

Pour la somme, n'oubliez pas que $\ln a + \ln b = \ln(a \times b)$.

Coup de pouce pour le test 2.18

Cherchez à retrouver une somme du type $\sum_{k=1}^n q^k$.

Coup de pouce pour le test 2.19

Attention ! Par rapport à l'exercice vu en cours à la page 28, ici i varie entre 1 et j .

Coup de pouce pour le test 2.20

Ruse habituelle : quand on n'a pas ce qu'on veut, on le fait apparaître...

$$\text{Ici, } P_i = \prod_{k=1}^n (2k-1) = \frac{(2n)(2n-1)(2n-2)\cdots 4 \times 3 \times 2 \times 1}{(2n)(2n-2)\cdots 4 \times 2}$$

et on peut faire apparaître des factorielles.

Coup de pouce pour le test 2.21

1. Réduisez au même dénominateur et comparez les numérateurs.

2. Linéarité puis réindexation.

Coup de pouce pour le test 2.22

Vous pourrez vérifier vos calculs à la machine. On obtient facilement l'expression formelle de la dérivée d'une fonction :

```
In [301]: x,t = sy.symbols('x t')
In [302]: f = lambda x: (x**2 + 3*x)**2
In [303]: sy.diff(f(x), x)
Out[303]:
      2
(4.x + 6). [x + 3.x]
```

On a deux solutions pour faire de l'expression de la dérivée une fonction :

```
In [304]: fp1=lambda x: sy.diff(f(t), t).
subs(t,x)
In [305]: fp1(x)
Out[305]:
      2
(4.x + 6). [x + 3.x]
In [306]: fp1(t)
Out[306]:
      2
(4.t + 6). [t + 3.t]
In [307]: fp1(2)
Out[307]: 140
```

ou

```
In [308]: fp2=sy.lambdify(x, sy.diff(f(x),
x))
In [309]: fp2(x)
Out[309]:
      2
(4.x + 6). [x + 3.x]
In [310]: fp2(2)
Out[310]: 140
```

Coup de pouce pour le test 2.24

Une petite révision sur les formules algébriques du type :

- $e^{a+b} = e^a e^b$
- $e^{a-b} = e^a / e^b$
- $(e^a)^b = e^{ab}$

Coup de pouce pour le test 2.25

Quelques rappels :

- $\lim_{x \rightarrow +\infty} e^x / x^n = +\infty$
- $\lim_{x \rightarrow +\infty} x^n e^{-x} = 0$
- $\lim_{x \rightarrow 0} \frac{e^x - 1}{x} = \lim_{x \rightarrow 0} \frac{e^x - e^0}{x - 0} = e^0 = 1$

Vérifiez si la fonction est paire ou impaire : cela peut diviser le domaine d'étude en deux.

Coup de pouce pour le test 2.28

Utilisez la bijectivité de la fonction \ln , en faisant attention aux cas particuliers.

Solutions et commentaires

Solution du test 2.1

1. D'une part $\sin^2 x = 1 - \cos^2 x$. D'autre part $\cos(x+y) = \cos x \cos y - \sin x \sin y$. Cela permet de retrouver les fameuses formules :

$$\cos(2x) = \cos(x+x) = \cos^2 x - \sin^2 x = 2\cos^2 x - 1 = 1 - 2\sin^2 x$$

qui permettent d'obtenir :

$$\cos^2 x = \frac{1 + \cos(2x)}{2} \quad \sin^2 x = \frac{1 - \cos(2x)}{2}$$

On élève au carré :

$$\cos^4 x = \frac{1 + \cos^2(2x) + 2\cos(2x)}{4} \quad \sin^4 x = \frac{1 + \cos^2(2x) - 2\cos(2x)}{4}$$

Il ne reste plus qu'à additionner :

$$\cos^4 x + \sin^4 x = \frac{1 + \cos^2(2x)}{2} = \frac{1 - \sin^2(2x)}{2}$$

Solution du test 2.2

$$\cos \frac{\pi}{4} = 2\cos^2 \frac{\pi}{8} - 1 = \frac{\sqrt{2}}{2} \quad \text{donc} \quad \cos \frac{\pi}{8} = \frac{\sqrt{\sqrt{2}+2}}{2} \text{ car } 0 < \frac{\pi}{8} < \frac{\pi}{2}.$$

Nous obtenons donc par analogie avec le calcul précédent :

$$\cos \frac{5\pi}{24} = \frac{1}{2} \sqrt{2 + \sqrt{2 - \sqrt{3}}}$$

D'autres formules sont possibles, car en étant plus rusés, nous aurions pu observer que :

$$\frac{5\pi}{24} = \frac{\pi}{3} - \frac{\pi}{8}$$

Solution du test 2.3

1. Avec $t = \sin^2 x$, nous sommes ramenés à résoudre une équation du second degré. Nous obtenons une solution : $1/2$, la deuxième solution étant à exclure d'après nos conditions. Les solutions cherchées vérifient donc l'équation $\sin x = \pm\sqrt{2}/2$. Les solutions sont les réels de la forme $\pi/4 + k\pi/2$ avec $k \in \mathbb{Z}$.

2. On a en tête les lignes trigonométriques de référence. $\sqrt{3}$ nous fait penser à $\sqrt{3}/2$ et $\sqrt{2}$ à $\sqrt{2}/2$. Nous obtenons donc l'équation :

$$(E) : \frac{\sqrt{3}}{2} \cos x + \frac{1}{2} \sin x = \frac{\sqrt{2}}{2} \quad (E) \Leftrightarrow \cos\left(x - \frac{\pi}{6}\right) = \cos\left(\frac{\pi}{4}\right)$$

Sachant que $\sqrt{3}/2 = \cos(\pi/6)$, $1/2 = \sin(\pi/6)$ et $\sqrt{2}/2 = \cos(\pi/4)$.

D'où $(E) \Leftrightarrow x - \frac{\pi}{6} \equiv \pm \frac{\pi}{4} [2\pi]$. Les solutions sont donc les réels de la forme $\frac{5\pi}{12} + 2k\pi$ ou $-\frac{\pi}{12} + 2k\pi$ avec $k \in \mathbb{Z}$.

Solution du test 2.4

$$1. X(\alpha) = \frac{2v_0^2 \tan \alpha}{g(1 + \tan^2 \alpha)} = \frac{2v_0^2 \tan \alpha}{g} \cos^2 \alpha = \frac{2v_0^2 \sin \alpha}{g \cos \alpha} \cos^2 \alpha = \frac{v_0^2}{g} 2 \sin \alpha \cos \alpha = \frac{v_0^2}{g} \sin 2\alpha.$$

$X(\alpha)$ est donc maximum lorsque $\sin 2\alpha$ est maximum, c'est-à-dire lorsque $\alpha = \pi/4$.

2. L'équation $\frac{v_0^2}{g} \sin \beta = x_0$ avec $x_0 \in]0; v_0^2/g[$ est équivalente à $\sin \beta = \frac{gx_0}{v_0^2}$ et admet deux solutions réelles dans $]0; \pi[: \beta_0$ et $\beta_1 = \pi - \beta_0$ avec $0 < \beta_0 = 2\alpha_0 < \pi/2 < \beta_1 = 2\alpha_1$. Ainsi, seulement deux angles de départ α_0 et α_1 , avec $\alpha_0 < \pi/4 < \alpha_1$, vont donner le même point d'impact. Le premier correspond à un tir tendu, le deuxième à un tir en cloche.

Solution du test 2.5

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y - z = -2 & (L_2) \\ -4y + 4z = 11 & (L_1 - 3L_2) \\ 3y = 1 & (L_2 + L_3) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y - z = -2 \\ \boxed{y} = \frac{1}{3} & \frac{1}{3}(L_3) \\ \boxed{z} = \frac{37}{12} & \left(\frac{1}{4}L_2 + \frac{1}{3}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = \frac{3}{4} & (L_1 - L_2 + L_3) \\ \boxed{y} = \frac{1}{3} \\ \boxed{z} = \frac{37}{12} \end{cases}$$

$$\mathcal{S} = \left\{ \left(\frac{3}{4}, \frac{1}{3}, \frac{37}{12} \right) \right\}$$

On vérifie avec **Python** :

```
In [201]: x,y,z = sy.symbols('x y z')
In [202]: sy.solve([x + y - z + 2, -4*y + 4*z - 11, 3*y - 1], [x,y,z])
Out[202]: {x: 3/4, z: 37/12, y: 1/3}
```

Solution du test 2.6

$$(S) \Leftrightarrow \begin{cases} 2z_1 + z_2 = 4 \\ (1+i)z_2 = 4i & (iL_1 + L_2) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} 2z_1 + z_2 = 4 \\ z_2 = \frac{4i}{1+i} = \frac{4i(1-i)}{2} = 2+2i & \left(\frac{1}{1+i}L_2\right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} 2z_1 = 2-2i & (L_1 - L_2) \\ \boxed{z_2} = 2+2i \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{z_1} & = 1-i & \left(\frac{1}{2}L_1\right) \\ \boxed{z_2} & = 2+2i \end{cases}$$

$$\mathcal{S} = \{(1-i, 2+2i)\}$$

```
In [211]: z1,z2 = sy.symbols('z1 z2', complex=True)
In [212]: sy.solve([2*z1 + z2 -4 , -2j*z1 + z2 ], [z1,z2])
Out [212]: {z1: 1.0 - 1.0*I, z2: 2.0 + 2.0*I}
```

Solution du test 2.7

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + 2y - z = -1 & (L_3) \\ y - 3z = -1 & (L_3 - L_2) \\ 3y - 3z = -3 & (2L_3 - L_1) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + 5z = 1 & (L_1 - 2L_2) \\ \boxed{y} - 3z = -1 \\ 6z = 0 & (L_3 - 3L_2) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = 1 \\ \boxed{y} = -1 \\ \boxed{z} = 0 \end{cases}$$

$$\mathcal{S} = \{(1, -1, 0)\}$$

```
In [22]: sy.solve([2*x + y + z - 1, x + y + 2*z, x + 2*y - z + 1], [x,y,z])
Out [22]: {x: 1, z: 0, y: -1}
```

Solution du test 2.8

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + 2y = 1 & \left(\frac{1}{2}L_3\right) \\ -3y - z = -1 & (L_1 - L_3) \\ -3y - z = -1 & \left(L_2 - \frac{3}{2}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + 2y = 1 \\ \boxed{y} + \frac{1}{3}z = \frac{1}{3} & \left(-\frac{1}{3}L_2\right) \end{cases}$$

$$(S) \Leftrightarrow \exists \alpha \in \mathbb{R} \mid \begin{cases} x = \frac{1}{3}(1+2\alpha) \\ y = \frac{1}{3}(1-\alpha) \\ z = \alpha \end{cases}$$

Le système a deux pivots. L'ensemble des solutions est :

$$\mathcal{S} = \left\{ \left(\frac{1}{3}(1+2\alpha), \frac{1}{3}(1-\alpha), \alpha \right) \in \mathbb{R}^3 / \alpha \in \mathbb{R} \right\}$$

Avec **Python** :

```
In [23]: sy.solve([2*x + y - z - 1, 3*x + 3*y - z - 2, 2*x + 4*y - 2], [x,y,z])
Out [23]: {x: 2*z/3 + 1/3, y: -z/3 + 1/3}
```

Solution du test 2.9

On choisit la deuxième ligne pour le premier pivot.

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y - z = 2 & (L_2) \\ 3y + 3z = -3 & (L_1 - 2L_2) \\ 3y + 3z = -5 & (L_3 - 4L_2) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y - z = 2 \\ \boxed{y} + z = -1 & \left(\frac{1}{3}L_2 \right) \\ 0 = -2 & (L_3 - L_2) \end{cases}$$

Il y a deux pivots mais la dernière équation rend le système incompatible. L'ensemble des solutions est vide : $\mathcal{S} = \emptyset$.

```
In [24]: sy.solve([2*x + y + z - 1, x - y - z - 2, 4*x - y - z - 3], [x,y,z])
Out [24]: [ ]
```

Solution du test 2.10

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y + z = 2 & (L_3) \\ 2y - 2z = -1 & (L_1 - 3L_3) \\ 3y - 3z = -3 & (L_2 - 2L_3) \\ 5y - 3z = -3 & (L_4 - 4L_3) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y + z = 2 \\ \boxed{y} - z = -1 & \left(\frac{1}{3}L_3 \right) \\ 0 = 1 & \left(L_2 - \frac{2}{3}L_3 \right) \\ 2z = 2 & \left(L_4 - \frac{5}{3}L_3 \right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y + z = 2 \\ \boxed{y} - z = -1 \\ \boxed{z} = 1 & (L_4) \\ 0 = 1 & (L_3) \end{cases}$$

Il y a trois pivots mais la dernière équation rend le système incompatible. L'ensemble des solutions est vide : $\mathcal{S} = \emptyset$.

```
In [25]: sy.solve([3*x - y + z - 5, 2*x + y - z - 1, x - y + z - 2,
                  4*x + y + z - 3], [x, y, z])

Out [25]: [ ]
```

Solution du test 2.11

On garde la première ligne pour le premier pivot.

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + z - t = 1 \\ 2y + 2z + t = -1 & (L_1 - L_2) \\ 2y + 2z - t = -2 & (L_1 - L_3) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + z - t = 1 \\ \boxed{y} + z = -1 & (L_3) \\ 2t = -1 & (L_3 - L_2) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - t = 2 & (L_1 - L_2) \\ \boxed{y} + z = -1 \\ \boxed{t} = -\frac{1}{2} & \left(\frac{1}{2}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - t = 2 \\ \boxed{y} + z = -1 \\ \boxed{t} = -\frac{1}{2} & \left(\frac{1}{2}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \exists \alpha \in \mathbb{R} \left\{ \begin{array}{l} x = \frac{1}{3}(1 + 2\alpha) \\ y = -1 - \alpha \\ z = \alpha \\ t = -\frac{1}{2} \end{array} \right.$$

Il y a trois pivots : le système est de rang 3. L'ensemble des solutions est :

$$\mathcal{S} = \left\{ \left(\frac{3}{2}, -1, -\alpha, \alpha, -\frac{1}{2} \right) \in \mathbb{R}^4 / \alpha \in \mathbb{R} \right\}$$

```
In [261]: t = sy.symbols('t')
In [262]: sy.solve([x + y + z - t - 1, x - y - z + t - 2,
                    x - y - z - t - 3], [x, y, z, t])
Out[262]: {x: 3/2, t: -1/2, y: -z-1}
```

Solution du test 2.12

On garde les deux premières lignes pour les deux premiers pivots.

$$(S) \Leftrightarrow \begin{cases} \boxed{x_1} + 2x_2 - x_3 + 3x_4 = 0 \\ \boxed{x_2} + x_3 - 2x_4 + 2x_5 = 0 \\ -3x_2 - 3x_3 - 6x_4 - 4x_5 = 0 \quad (L_3 - 2L_1) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x_1} + 2x_2 - x_3 + 3x_4 = 0 \\ \boxed{x_2} + x_3 - 2x_4 + 2x_5 = 0 \\ -12x_4 + 2x_5 = 0 \quad (3L_2 - L_3) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x_1} + 2x_2 - x_3 + 3x_4 = 0 \\ \boxed{x_2} + x_3 - 2x_4 + 2x_5 = 0 \\ \boxed{x_4} - \frac{1}{6}x_5 = 0 \quad \left(-\frac{1}{12}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \exists (\alpha, \beta) \in \mathbb{R}^2 \left\{ \begin{array}{l} x_1 = 3\alpha + \frac{17}{6}\beta \\ x_2 = -\alpha - \frac{5}{3}\beta \\ x_3 = \alpha \\ x_4 = \frac{1}{6}\beta \\ x_5 = \beta \end{array} \right.$$

Il y a trois pivots pour 5 inconnues : il y a donc trois inconnues principales et 2 inconnues non principales qui « servent » de paramètres. L'ensemble des solutions est :

$$\mathcal{S} = \left\{ \left(3\alpha + \frac{17}{6}\beta, -\alpha - \frac{5}{3}\beta, \alpha, \frac{1}{6}\beta, \beta \right) \in \mathbb{R}^5 / (\alpha, \beta) \in \mathbb{R}^2 \right\}$$

```
In [271]: x1,x2,x3,x4,x5 = sy.symbols('x1 x2 x3 x4 x5')
In [272]: sy.solve([x1 + 2*x2 - x3 + 3*x4 ,
                    x2 + x3 - 2*x4 + 2*x5 ,
                    2*x1 + x2 - 5*x3 - 4*x5 ],
                    [x1,x2,x3,x4,x5])
Out[272]: {x1: 3*x3 + 17*x5/6, x4: x5/6, x2: -x3 - 5*x5/3}
```

Solution du test 2.13

1. La troisième ligne fournit le premier pivot :

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + mz = m & (L_3) \\ x + my + z = m & (L_2) \\ mx + y + z = m & (L_1) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + mz = m \\ (m-1)y + (1-m)z = 0 & (L_2 - L_1) \\ (1-m)y + (1-m^2)z = m(1-m) & (L_3 - mL_1) \end{cases}$$

• Si $m = 1$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + z = 1 \\ 0 = 0 \\ 0 = 0 \end{cases}$$

$$(S) \Leftrightarrow \exists (\alpha, \beta) \in \mathbb{R}^2, \begin{cases} x = 1 - \alpha - \beta \\ y = \alpha \\ z = \beta \end{cases}$$

Il y a un pivot : le système est de rang 1. L'ensemble des solutions est :

$$\mathcal{S} = \{(1 - \alpha - \beta, \alpha, \beta) \in \mathbb{R}^3 \mid (\alpha, \beta) \in \mathbb{R}^2\}$$

• Si $m \neq 1$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + y + mz = m \\ y - z = 0 & \left(\frac{1}{m-1}L_2\right) \\ y + (m+1)z = m & \left(\frac{1}{1-m}L_3\right) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + (m+1)z = m & (L_1 - L_2) \\ \boxed{y} - z = 0 \\ (m+2)z = m & (L_3 - L_2) \end{cases}$$

•• Si $m = -2$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - z = -2 \\ \boxed{y} - z = 0 \\ 0 = -2 \end{cases}$$

Il y a deux pivots mais la dernière équation rend le système incompatible. L'ensemble des solutions est vide.

$$\mathcal{S} = \emptyset$$

• Si $m \notin \{-2, 1\}$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + \quad + (m+1)z = m \\ \quad \boxed{y} - \quad z = 0 \\ \quad \quad \boxed{z} = \frac{m}{m+2} \end{cases} \quad \left(\frac{1}{m+2} L_3 \right)$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = m - \frac{m(m+1)}{m+2} = \frac{m}{m+2} & L_1 - (m+1)L_3 \\ \quad \boxed{y} = \frac{m}{m+2} & (L_2 + L_3) \\ \quad \quad \boxed{z} = \frac{m}{m+2} \end{cases}$$

Le système est de rang 3 et est composé de 3 équations à 3 inconnues : on a donc une unique solution. L'ensemble des solutions est le singleton :

$$\mathcal{S} = \left\{ \left(\frac{m}{m+2}, \frac{m}{m+2}, \frac{m}{m+2} \right) \right\}$$

```
In [281]: sy.solve([m*x + y + z - m, x + m*y + z - m, x + y + m*z - m],
                  [x,y,z])
Out [281]: {x: m/(m + 2), z: m/(m+2), y: m/(m+2)}
```

Pour régler les cas particuliers, on peut décomposer le problème ainsi :

```
In [282]: eq1 = m*x + y + z - m
In [283]: eq2 = x + m*y + z - m
In [284]: eq3 = x + y + m*z - m
In [285]: sys = [eq1,eq2,eq3]
In [286]: sy.solve([eq.subs(m, 1) for eq in sys], [x,y,z])
Out [286]: {x: -y - z + 1}
In [287]: sy.solve([eq.subs(m, -2) for eq in sys], [x,y,z])
Out [287]: [ ]
```

2. Pour obtenir des pivots il faut commencer par distinguer trois cas :

• Si $m = 0$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = 0 \\ \quad \boxed{y} = 1 \end{cases}$$

Le système est de rang 2 et est composé de 2 équations à 2 inconnues : on a donc une unique solution. L'ensemble des solutions est le singleton :

$$\mathcal{S} = \{(0,1)\}$$

• Si $m = -1$ alors

$$(S) \Leftrightarrow \begin{cases} \quad \boxed{y} = 2 \\ \boxed{x} = -1 \end{cases}$$

Le système est de rang 2 et est composé de 2 équations à 2 inconnues : on a donc une unique solution. L'ensemble des solutions est le singleton :

$$\mathcal{S} = \{(-1, 2)\}$$

• Si $m \notin \{-1, 0\}$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + \frac{m+1}{m}y = \frac{1}{m} & \left(\frac{1}{m}L_2\right) \\ (2m+1)y = -2m^2 + m + 1 & ((m+1)L_2 - mL_1) \end{cases}$$

•• Si $m = -\frac{1}{2}$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - y = -2 \\ 0 = 0 \end{cases}$$

$$(S) \Leftrightarrow \exists \alpha \in \mathbb{R}, \begin{cases} x = \alpha - 2 \\ y = \alpha \end{cases}$$

Il y a un pivot : le système est de rang 1. L'ensemble des solutions est :

$$\mathcal{S} = \{(\alpha - 2, \alpha) \in \mathbb{R}^2 \mid \alpha \in \mathbb{R}\}$$

•• Si $m \notin \left\{-1, -\frac{1}{2}, 0\right\}$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + \frac{m+1}{m}y = \frac{1}{m} \\ \boxed{y} = 1 - m & \left(\frac{1}{2m+1}L_2\right) \end{cases}$$

Le système est de rang 2 et est composé de 2 équations à 2 inconnues : on a donc une unique solution. L'ensemble des solutions est le singleton :

$$\mathcal{S} = \{(m, 1 - m)\}$$

On peut en fait regrouper les deux premiers cas avec le dernier et en conclure que :

– si $m = -\frac{1}{2}$ alors le système est de rang 1 et les solutions sont les couples $(\alpha, \alpha - 2)$ avec $\alpha \in \mathbb{R}$;

– si $m \neq -\frac{1}{2}$ alors le système est de rang 2 et admet la solution unique $(m, 1 - m)$.

```
In [288]: eq1 = (m+1)*x + m*y - 2*m
In [289]: eq2 = m*x + (m+1)*y - 1
In [2810]: sys = [eq1, eq2]
In [2811]: sy.solve(sys, x, y)
Out[2811]: {x: m, y: -m + 1}
In [2812]: sy.solve([eq.subs(m, -1/2) for eq in sys], [x, y])
Out[2812]: {x: y - 2.0}
```

3. La première ligne fournit le premier pivot :

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - my + m^2z = 2m \\ (m^2+1)y - m^2(m+1)z = -2m^2-m+1 & (L_3 - mL_1) \\ m(1-m^2)z = 2m(1-m) & (L_2 - mL_1) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} - my + m^2z = 2m \\ y - \frac{m^2(m+1)}{1+m^2}z = \frac{(1-2m)(1+m)}{1+m^2} & \left(\frac{1}{1+m^2}L_2 \right) \\ m(1-m)(1+m)z = 2m(1-m) \end{cases}$$

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + \frac{m^2(1-m)}{1+m^2}z = \frac{m(3-m)}{1+m^2} & (L_1 + mL_2) \\ \boxed{y} - \frac{m^2(m+1)}{1+m^2}z = \frac{(1-2m)(1+m)}{1+m^2} \\ m(1-m)(1+m)z = 2m(1-m) \end{cases}$$

Il y a donc trois valeurs de m qui peuvent poser problème :

- Si $m = 0$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = 0 \\ \boxed{y} = 1 \\ 0 = 0 \end{cases}$$

$$(S) \Leftrightarrow \exists \alpha \in \mathbb{R}, \begin{cases} x = 0 \\ y = 1 \\ z = \alpha \end{cases}$$

Il y a deux pivots. L'ensemble des solutions est :

$$\mathcal{S} = \{(0, 1, \alpha) \in \mathbb{R}^3 \mid \alpha \in \mathbb{R}\}$$

- Si $m = 1$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = 1 \\ \boxed{y} - z = -1 \\ 0 = 0 \end{cases}$$

$$(S) \Leftrightarrow \exists \alpha \in \mathbb{R}, \begin{cases} x = 1 \\ y = \alpha - 1 \\ z = \alpha \end{cases}$$

Il y a deux pivots. L'ensemble des solutions est :

$$\mathcal{S} = \{(1, \alpha - 1, \alpha) \in \mathbb{R}^3 \mid \alpha \in \mathbb{R}\}$$

- Si $m = -1$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} + z = -2 \\ \boxed{y} = 0 \\ 0 = -4 \end{cases}$$

Il y a deux pivots. La dernière équation rend le système incompatible. L'ensemble des solutions est :

$$\mathcal{S} = \emptyset$$

- Si $m \notin \{-1, 0, 1\}$ alors

$$(S) \Leftrightarrow \begin{cases} \boxed{x} = \frac{m(m-3)}{1+m^2} - \frac{m^2(1-m)}{1+m^2} \times \frac{2}{1+m} = \frac{m(m^2+3)}{(1+m^2)(1+m)} & \left(L_1 - \frac{m^2(1-m)}{1+m^2} L_3 \right) \\ \boxed{y} = \frac{(1-2m)(1+m)}{1+m^2} + \frac{m^2(1+m)}{1+m^2} \times \frac{2}{1+m} = \frac{1-m}{1+m^2} & \left(L_2 + \frac{m^2(1+m)}{1+m^2} L_3 \right) \\ \boxed{z} = \frac{2}{1+m} & L_3 \end{cases}$$

Le système est de rang 3 et est composé de 3 équations à 3 inconnues : on a donc une unique solution.

$$\mathcal{S} = \left\{ \left(\frac{m(3+m^2)}{(m^2+1)(1+m)}, \frac{1-m}{m^2+1}, \frac{2}{1+m} \right) \right\}$$

```
In [2813]: sys = [x - m*y + m**2*z - 2*m, m*x - m**2*y + m*z - 2*m,
                  m*x + y - m**2*z - (1-m)]
In [2814]: sy.solve(sys, x, y, z)
Out[2814]: {x: m*(m**2 + 3)/(m**3 + m**2 + m + 1), z: 2/(m + 1), y: (-m + 1)/(m**2 + 1)}
In [2815]: sy.solve([eq.subs(m, 0) for eq in sys], [x,y,z])
Out[2815]: {x: 0, y: 1}
In [2816]: sy.solve([eq.subs(m, 1) for eq in sys], [x,y,z])
Out[2816]: {x: 1, y: z-1}
In [2817]: sy.solve([eq.subs(m, -1) for eq in sys], [x,y,z])
Out[2817]: [ ]
```

Solution du test 2.14

- (a) $\sum_{n=0}^N (-1)^n = (-1)^0 + (-1)^1 + \dots + (-1)^N = (1-1) + (1-1) + \dots + (1-1) + ?$, la valeur de ? dépendant de la parité de N. La somme vaut donc 1 si N est pair, 0 sinon.

$$(b) \sum_{p=1}^n p = \frac{n(n+1)}{2}$$

$$(c) \sum_{p=1}^n n = n + n + \dots + n = n^2$$

$$(d) \sum_{p=1}^n \frac{p}{n} = \frac{1}{n} \times \frac{n(n+1)}{2} = \frac{n+1}{2}$$

$$2. 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \pm \frac{1}{99} = \sum_{k=0}^{49} \frac{(-1)^k}{2k+1}$$

3. Les trois premiers membres sont bien égaux $\left(\text{à } \frac{n^2(n+1)}{2} \right)$. Le dernier est égal à kn^2 .

4. (a) Quand on compte de 1 à $2n^2$ avec ses doigts, on utilise $2n^2$ doigts.

$$(b) S_{2n} = \sum_{p=1}^{2(2n)^2} \frac{1}{\sqrt{p}} = \sum_{p=1}^{8n^2} \frac{1}{\sqrt{p}}. S_{2n} \text{ contient donc } 8n^2 \text{ termes.}$$

$$(c) S_{2n} = \frac{1}{\sqrt{2 \times 0 + 1}} + \frac{1}{\sqrt{2 \times 1}} + \frac{1}{\sqrt{2 \times 1 + 1}} + \frac{1}{\sqrt{2 \times 2}} + \dots + \frac{1}{\sqrt{2 \times (4n^2 - 1) + 1}} + \frac{1}{\sqrt{2 \times 4n^2}}.$$

$$\text{Donc } S_{2n}^{\text{pair}} = \sum_{k=1}^{4n^2} \frac{1}{\sqrt{2k}} = \frac{1}{\sqrt{2}} \sum_{k=1}^{4n^2} \frac{1}{\sqrt{k}} \text{ et}$$

$$S_{2n}^{\text{impair}} = \sum_{k=0}^{4n^2-1} \frac{1}{\sqrt{2k+1}}$$

$$5. (a) \sum_{k=1}^n \sum_{p=0}^k p = \sum_{k=1}^n \frac{k(k+1)}{2} = \frac{1}{2} \left(\sum_{k=1}^n k^2 + \sum_{k=1}^n k \right) = \frac{1}{2} \left(\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right) = \frac{n(n+1)(n+2)}{6}$$

$$(b) \sum_{k=1}^n \sum_{p=0}^k k = \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$(c) \sum_{k=1}^n \sum_{p=0}^k n = \sum_{k=1}^n nk = \frac{n^2(n+1)}{2}$$

Pour vérifier tous ces résultats, nous utilisons la fonction `summation` de **sympy**. On en profite pour ajouter une option qui permet d'obtenir un joli affichage...

Voici le préambule correspondant à toutes les lignes de code de l'exercice :

- import de la bibliothèque **sympy** de calcul symbolique ;
- joli affichage ;
- déclaration de la variable **x** et du paramètre **k**.

```
In [291]: import sympy as sy
In [292]: sy.init_printing()
In [293]: k, n, p = sy.symbols('k n p')
```

```
In [1a]: sy.summation((-1)**k, (k,1,n))
Out [1a]:
      n+1
  -  (-1)  -  1
     2      2

In [1b]: sy.summation(k, (k,1,n)) # somme pour k variant de 1 à n
Out [1b]:
      2
  n  +  n
  2   2
```



```

In [1c]: sy.summation(n, (k,1,n))
Out [1c]:
  2
n
In [1d]: sy.summation(k/n, (k,1,n))
Out [1d]:
  2
  n + n
  2   2
  ---
  n
In [3a]: sy.summation(k*n, (k,1,n))
Out [3a]:
  [ 2 ]
  |n  n|
n. |- + -|
  [2  2]
In [3b]: n*sy.summation(k, (k,1,n))
Out [3b]:
  [ 2 ]
  |n  n|
n. |- + -|
  [2  2]
In [3c]: k*sy.summation(k, (k,1,n))
Out [3c]:
  [ 2 ]
  |n  n|
k. |- + -|
  [2  2]
In [5a]: sy.summation(sy.summation(p, (p,1,k)), (k,1,n))
Out [5a]:
  3  2
  n + n + n
  6  2  3
In [5b]: sy.summation(sy.summation(k, (p,1,k)), (k,1,n))
Out [5b]:
  3  2
  n + n + n
  3  3  6
In [5c]: sy.summation(sy.summation(n, (p,1,k)), (k,1,n))
Out [5c]:
  [ 2 ]
  |n  n|
n. |- + -|
  [2  2]

```

Solution du test 2.15

$$S = 1^2 + 3^2 + 5^2 + \dots + 99^2 = (2 \times 0 + 1)^2 + (2 \times 1 + 1)^2 + (2 \times 2 + 1)^2 + \dots + (2 \times 49 + 1)^2$$

$$S = \sum_{k=0}^{49} (2k+1)^2 = 4 \sum_{k=0}^{49} k^2 + 4 \sum_{k=0}^{49} k + \sum_{k=0}^{49} 1 = 4 \frac{49 \times 50 \times 99}{6} + 4 \frac{49 \times 50}{2} + 50 \times 1 = 166650$$

Avec **Python**, on peut utiliser directement `sum` :

```
>>> sum([ (2*k+1)**2 for k in range(50) ])
166650
```

ou construire `sum` « à la main » :

```
>>> S = 0
>>> for k in range(50): S+= (2*k + 1)**2
>>> S
166650
```

Solution du test 2.16

$$1. (k+1)^3 - k^3 = k^3 + 3k^2 + 3k + 1 - k^3 = 3k^2 + 3k + 1$$

$$2. \sum_{k=1}^n k^3 + (n+1)^3 = \sum_{k=1}^{n+1} k^3 \Leftrightarrow (n+1)^3 = \sum_{k=1}^{n+1} k^3 - \sum_{k=1}^n k^3 = \sum_{k=0}^n (k+1)^3 - \sum_{k=1}^n k^3 = 1^3 + \sum_{k=0}^n ((k+1)^3 - k^3)$$

$$(n+1)^3 = 1 + 3 \sum_{k=0}^n k^2 + 3 \sum_{k=0}^n k + \sum_{k=0}^n 1 = 1 + 3S_2 + 3 \frac{n(n+1)}{2} + n + 1 \text{ en notant } S_2 = \sum_{k=0}^n k^2.$$

$$3S_2 = (n+1)^3 - (n+1) - 3 \frac{n(n+1)}{2} = (n+1) \left(n^2 + 2n + 1 - n - 1 - \frac{3}{2}n \right) = (n+1) \left(n^2 + \frac{n}{2} \right) = (n+1)n \frac{2n+1}{2}$$

$$\text{Finalement, on retrouve le résultat connu : } S_2 = \frac{n(n+1)(2n+1)}{6}.$$

Solution du test 2.17

$$P = \prod_{k=2}^n \frac{k-1}{k} = \frac{\prod_{k=2}^n (k-1)}{\prod_{k=2}^n k} = \frac{\prod_{k=1}^{n-1} k}{\prod_{k=2}^n k} = \frac{1 \times \prod_{k=2}^{n-1} k}{n \times \prod_{k=2}^{n-1} k} = \frac{1}{n}$$

Pour le calcul de S , il suffit de remarquer que $S = \ln(P)$.

Solution du test 2.18

$$2^{2k-1} = \frac{1}{2} \times 4^k \text{ donc } S_1 = \frac{1}{2} \sum_{k=1}^n 4^k = \frac{1}{2} \times \frac{4^n - 1}{4 - 1} = \frac{4^n - 1}{6}.$$

$$\frac{1}{2^{k-1}} = \frac{2}{2^k} \text{ donc } S_2 = 2 \sum_{k=1}^n \left(\frac{-1}{2} \right)^k = 2 \times \frac{\left(\frac{-1}{2} \right)^n - 1}{\frac{-1}{2} - 1} = \frac{4}{3} \left(1 - \left(\frac{-1}{2} \right)^n \right).$$

Solution du test 2.19

$$S = \sum_{j=1}^n \sum_{i=1}^j i \times j = \sum_{j=1}^n j \sum_{i=1}^j i = \sum_{j=1}^n j \times \frac{j(j+1)}{2} = \frac{1}{2} \left(\sum_{j=1}^n k^3 + \sum_{j=1}^n k^2 \right)$$

En utilisant les formules du cours, on obtient alors :

$$S = \frac{1}{2} \left(\frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6} \right) = \frac{1}{4} n(n+1) \left(\frac{2n+1}{3} + \frac{n^2+n}{2} \right) = \frac{1}{24} n(n+1)(3n^2+7n+2)$$

Finalement, après factorisation :

$$S = \frac{1}{24} n(n+1)(n+2)(3n+1)$$

Créons deux fonctions **Python** pour calculer cette somme :

```
>>> def S(n):
...     S = 0
...     for j in range(1, n+1):
...         for i in range(1, j+1):
...             S += i*j
...     return S

>>> def F(n):
...     return n*(n+1)*(n+2)*(3*n+1)/24

>>> S(100)
12920425

>>> F(100)
12920425.0
```

Solution du test 2.20

$$1. P_i = \prod_{k=1}^n (2k-1) = \frac{(2n)(2n-1)(2n-2)\dots 4 \times 3 \times 2 \times 1}{(2n)(2n-2)\dots 4 \times 2} = \frac{(2n)!}{\prod_{k=1}^n 2k} = \frac{(2n)!}{2^n \prod_{k=1}^n k} = \frac{(2n)!}{2^n n!}$$

```
2. >>> def produit(n):
...     P = 1
...     for k in range(2, n+1):
...         P *= (2*k-1)
...     return P
```

$$3. P = \frac{P_i}{\prod_{k=1}^n 2k} = \frac{P_i}{2^n \prod_{k=1}^n k} = \frac{P_i}{2^n n!} = \frac{(2n)!}{4^n (n!)^2}$$

Solution du test 2.21

$$1. \frac{a}{k-2} + \frac{b}{k} + \frac{c}{k+2} = \frac{ak(k+2) + b(k^2-4) + ck(k-2)}{k(k^2-4)} = \frac{(a+b+c)k^2 + 2(a-c)k - 4b}{k(k^2-4)}$$

$$\frac{a}{k-2} + \frac{b}{k} + \frac{c}{k+2} = \frac{1}{k(k^2-4)} \Leftrightarrow \begin{cases} a+b+c=0 \\ a-c=0 \\ -4b=1 \end{cases}$$

On obtient $b = -\frac{1}{4}$ et $a = c = \frac{1}{8}$. On en déduit que :

$$\frac{1}{k(k^2-4)} = \frac{1}{8} \left(\frac{1}{k-2} - \frac{2}{k} + \frac{1}{k+2} \right)$$

$$2. \text{ Par linéarité, } 8S = \sum_{k=3}^n \frac{1}{k-2} - 2 \sum_{k=3}^n \frac{1}{k} + \sum_{k=3}^n \frac{1}{k+2}.$$

On va réindexer la première et la troisième sommes.

$$8S = \sum_{k=1}^{n-2} \frac{1}{k} - 2 \sum_{k=3}^n \frac{1}{k} + \sum_{k=5}^{n+2} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} - \frac{2}{3} - \frac{2}{4} - \frac{2}{n-1} - \frac{2}{n} + \frac{1}{n-1} + \frac{1}{n} + \frac{1}{n+1} + \frac{1}{n+2}$$

$$S = \frac{11}{96} - \frac{1}{8} \left(\frac{1}{n-1} + \frac{1}{n} - \frac{1}{n+1} - \frac{1}{n+2} \right)$$

Solution du test 2.22

$$1. f_1(x) = (u(x))^3 \text{ avec } u(x) = x^2 + x. f_1'(x) = 3 \times u'(x) \times (u(x))^2 = 3(2x+1)(x^2+x)^2.$$

```
In [301]: f1 = lambda x : (x**2 + x)**3
```

```
In [302]: sy.diff(f1(x), x)
```

```
Out [302]:
```

$$(6 \cdot x + 3) \cdot \begin{bmatrix} 2 \\ x + x \end{bmatrix}$$

$$2. \text{ Même tactique : } f_2'(x) = 4(6x+4)(3x^2+4x-6)^3.$$

$$3. \text{ Même tactique : } f_3'(x) = -4(6x+4)(3x^2+4x-6)^{-5}.$$

$$4. \text{ Avec les notations usuelles : } f_4'(x) = -\frac{u'(x)}{(u(x))^2} = -\frac{6x-5}{(3x^2-5x+1)^2}.$$

$$5. f_5'(x) = -u'(x) \times \sin(u(x)) = -\frac{\sin(\sqrt{x})}{2\sqrt{x}}$$

$$6. f_6'(x) = u'(x) \times \frac{1}{2\sqrt{u(x)}} = -\frac{\sin(x)}{2\sqrt{\cos(x)}}$$

$$7. f_7(x) = \underbrace{\left(\cos \left(\frac{x^2+5x+1}{v(x)} \right) \right)}_{u(x)}^2 \text{ donc :}$$

$$f_7'(x) = 2u'(x) \times u(x) = -2v'(x) \sin(v(x)) \times u(x) = -(2x+5) \sin(2(x^2+5x+1)) \text{ en se rappelant que } \sin(2t) = 2 \sin(t) \cos(t).$$

$$8. f_8(x) = \cos \underbrace{\left(\frac{x^2+5x+1}{v(x)} \right)}_{u(x)}^2 \text{ donc :}$$

$$f_8'(x) = -u'(x) \sin(u(x)) = -2v'(x)v(x) \sin(u(x)) = -2(2x+5)(x^2+5x+1) \sin((x^2+5x+1)^2).$$

$$9. f_9(x) = \frac{1}{f_8(x)} \text{ donc } f_9'(x) = -\frac{f_8'(x)}{f_8^2(x)} = \frac{2(2x+5)(x^2+5x+1)\sin((x^2+5x+1)^2)}{\cos^2((x^2+5x+1)^2)}.$$

$$10. f_{10}(x) = \frac{\sin(x)}{\cos(x)} \text{ donc } f_{10}'(x) = \frac{\cos^2(x) + \sin^2(x)}{\cos^2(x)} = 1 + \tan^2(x) = \frac{1}{\cos^2(x)}.$$

$$11. f_{11}(x) = f_{10}(\sqrt{x}) \text{ donc } f_{11}'(x) = \frac{1}{2\sqrt{x}} \times f_{10}'(\sqrt{x}) = \frac{1 + \tan^2(\sqrt{x})}{2\sqrt{x}}.$$

$$12. f_{12}'(x) = \frac{1 + \frac{2x+2}{2\sqrt{x^2+2x+1}}}{2\sqrt{x+\sqrt{x^2+2x+1}}} = \frac{\sqrt{x^2+2x+1} + x+1}{2\sqrt{x^2+2x+1+x\sqrt{x^2+2x+1}}}$$

$$13. f_{13}(x) = \sqrt{x^2 - 4^2} = x - 16 \text{ donc } f_{13}'(x) = 1 \dots$$

$$14. f_{14}'(x) = \frac{2\cos(2x)\cos(3x) + 3\sin(3x)\sin(2x)}{\cos^2(3x)}$$

$$15. f_{15}(x) = x^{\frac{1}{8}} \text{ donc } f_{15}'(x) = \frac{1}{8} x^{-\frac{7}{8}}.$$

$$16. f_{16}(x) = \ln(\ln(\ln(\ln(x)))) \text{ donc } f_{16}'(x) = \underbrace{u'}_{\underbrace{v(x)}_{\underbrace{u(x)}}} \cdot \frac{1}{u} = v' \cdot \frac{1}{v} \cdot \frac{1}{u} = \omega' \cdot \frac{1}{\omega} \cdot \frac{1}{v} \cdot \frac{1}{u}.$$

$$\text{Finalement } f_{16}'(x) = \frac{1}{x \ln(x) \ln(\ln(x)) \ln(\ln(\ln(x)))}.$$

Solution du test 2.23

1. Faux ! Considérez $f : x \mapsto 32 - 1/x$.

2. Vrai ! C'est dans le cours.

3. Faux ! C'est aussi dans le cours $f : x \mapsto |x|$ en 0.

4. Faux ! Considérez par exemple $f : x \mapsto x^2 \sin(1/x)$ si $x \neq 0$ et $f(0) = 0$. Elle est dérivable en 0 mais sa dérivée n'est pas continue en 0. Démonstrons-le ! $\frac{f(x) - f(0)}{x - 0} = x \sin(1/x)$ qu'on peut encadrer entre $-x$ et x donc la limite de ce rapport est zéro d'après le théorème des gendarmes et f est bien dérivable en 0 avec $f'(0) = 0$.

D'autre part, pour tout réel x non nul :

$$f'(x) = 2x \sin(1/x) + x^2(-1/x^2) \cos(1/x) = 2x \sin(1/x) - \cos(1/x)$$

Or \cos n'admet pas de limite en l'infini (cf. démonstration 15 de la page 61).

Donc f' n'est pas continue en zéro.

5. Faux ! Considérez $x \mapsto \ln x$.

6. Faux ! Voir le cours.

7. (a) Faux ! Considérez une fonction constante sur \mathbb{R} . Elle est en particulier croissante (au sens large) et pourtant $f(32) \leq f(3)$ alors que $32 > 3$.

(b) Vrai ! Considérez la contraposée¹. La contraposée, « si $x > y$ alors $f(x) \geq f(y)$ », est vraie donc la proposition directe l'est aussi.

1. Rappel : la contraposée de $A \Rightarrow B$ est *contraire* de $A \Rightarrow$ *contraire* de B , cf. chapitre 1, section 1.1.

8. C'est vrai. Supposons par exemple que f est croissante sur $[0; 1]$, alors $f(0) \leq f(x) \leq f(1)$ pour tout $x \in E$.
9. C'est faux : il suffit de chercher une fonction qui tende vers zéro tout en oscillant énormément. $x \mapsto \frac{1}{x} \sin(x^3)$ devrait convenir, sa dérivée valant $-\frac{\sin(x^3)}{x^2} + 3x \cos(x^3)$, qui n'admet pas de limite en $+\infty$ (cf. question 15 page 61).
- Comment en sommes-nous arrivés là ? Le $\sin(x^3)$ nous donne quelque chose qui oscille *énormément*, le $1/x$ permet de faire converger² f en $+\infty$, mais n'est pas assez « puissant » pour compenser l'*énorme* oscillation en $+\infty$.
10. (a) C'est vrai : dérivez la relation $\forall x \in \mathbb{R} \quad f(x+T) = f(x)$, T désignant la période de f .
- (b) C'est faux : $x \mapsto \sin x + x$ est non périodique mais de dérivée périodique.
- (c) C'est vrai : dérivez la relation $f(x) = f(-x)$
- (d) C'est vrai : dérivez la relation $f(x) = -f(-x)$
- (e) C'est vrai : si $-f'(-x) = f'(x)$ alors $f(-x) = f(x) + k$. En appliquant cette égalité pour $x = 0$ nous trouvons que k est nul et donc que f est paire.
- (f) C'est faux : la fonction $x \mapsto 1 + x^3$ n'est ni paire ni impaire, mais sa dérivée est paire.

Solution du test 2.24

1. $e^x e^{-x} = e^{x-x} = 1$
2. $e^x e^{-x+1} = e^{x-x+1} = e^1 = e$
3. $ee^{-x} = e^{1-x}$
4. $(e^{-x})^2 = e^{-2x}$
5. $e^{2x}/e^{2-x} = e^{2x-2+x} = e^{3x-2}$
6. $(e^x)^3/e^{2x} = e^{3x-2x} = e^x$
7. $e^x(e^x + e^{-x}) = e^{2x} + 1$
8. $(e^x)^5(e^{-2x})^2 = e^{5x}e^{-4x} = e^{5x-4x} = e^x$
9. $e^{-3x+1}(e^x)^3 = e^{-3x+1+3x} = e$
10. $\sqrt{e^{-2x}} = e^{-2x/2} = e^{-x}$
11. $\frac{e^{-4x}e}{(e^{-x})^2} = e^{-4x+1+2x} = e^{-2x+1}$
12. $(e^x + e^{-x})^2 - (e^x - e^{-x})^2 = (e^x + e^{-x} + e^x - e^{-x})(e^x + e^{-x} - e^x + e^{-x}) = (2e^x)(2e^{-x}) = 4e^{x-x} = 4$
13. $(e^x - e^{-x})^2 - e^{-x}(e^{3x} - e^{-x}) = e^{2x} - 2e^{x-x} + e^{-2x} - e^{3x-x} + e^{-x-x} = 2e^{-2x} - 2$
14. $(e^x - e^{-x})(e^{2x} + e^x + 1) = e^{3x} + e^{2x} + e^x - e^x - 1 - e^{-x} = e^{3x} + e^{2x} - 1 - e^{-x}$

Solution du test 2.25

1. f_1 est définie sur \mathbb{R} .
- $\lim_{x \rightarrow -\infty} x^2 + 1 = +\infty$ et $\lim_{x \rightarrow -\infty} e^x = 0$ donc $\lim_{x \rightarrow -\infty} f_1(x) = +\infty$
- $f_1(x) = e^x(1 + x^2e^{-x} + e^{-x})$ donc $\lim_{x \rightarrow +\infty} f_1(x) = +\infty$
- $f_1'(x) = e^x + 2x$.

On peut vérifier avec **Python** :

```
In [11]: f1 = lambda x : sy.exp(x) + x**2 + 1
In [12]: sy.limit(f1(x), x, sy.oo)
Out [12]: ∞
In [13]: sy.limit(f1(x), x, -sy.oo)
Out [13]: ∞
In [14]: sy.diff(f1(x), x)
Out [14]:
      x
2.x + e
```

2. f_2 est définie sur \mathbb{R} .

$$\lim_{x \rightarrow +\infty} f_2(x) = +\infty$$

$$\lim_{x \rightarrow -\infty} x e^x = 0 \text{ donc } \lim_{x \rightarrow -\infty} f_2(x) = 0$$

$$f_2'(x) = 5e^x + 5e^x + 5xe^x = 5e^x(2+x)$$

3. f_3 est définie sur \mathbb{R}^* et est impaire :

$$f_3(-x) = \frac{\frac{1}{e^x} + 1}{\frac{1}{e^x} - 1} = \frac{1 + e^x}{1 - e^x} = -f_3(x)$$

$$\lim_{\substack{x \rightarrow 0 \\ x > 0}} e^x - 1 = 0^+ \text{ et } \lim_{\substack{x \rightarrow 0 \\ x < 0}} e^x + 1 = 2 \text{ donc } \lim_{\substack{x \rightarrow 0 \\ x > 0}} f_3(x) = +\infty.$$

$$\text{D'autre part, } f_3(x) = \frac{e^x(1+e^{-x})}{e^x(1-e^{-x})} = \frac{1+e^{-x}}{1-e^{-x}} \text{ et } \lim_{x \rightarrow +\infty} 1 \pm e^{-x} = 1 \text{ donc } \lim_{x \rightarrow +\infty} f_3(x) = 1.$$

$$f_3'(x) = \frac{e^x(e^x - 1) - e^x(e^x + 1)}{(e^x - 1)^2} = -2 \frac{e^x}{(e^x - 1)^2}$$

Sur **Python**, pour la limite à droite on précise « dir='+' » comme troisième argument et « dir='- ' » pour la limite à gauche :

```
In [31]: f3 = lambda x : (sy.exp(x) + 1) / (sy.exp(x) - 1)
In [32]: sy.limit(f3(x), x, sy.oo)
Out [32]: 1
In [33]: sy.limit(f3(x), x, 0, '+')
Out [33]: ∞
In [34]: sy.factor(sy.diff(f3(x), x))
Out [34]:
      x
-2.e
-----
      2
[ x ]
[e - 1]
```

4. f_4 est définie sur \mathbb{R} .

$f_4(x) = (3x+1)e^{-x} - 1$ donc $\lim_{x \rightarrow -\infty} f_4(x) = -\infty$ et $\lim_{x \rightarrow +\infty} f_4(x) = -1$.

$$f_4'(x) = 3e^{-x} - (3x+1)e^{-x} = (-3x+2)e^{-x}$$

Sur Python :

```
In [41]: f4 = lambda x : (3*x + 1 - sy.exp(x)) / sy.exp(x)
In [42]: sy.limit(f4(x), x, -sy.oo)
Out [42]: -∞
In [43]: sy.limit(f4(x), x, sy.oo)
Out [43]: -1
In [44]: sy.factor( sy.diff(f4(x), x) )
Out [44]:
```

$$-(3.x - 2).e^{-x}$$

5. f_5 est définie sur \mathbb{R} .

$$\lim_{x \rightarrow -\infty} f_5(x) = +\infty$$

$$\lim_{x \rightarrow +\infty} f_5(x) = 0$$

$$f_5'(x) = 3x^2e^{-x} - x^3e^{-x} = x^2(3-x)e^{-x}$$

6. f_6 est définie sur $\mathbb{R} \setminus \{-1\}$.

```
In [61]: f6 = lambda x : (x**2 * sy.exp(x)) / (x + 1)
In [62]: sy.limit(f6(x), x, sy.oo)
Out [62]: ∞
In [63]: sy.limit(f6(x), x, -sy.oo)
Out [63]: 0
In [64]: sy.limit(f6(x), x, -1, '+')
Out [64]: ∞
In [65]: sy.limit(f6(x), x, -1, '-')
Out [65]: -∞
In [66]: sy.factor( sy.diff(f6(x), x) )
Out [66]:
```

$$\frac{x^2(x+2)e^x}{(x+1)^2}$$

7. f_7 est définie sur \mathbb{R}^* .

```
In [71]: f7 = lambda x : sy.exp(x) / x
In [72]: sy.limit(f7(x), x, sy.oo)
Out [72]: ∞
In [73]: sy.limit(f7(x), x, -sy.oo)
Out [73]: 0
```



```

In [74]: sy.limit(f7(x), x, 0, '+')
Out[74]: ∞
In [75]: sy.limit(f7(x), x, 0, '-')
Out[75]: -∞
In [76]: sy.factor( sy.diff(f7(x), x) )
Out[76]:
      x
(x - 2).e
-----
      2
      x

```

8. f_8 est définie sur \mathbb{R} . $f_8(x) = e^{-x}$.

```

In [81]: f8 = lambda x : 1 / sy.exp(x)
In [82]: sy.limit(f8(x), x, sy.oo)
Out[82]: 0
In [83]: sy.limit(f8(x), x, -sy.oo)
Out[83]: ∞
In [84]: sy.factor( sy.diff(f8(x), x) )
Out[84]:
      -x
      -e

```

9. $f_9(x) = e^{2x} + e^{-x}$. f_9 est définie sur \mathbb{R} .

```

In [91]: f9 = lambda x : sy.exp(x)**2 + 1 / sy.exp(x)
In [92]: sy.limit(f9(x), x, sy.oo)
Out[92]: ∞
In [93]: sy.limit(f9(x), x, -sy.oo)
Out[93]: ∞
In [94]: sy.factor( sy.diff(f9(x), x) )
Out[94]:
[ 3.x   ] -x
[ 2.e   - 1 ].e

```

10. $f_{10} = f_8$.

11. f_{11} est définie sur \mathbb{R} .

```

In [111]: f11 = lambda x : sy.exp(4*x + 1)
In [112]: sy.limit(f11(x), x, -sy.oo)
Out[112]: 0
In [113]: sy.limit(f11(x), x, sy.oo)
Out[113]: ∞
In [114]: sy.factor( sy.diff(f11(x), x) )
Out[114]:
      4.x
4.e.e

```

12. f_{12} est définie sur \mathbb{R} .

```
In [121]: f12 = lambda x : sy.exp(5*x**3 + 7*x + 4)
In [122]: sy.limit(f12(x), x, -sy.oo)
Out[122]: 0
In [123]: sy.limit(f12(x), x, sy.oo)
Out[123]: ∞
In [124]: sy.factor( sy.diff(f12(x), x) )
Out[124]:
```

$$\left(\begin{array}{c} 2 \\ 15.x + 7 \end{array} \right)^4 7.x^5 .e^{5.x^3 + 7.x + 4}$$

13. f_{13} est définie sur \mathbb{R} .

```
In [131]: f13 = lambda x : (x + 1)*sy.exp(-x + 1)
In [132]: sy.limit(f13(x), x, -sy.oo)
Out[132]: -∞
In [133]: sy.limit(f13(x), x, sy.oo)
Out[133]: 0
In [134]: sy.factor( sy.diff(f13(x), x) )
Out[134]:
```

$$-e^{-x} \cdot x$$

14. f_{14} est définie sur \mathbb{R}^* .

```
In [141]: f14 = lambda x : (sy.exp(2*x) - 1) / x
In [142]: sy.limit(f14(x), x, -sy.oo)
Out[142]: 0
In [143]: sy.limit(f14(x), x, sy.oo)
Out[143]: ∞
In [144]: sy.limit(f14(x), x, 0, '-')
Out[144]: 2
In [145]: sy.limit(f14(x), x, 0, '+')
Out[145]: 2
In [146]: sy.factor( sy.diff(f14(x), x) )
Out[146]:
```

$$\frac{2.x.e^{2.x} - e^{2.x} + 1}{x^2}$$

Pour la limite en 0, on remarque que $f_{14}(x) = 2 \frac{e^u - e^0}{u - 0}$ en posant $u = 2x$.

Et $\lim_{u \rightarrow 0} \frac{e^u - e^0}{u - 0} = e^0$, nombre dérivé de $u \mapsto e^u$ en 0.

15. f_{15} est définie sur \mathbb{R} .

```
In [151]: f15 = lambda x : sy.exp( sy.cos(x) )
In [152]: sy.limit(f15(x), x, -sy.oo)
Out[152]: <exp(-1), E>
In [153]: sy.limit(f15(x), x, sy.oo)
Out[153]: <exp(-1), E>
In [154]: sy.factor( sy.diff(f15(x), x) )
Out[154]:
      cos(x)
    -e      .sin(x)
```

Il y a en effet un problème en l'infini car la fonction cos n'admet pas de limite.

Pour le démontrer, remarquons d'abord que, pour tout réel x , $\cos(x) \in [-1, 1]$ donc $f_{15}(x) \in [e^{-1}, e^1]$ car la fonction exponentielle est strictement croissante sur $[-1, 1]$.

Si la fonction admet une limite en l'infini, celle-ci est finie. Notons-la ℓ . Cela signifie que l'on peut rendre $f(x)$ aussi proche que l'on veut de ℓ .

En d'autres termes, pour tout réel strictement positif ε , il existe un réel x_ε , tel que pour tout réel x supérieur à x_ε , la distance de $f(x)$ à ℓ soit inférieure à ε .

De manière plus formelle :

$$(\forall \varepsilon > 0)(\exists x_\varepsilon \in \mathbb{R})(x > x_\varepsilon \Rightarrow |f(x) - \ell| < \varepsilon)$$

Or, pour tout entier n , $f(2\pi n + \pi/2) = e^{\cos(2\pi n + \pi/2)} = 0$, en particulier pour $n = \lfloor x_\varepsilon \rfloor + 1$.

On en déduit que pour tout ε strictement positif, $|\ell| < \varepsilon$ donc $\ell = 0$ (vous préciserez ce point en étudiant la notion de *borne inférieure* l'an prochain).

Ainsi, si une limite existe, alors elle vaut 0.

Or pour tout entier n , $f(2\pi n) = e^1 = e$.

$$(\exists \varepsilon/2 > 0)(\forall x_\varepsilon \in \mathbb{R})(x = 2\pi(\lfloor x_\varepsilon \rfloor + 1) > x_\varepsilon \text{ et } |f(x) - \ell| = \varepsilon \geq \varepsilon/2)$$

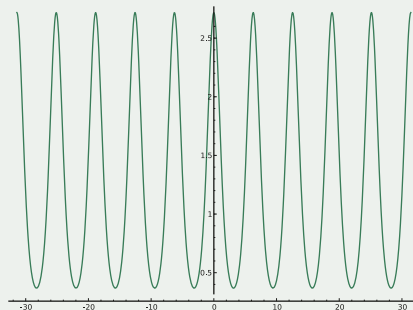
La convergence de f vers 0 est donc contredite (en effet $\neg(A \rightarrow B) \equiv A \wedge (\neg B)$ comme étudié dans la section 1.1 du chapitre 1, sur la logique).

Or si f admet une limite en l'infini, elle vaut 0. On en déduit que f n'admet pas de limite en l'infini.

Retenez bien qu'une fonction peut admettre en l'infini ou en une valeur finie une limite (finie ou infinie) ou pas de limite du tout.

Regardons l'allure de la courbe sur l'intervalle $[-10\pi, 10\pi]$:

```
In [155]: sy.plot(f15(x), (x, -5*2*sy.pi, 5*2*sy.pi))
```



Solution du test 2.26

1. La masse d'un nouveau-né : $W(0) = 2\,600 \times 0,49^3 = 305,8874$.

$$W'(t) = 3 \times 2\,600 \times 0,51 \times 0,075 e^{-0,075t} (1 - 0,51 e^{-0,075t})^2 = 298,35 e^{-0,075t} (1 - 0,51 e^{-0,075t})^2$$

Le taux de croissance d'un nouveau-né : $W'(0) = 298,35 \times 0,49^2 = 71,633835$.

```
In [11]: t, x = sy.symbols('t x')
In [12]: W = lambda t : 2600 * (1 - 0.51*sy.exp(-0.075*t))**3
In [13]: W(0)
Out [13]: 305.887400000000
In [14]: Wp = lambda x : sy.diff(W(t), t).subs(t, x)
In [15]: Wp(x)
Out [15]:
          2
298.35 * [ 1 - 0.51 * e-0.075 * x ] * e-0.075 * x
In [16]: Wp(0)
Out [16]: 71.6338350000000
```

2. On étudie le signe de $W'(t)$ sur \mathbb{R}_+ qui est positif comme produit de nombres positifs.

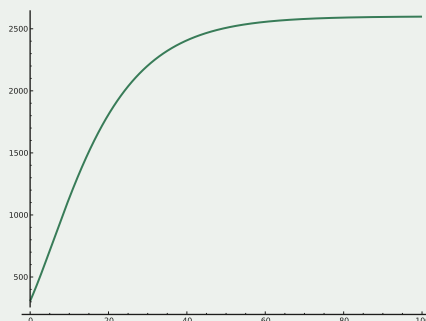
$$\lim_{t \rightarrow +\infty} -0,075t = -\infty \text{ et } \lim_{t \rightarrow -\infty} e^x = 0 \text{ donc } \lim_{t \rightarrow +\infty} e^{-0,075t} = 0.$$

On en déduit que $\lim_{t \rightarrow +\infty} W(t) = 2\,600$.

Les éléphants grossissent chaque année mais leur masse se stabilise vers 2,6 tonnes.

t	0	$+\infty$
Signe de $W'(t)$	+	
Variations de W	305,8874	2 600

```
In [21] sy.plot(W(t), (t, 0, 100))
```



3. On va étudier la fonction W' sur \mathbb{R}_+ .

$$W''(t) = 298,35 \times 0,075 \times e^{-0,075t} (1 - 0,51 e^{-0,075t})(1,53 e^{-0,075t} - 1), \text{ ce qui n'est pas très sympathique.}$$

Cependant, $W''(t)$ est du signe de $(1 - 0,51 e^{-0,075t})(1,53 e^{-0,075t} - 1)$.

Posons $x = 0,51e^{-0,075t}$ et étudions le signe de $P(x) = (1-x)(3x-1)$. Comme $t \mapsto 0,51e^{-0,075t}$ est décroissante et à valeurs dans $[0; 0,51]$, on étudie le signe de $P(x)$ sur cet intervalle :

x	$0 (t \rightarrow +\infty)$	$\frac{1}{3}$	$0,51 (t = 0)$
Signe de $(1-x)(3x-1)$	-	0	+

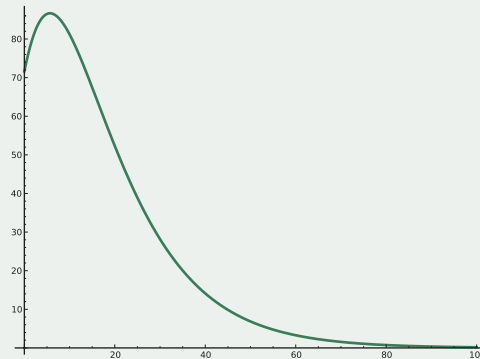
Or $0,51e^{-0,075t} = \frac{1}{3} \Leftrightarrow t = \frac{1}{0,075} \ln(1,53) \approx 5,67$ donc :

t	0	$\approx 5,67$	$+\infty$
Signe de $W''(t)$	+	0	-

puis :

t	0	$\approx 5,67$	$+\infty$
Signe de $W''(t)$	+	0	-
Variations de W'	$\nearrow \approx 86,7$ $71,633835 \quad \searrow 0$		

```
In [31]: sy.plot(Wp(t), (t, 0, 100))
```



Solution du test 2.27

Étudions la fonction f sur \mathbb{R}_+ .

$$f'(v) = 2cv e^{-mv^2/(2kT)} - \frac{mc v^3}{kT} e^{-mv^2/(2kT)} = \left(2 - \frac{mv^2}{kT}\right) c v e^{-mv^2/(2kT)}$$

Tous les coefficients étant positifs, $f'(v)$ est du signe de $2 - \frac{mv^2}{kT}$.

$$2 - \frac{mv^2}{kT} > 0 \Leftrightarrow v^2 < 2kT/m$$

On en déduit le tableau suivant :

v	0	$\sqrt{2kT/m}$	$+\infty$
Signe de $f''(v)$	+	0	-
Variations de f	$ \begin{array}{c} \nearrow 2kTc/me \searrow \\ \hline \end{array} $		

```

In [351]: c, v, m, k, T = sy.symbols('c v m k T')
In [352]: f = lambda v : c * v**2 * sy.exp(-m*v**2 / (2*k*T))
In [353]: f(v)
Out [353]:
      2
    -m.v
  -----
    2  2.T.k
  c.v .e
In [354]: sy.factor( sy.diff(f(v), v) )
Out [354]:
      2
    -m.v
  -----
    2  2.T.k
  -c.v. [ -2.T.k + m.v ].e
  -----
      T.k
In [355]: sy.solve( sy.factor( sy.diff(f(v), v) ), v )
Out [355]:
[
  [
    0, -sqrt(2).sqrt(T.k/m), sqrt(2).sqrt(T.k/m)
  ]
]

```

Solution du test 2.28

(E) $\Leftrightarrow x^x \ln x = x^2 \ln x$, en passant au logarithme népérien, car la fonction \ln est bijective de $]0, +\infty[$ sur \mathbb{R} . Donc 1 est une solution de (E). Les autres solutions éventuelles sont les solutions de l'équation (E') : $x^x = x^2$.

(E') $\Leftrightarrow x \ln x = 2 \ln x$ car la fonction \ln est bijective, d'où 2 est la solution de (E') distincte de 1.

Les solutions de (E) sont donc 1 et 2.

Sur Python :

```
In [361]: f = lambda x : x**x * sy.ln(x) - x**2 * sy.ln(x)

In [362]: f(x)
Out [362]:
      2      x
- x .log(x) + x .log(x)

In [363]: sy.solve(f(x), x)
Out [363]: [1, 2]
```

Solution du test 2.29

1. $\text{ch}'(x) = \text{sh}(x)$, $\text{sh}'(x) = \text{ch}(x)$ et $\text{th}'(x) = 1 - \text{th}^2(x) = 1/\text{ch}^2(x)$.

2. On obtient les tableaux de variations suivants :

x	$-\infty$	$+\infty$
$\text{th } x$	-1	$+1$

x	$-\infty$	$+\infty$
$\text{sh } x$	$-\infty$	$+\infty$

x	$-\infty$	0	$+\infty$
$\text{ch } x$	$+\infty$	1	$+\infty$

$$3. \text{th}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

4. On développe et il ne reste que les doubles produits, ce qui donne le résultat.

5. On calcule séparément chaque membre en revenant aux formes exponentielles.

6. On écrit le membre de gauche avec des ch et des sh et on utilise la question précédente. C'est encore un entraînement de calcul algébrique, mais il faut s'entraîner pour calculer vite et bien...

7. D'après la question 5, on a $\text{ch}(2\alpha) = \text{ch}^2(\alpha) + \text{sh}^2(\alpha) = \text{ch}^2(\alpha)(1 + \text{th}^2(\alpha))$. Or, d'après le calcul de la dérivée de th de la question 1, on a $\text{ch}^2(\alpha) = \frac{1}{1 - \text{th}^2(\alpha)}$ donc finalement $\text{ch}(2\alpha) = \frac{1 + \text{th}^2(\alpha)}{1 - \text{th}^2(\alpha)}$. On pose $\alpha = x/2$ et on obtient le résultat. On procède de même pour $\text{sh}(x)$.

8. On écrit cette équation avec des exponentielles :

$$e^x + 9e^{-x} = 6$$

et on obtient un équation du second degré en e^x :

$$(e^x)^2 - 6e^x + 9 = 0 = (e^x - 3)^2$$

On obtient :

$$e^x = 3 \Leftrightarrow x = \ln 3 \text{ d'où } x \simeq 1,099$$

```
In [8]: sy.solve(5*sy.cosh(x) - 4*sy.sinh(x) - 3, x)
Out [8]: [log(3)]
```

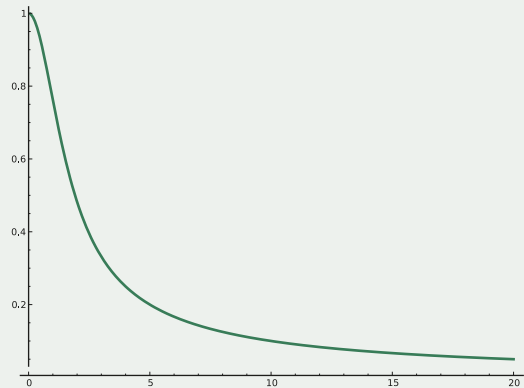
9.
$$\frac{2\cos(x)\operatorname{sh}(x)+2\sin(x)\operatorname{ch}(x)}{(\sin(x)+\operatorname{sh}(x))^2} - \frac{4\sin(x)\operatorname{sh}(x)(\cos(x)+\operatorname{ch}(x))}{(\sin(x)+\operatorname{sh}(x))^3}$$

(a) Comme souvent pour ce genre de problème, étudions la fonction $g : y \mapsto \operatorname{sh} y - y$. Sachant que $g'(y) = \operatorname{ch} y - 1 \geq 0$ sur \mathbb{R} et ne s'annule qu'une seule fois, alors g est continue, strictement croissante et change de signe pour $y = 0$: nous en déduisons que $g(y)$ est du signe de y si $y \neq 0$, et que $g(0) = 0$ et nous pouvons conclure.

(b) La formule se montre en développant et réduisant le second membre.

Nous obtenons ensuite $f'(x) = \frac{x}{x^2 \operatorname{ch}^2 x} - \frac{\operatorname{sh} x}{x^2 \operatorname{ch} x} = \frac{1}{2x^2 \operatorname{ch}^2 x} ((2x - \operatorname{sh}(2x)))$. Utilisons alors le résultat précédent pour en déduire que $f'(x)$ est du signe opposé à celui de x . La fonction f est donc strictement décroissante sur $]0; +\infty[$. De plus $\lim_{x \rightarrow 0} f(x) = 1$ et $\lim_{x \rightarrow +\infty} f(x) = 0$.

```
In [9]: sy.plot(sy.tanh(x) / x, (x, 0, 20))
```



Même si elles ne constituent qu'un cas particulier des fonctions numériques, les suites méritent une étude à part entière car elles jouent un rôle extrêmement important à la fois en mathématiques et en physique. Elles jouent en effet dans les deux cas le rôle de modélisation du « discret ». Nous découvrirons donc la notion de limite de suite sous les angles physique et mathématique, puis nous parlerons d'applications importantes : en particulier les suites vérifiant une relation du type $u_{n+1} = f(u_n)$ et les suites adjacentes.

3.1 Convergence d'une suite

Qu'est-ce qu'une suite ?

Définition

Pour faire court, on pourrait se contenter de dire :

Définition : suite numérique

Une suite numérique réelle est une fonction de \mathbb{N} vers \mathbb{R} .

Mais comme pour les fonctions numériques quelconques, une suite peut être définie uniquement sur une partie de \mathbb{N} .

Définition : suite numérique définie sur un ensemble I

Soit I un ensemble d'entiers consécutifs, fini ou infini.

Une suite numérique définie sur I est une fonction de I dans \mathbb{R} , notée $(u_n)_{n \in I}$.

On identifie ainsi la fonction et les valeurs qu'elle prend, ce qui se comprend car on peut énumérer ces valeurs.



Lorsque cela est possible, on peut définir une suite en donnant son **terme général**, par exemple $u_n = \sqrt{n-2}$.

Si I est fini, la suite est dite finie. Dans ce qui suit, I est infini. Notons h le plus petit élément de I, alors $I = \{h, h+1, \dots\}$. Dans ce cas, u_n existe si, et seulement si, $n \geq h$. On appelle u_h le premier terme de la suite, u_{h+1} le deuxième, etc.

Considérons par exemple la suite définie par $u_n = \sqrt{n-2}$. Cette suite est définie sur $I = \llbracket 2; +\infty \rrbracket$. Son premier terme est $u_2 = \sqrt{2-2} = 0$.

Interprétation physique

Enfilons une blouse : et hop ! Nous voici devenus physiciens (enfin presque...).

Laissons tomber un objet dans un tube où nous avons au préalable fait le vide. Notons h_n la distance parcourue par l'objet au bout de n secondes :

$$h_0 = 0 \text{ m}, h_1 = 4,9 \text{ m}, h_2 = 19,6 \text{ m}, h_3 = 44,1 \text{ m}, h_4 = 78,4 \text{ m}$$

Avec un bon sens de l'observation, nous remarquons que :

$$h_n \approx \frac{1}{2} g n^2$$

avec $g \approx 9,81 \cdot \text{ms}^{-2}$ et n le *rang* de la mesure, correspondant ici au nombre de secondes écoulées depuis le début de la chute.

Nous avons ainsi tout naturellement construit une *suite* de mesures qui est en fait une suite numérique de *terme général* $h_n = 4,905n^2$.

Aurait-il été plus simple de noter $h(t) = 4,905t^2$ avec t le temps en secondes et $h(t)$ la hauteur en mètres ?

Attention ! Nous avons pris des mesures chaque seconde. Rien ne nous dit qu'entre chaque mesure il ne se passe pas des choses extrêmement bizarres. Bien sûr le physicien généralisera le résultat à n'importe quelle valeur de t , entière ou non, car il a en poche des lois qui le lui permettent : il passe naturellement du *discret* au *continu*.



Ce passage du discret au continu est l'un des points forts de votre formation : nous en reparlerons tout au long de l'ouvrage, notamment au moment de la découverte du calcul intégral, au chapitre 4.

Hors d'un contexte physique, un mathématicien aura besoin d'être convaincu avant de pouvoir généraliser.

Considérez par exemple la suite de terme général :

$$u_n = \sin(n\pi)$$

En fait, u_n est toujours nul pour tout entier naturel n .

Considérez maintenant la fonction définie pour tout réel t par $f(t) = \sin(t\pi)$...

Par exemple $f(1/2) = \sin(\pi/2) = 1$ donc la fonction f n'est pas nulle partout.

Imaginez un physicien prenant chaque seconde des mesures d'un phénomène obéissant à cette loi, par exemple, une onde se propageant à la surface de l'eau : il pourrait conclure qu'après avoir jeté un caillou dans l'eau, la surface reste immobile...

Pour en revenir à nos moutons, une suite numérique peut apparaître comme une « suite de mesures » à intervalles de temps réguliers : garder cette image en tête pourra peut-être vous aider à mieux appréhender l'étude des suites, et l'étude des suites devrait elle-même vous aider à appréhender les propriétés des fonctions définies sur \mathbb{R} .

Comment traduire qu'une suite est bornée ?

Definition : suite bornée

On dit qu'une suite $(u_n)_{n \in \mathbb{N}}$ est bornée si, et seulement si :

$$(\exists(m, M) \in \mathbb{R}^2)(\forall n \in \mathbb{N})(m \leq u_n \leq M)$$

Remarque

Cela revient à dire que :

$$(\exists M \in \mathbb{R}_+)(\forall n \in \mathbb{N})(|u_n| \leq M)$$

Sens de variation d'une suite**Définition**

On dit que la suite $(u_n)_{n \geq 0}$ est :

- strictement croissante si pour tout $n \in \mathbb{N}$, $u_{n+1} > u_n$;
- strictement décroissante si pour tout $n \in \mathbb{N}$, $u_{n+1} < u_n$.

Exemple

On pose, pour tout $n \in \mathbb{N}$, $u_n = (2n+1)^2$. Pour étudier les variations de $(u_n)_{n \geq 0}$, on calcule $u_{n+1} - u_n$:

$$\begin{aligned} u_{n+1} - u_n &= (2(n+1)+1)^2 - (2n+1)^2 \\ &= (2n+3)^2 - (2n+1)^2 \\ &= 4n^2 + 12n + 9 - (4n^2 + 4n + 1) \\ &= 8n + 8 > 0, \text{ pour } n \in \mathbb{N} \end{aligned}$$

Donc la suite $(u_n)_{n \geq 0}$ est strictement croissante.

Exemple

On considère la suite $(v_n)_{n \geq 0}$ définie par récurrence par :

$$\begin{cases} v_0 = 10 \\ v_{n+1} = (v_n)^2 + 3v_n + 1 \end{cases}$$

Pour étudier les variations de (v_n) , on va calculer $v_{n+1} - v_n$:

$$\begin{aligned} v_{n+1} - v_n &= (v_n)^2 + 3v_n + 1 - v_n \\ &= (v_n)^2 + 2v_n + 1 \\ &= (v_n + 1)^2 > 0, \text{ pour } n \in \mathbb{N} \end{aligned}$$

Donc la suite $(v_n)_{n \geq 0}$ est strictement croissante.

Théorème

Soit u une suite à TERMES STRICTEMENT POSITIFS. Alors :

- la suite u est strictement croissante si, et seulement si, pour tout $n \in \mathbb{N}$, on a $\frac{u_{n+1}}{u_n} > 1$;
- la suite u est strictement décroissante si, et seulement si, pour tout $n \in \mathbb{N}$, on a $\frac{u_{n+1}}{u_n} < 1$.

Théorème

Soit f une fonction définie sur \mathbb{R}_+ et u la suite définie pour $n \in \mathbb{N}$ par $u_n = f(n)$. Alors :

- si f est croissante sur \mathbb{R}_+ alors u est croissante ;
- si f est décroissante sur \mathbb{R}_+ alors u est décroissante.



Attention ! Les réciproques des deux propositions précédentes sont fausses : u peut être croissante et f ne pas l'être sur \mathbb{R}_+ (f peut avoir des variations quelconques entre deux entiers consécutifs).

Variations autour d'un exemple

On veut étudier le sens de variation de la suite u définie par $u_n = n^3$.

- **En déterminant le signe de $u_{n+1} - u_n$**

Pour tout entier n :

$$u_{n+1} - u_n = (n+1)^3 - n^3 = 3n^2 + 3n + 1$$

Le dernier terme est strictement positif comme somme de deux termes positifs et d'un terme strictement positif.

Ainsi, pour tout entier naturel n :

$$u_{n+1} - u_n > 0$$

La suite u est donc strictement croissante.



N'oubliez pas que n est positif !

- **En utilisant une fonction auxiliaire**

La fonction $f : x \mapsto x^3$ est strictement croissante sur \mathbb{R} donc, pour tout entier naturel n :

$$n < n+1 \Rightarrow f(n) = u_n < f(n+1) = u_{n+1}$$

La suite u est donc strictement croissante.

- **En comparant $\frac{u_{n+1}}{u_n}$ à 1**

Pour tout $n \geq 1$ on a $n^3 > 0$ donc la suite est à termes strictement positifs à partir du rang 1.

Pour tout $n \geq 1$:

$$\frac{u_{n+1}}{u_n} = \left(\frac{n+1}{n}\right)^3 = \left(1 + \frac{1}{n}\right)^3 > 1$$

La suite est strictement croissante à partir du rang 1. Or $u_0 = 0 < 1 = u_1$ donc la suite u est strictement croissante.

À retenir

Quelle méthode vous paraît plus adaptée ? Une chose est sûre, la troisième demande une vérification supplémentaire, la deuxième ne concerne que certaines suites, la première est plus générale et permet d'utiliser nos outils d'étude du signe d'une expression.

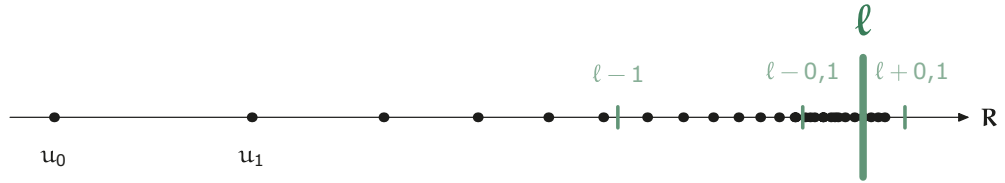
Elles sont donc classées dans l'ordre de préférence mais si vous avez une suite définie par un produit, on préférera la troisième, et si elle est définie par une somme alors on choisira la première.

Comment traduire qu'une suite converge ?

On dit qu'une suite est **convergente** si, et seulement si, elle tend vers une valeur finie.

Définition : suite convergente

On dit qu'une suite $(u_n)_{n \in \mathbb{N}}$ converge vers le réel ℓ si, et seulement si, tout intervalle ouvert contenant ℓ contient aussi **tous les termes de la suite à partir d'un certain rang** (voir figure suivante).



Prenons un exemple simple : la suite de terme général $v_n = \frac{n+1}{n}$ définie pour $n \geq 1$.

On observe que $v_1 = 2, v_2 = 3/2, v_{100} = 1,01, v_{10000} = 1,0001$: la suite semble converger vers 1.

Prenons un intervalle ouvert centré en 1 : il est de la forme $]1 - \varepsilon; 1 + \varepsilon[$ avec $\varepsilon > 0$.

Réolvons alors :

$$(I_n) : 1 - \varepsilon < v_n < 1 + \varepsilon$$

$$(I_n) \Leftrightarrow 1 - \varepsilon < 1 + 1/n < 1 + \varepsilon$$

$$(I_n) \Leftrightarrow -\varepsilon < 1/n < \varepsilon$$

$$I_n \Leftrightarrow 0 < 1/n < \varepsilon \text{ car } n \text{ est strictement positif}$$

$$(I_n) \Leftrightarrow n > 1/\varepsilon$$

Donc, quel que soit $\varepsilon > 0$, c'est-à-dire quel que soit l'intervalle ouvert centré en 1, tous les termes v_n de la suite seront dans l'intervalle dès que n sera supérieur à $1/\varepsilon$.

Définition avec des ε

$$\lim_{n \rightarrow \infty} x_n = \ell \Leftrightarrow (\forall \varepsilon > 0)(\exists N \in \mathbb{N})(\forall n \in \mathbb{N})(n \geq N \Rightarrow |x_n - \ell| < \varepsilon)$$

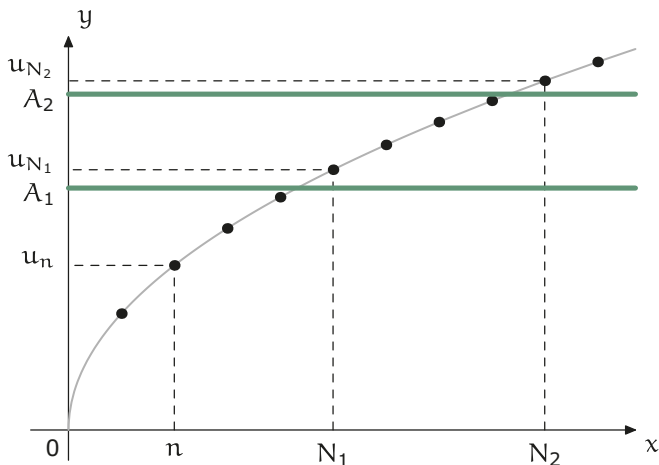
Comment traduire qu'une suite diverge vers l'infini ?

Une suite **diverge** lorsqu'elle ne tend pas vers une limite finie. Il existe plusieurs cas de divergence :

- Si la suite diverge vers l'infini (i.e. admet l'infini pour limite) alors elle est dite **divergente de première espèce**.
- Sinon (i.e. la suite n'admet pas de limite) elle est dite **divergente de seconde espèce**.

Definition : suite divergeant vers l'infini

Soit $(u_n)_{n \in \mathbb{N}}$ une suite réelle. On dit que $(u_n)_{n \in \mathbb{N}}$ diverge vers $+\infty$ si, et seulement si, pour tout réel positif A , il existe **au moins un** entier N tel que **pour tout** entier n supérieur à N , on a $u_n \geq A$ (voir figure suivante).



Reprenons par exemple le cas de l'objet tombant dans le vide : considérons donc la suite $(h_n)_{n \in \mathbb{N}}$ de terme général $h_n = 4,905n^2$.

Soit A un réel positif quelconque. Nous voudrions savoir s'il existe un rang N à partir duquel les valeurs prises par la suite $(h_n)_{n \in \mathbb{N}}$ sont *toutes* supérieures à A .

Il s'agit donc de résoudre **dans** \mathbb{N} l'inéquation :

$$(I): 5n^2 \geq A$$

$$(I) \Leftrightarrow n^2 \geq A/5$$

$$(I) \Leftrightarrow n \geq \sqrt{A/5} \text{ car } A \geq 0$$

Donc, dès que n sera supérieur à $\sqrt{A/5}$, on aura h_n supérieur à A . Donc, d'après notre définition, on peut dire que $(h_n)_{n \in \mathbb{N}}$ tend vers $+\infty$.



Mais attention ! $\sqrt{A/5}$ n'a aucune raison d'être un entier ! On peut prendre pour N le premier entier supérieur à $\sqrt{A/5}$.

— Quelques résultats utiles sur la convergence

Théorème

Si (u_n) converge, alors $\lim_{n \rightarrow +\infty} (u_{n+1} - u_n) = 0$.

Le travail sur les limites de suite correspond souvent à des manipulations d'inégalités sur des valeurs absolues, notamment les inégalités triangulaires :

- $|x + y| \leq |x| + |y|$
- $|x| - |y| \leq |x - y|$

Ici, on part d'une suite (u_n) convergeant vers un certain réel ℓ .

Soit ε un réel strictement positif quelconque. À partir d'un certain rang N dépendant de la précision ε choisie, on a, pour tout entier $n \geq N$, $|u_n - \ell| \leq \varepsilon / 2$ donc, *a fortiori*, on a aussi $|u_{n+1} - \ell| \leq \varepsilon / 2$.

On va utiliser la ruse habituelle du « ajouter/enlever » (ici ℓ) :

$$|u_{n+1} - u_n| = |u_{n+1} - \ell + \ell - u_n| \leq |u_{n+1} - \ell| + |u_n - \ell| \leq \varepsilon / 2 + \varepsilon / 2 = \varepsilon$$

Nous avons donc démontré :

$$(\forall \varepsilon > 0)(\exists N \in \mathbb{N})(n \geq N \Rightarrow |u_{n+1} - u_n| < \varepsilon)$$

qui traduit bien la convergence de $(u_{n+1} - u_n)$ vers 0.



Le théorème précédent indique que, lorsqu'une suite converge, deux termes successifs ont tendance à se « rapprocher », mais attention ! Ça ne marche pas dans l'autre sens : même si les termes se rapprochent, cela ne signifie pas que la suite converge.

Considérez par exemple la suite définie sur \mathbb{N}^* par $v_n = (\ln(n))$.

Alors $\lim_{n \rightarrow +\infty} (v_{n+1} - v_n) = 0$ et pourtant (v_n) diverge vers $+\infty$.

Application à la preuve de la divergence de la suite $((-1)^n)$

On peut utiliser la contraposée de ce théorème pour démontrer qu'une suite diverge.

Par exemple, pour la suite définie par $u_n = (-1)^n$, on a, pour tout entier n :

$$|u_{n+1} - u_n| = |-u_n - u_n| = 2|(-1)^n| = 2$$

Ainsi la suite $(u_{n+1} - u_n)$ est constante et égale à 2.

Or le théorème précédent indique que :

$$(u_n) \text{ converge} \Rightarrow (u_{n+1} - u_n) \text{ converge vers } 0$$

qui a la même valeur de vérité que sa contraposée :

$$(u_{n+1} - u_n) \text{ ne converge pas vers } 0 \Rightarrow (u_n) \text{ diverge}$$

ce qui assure donc que la suite $((-1)^n)$ diverge, or elle est bornée en valeur absolue par 1, donc elle ne peut diverger vers l'infini. C'est donc une suite divergente de seconde espèce.

Théorème

Toute suite convergente est bornée.

On part encore d'une suite (u_n) convergeant vers un certain réel ℓ .

Soit ε un réel strictement positif quelconque. À partir d'un certain rang N dépendant de la précision ε choisie, on a :

$$|u_n| - |\ell| \leq |u_n - \ell| \leq \varepsilon$$

Notons $M_2 = |\ell| + \varepsilon$ alors nous venons de démontrer que, pour tout entier $n \geq N$, on a $|u_n| \leq M_2$.

D'autre part, notons M_1 la valeur maximale prise par $|u_0|, |u_1|, \dots, |u_{N-1}|$. Cette valeur existe car il n'y a qu'un nombre fini de valeurs.

Alors, pour tout entier $0 \leq n < N$, on a $|u_n| \leq M_1$.

Finalement, on a démontré que, pour tout entier naturel n , on a $|u_n| \leq M = \max(M_1, M_2)$, ce qui traduit que la suite est bornée.



Attention ! Toute suite divergente ne diverge pas vers l'infini !

Considérez par exemple la suite définie par $u_n = (-1)^n$: elle est bornée mais nous avons démontré qu'elle divergeait.

Opérations sur les limites

En utilisant la définition des suites convergentes et les inégalités triangulaires, on obtient le tableau suivant :

Somme $\lim_{n \rightarrow +\infty} (u_n + v_n)$	$\lim_{n \rightarrow +\infty} v_n = \ell$	$\lim_{n \rightarrow +\infty} v_n = +\infty$	$\lim_{n \rightarrow +\infty} v_n = -\infty$
$\lim_{n \rightarrow +\infty} u_n = \ell'$	$\ell + \ell'$	$+\infty$	$-\infty$
$\lim_{n \rightarrow +\infty} u_n = +\infty$	$+\infty$	$+\infty$	F.I.
$\lim_{n \rightarrow +\infty} u_n = -\infty$	$-\infty$	F.I.	$-\infty$

F.I. : forme indéterminée

Pour les produits, on a le tableau suivant :

Produit $\lim_{n \rightarrow +\infty} (u_n \times v_n)$	$v_n \rightarrow \ell > 0$	$v_n \rightarrow \ell < 0$	$v_n \rightarrow 0$	$v_n \rightarrow +\infty$	$v_n \rightarrow -\infty$
$u_n \rightarrow \ell' > 0$	$\ell \ell'$	$\ell \ell'$	0	$+\infty$	$-\infty$
$u_n \rightarrow \ell' < 0$	$\ell \ell'$	$\ell \ell'$	0	$-\infty$	$+\infty$
$u_n \rightarrow 0$	0	0	0	F.I.	F.I.
$u_n \rightarrow +\infty$	$+\infty$	$-\infty$	F.I.	$+\infty$	$-\infty$
$u_n \rightarrow -\infty$	$-\infty$	$+\infty$	F.I.	$-\infty$	$+\infty$

Théorèmes de comparaisons

Théorèmes de comparaison des limites infinies

Soit u et v deux suites réelles telles que, à partir d'un certain rang, $u_n \leq v_n$.

- Si u diverge vers $+\infty$ alors v diverge vers $+\infty$.
- Si v diverge vers $-\infty$ alors u diverge vers $-\infty$.

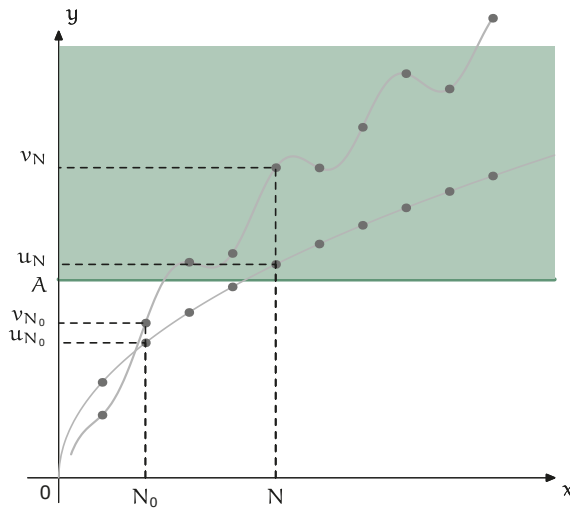
En fait, ça veut dire que si on est plus grand que quelque chose qui tend vers $+\infty$, on tend soi-même vers $+\infty$.

Maintenant, le dessin est bien beau, mais il s'agirait de démontrer ce résultat. Or nous n'avons que la définition de la limite en magasin, donc utilisons-la.

On veut prouver que $\lim_{n \rightarrow +\infty} v_n = +\infty$, donc on considère un réel positif A quelconque.

Puisque $\lim_{n \rightarrow +\infty} u_n = +\infty$, il existe un rang N à partir duquel tous les termes de la suite (u_n) sont au-dessus de A .

Il existe un entier N_0 tel que, pour tout entier $n \geq N_0$, on a $v_n \geq u_n$.



Soit $N_{\max} = \max(N_0, N)$, alors :

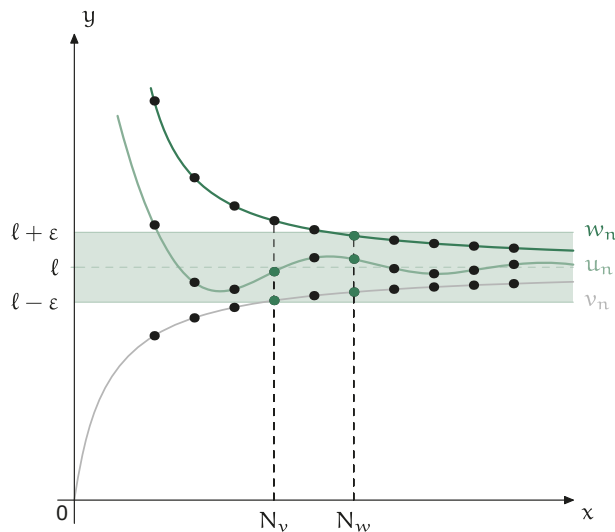
$$n \geq N_{\max} \Rightarrow v_n \geq u_n \geq A$$

ce qui traduit bien que la suite (v_n) diverge vers $+\infty$.

Théorème d'encadrement de limites (auss appelé « théorème des gendarmes »)

Soit u , v et w trois suites réelles.

- **SI** $(\exists N \in \mathbb{N})(\forall n \in \mathbb{N})(n \geq N \Rightarrow v_n \leq u_n \leq w_n)$
- **ET** v et w convergent vers une même limite $\ell \in \mathbb{R}$
- **ALORS** u converge et $\lim_{n \rightarrow +\infty} u_n = \ell$



Pour la démonstration, inspirez-vous de ce qui a été fait précédemment.

— Convergence et sens de variation

Que penser d'une suite croissante et convergeant vers un nombre ℓ ?

Considérons un cas simple.

Vos mains sont à 2 mètres de distance (oui, vous êtes un joueur du XV de France...) et vous voulez les frapper. Disons que votre main gauche reste immobile et que votre main droite s'en rapproche à 1 mètre par seconde. Calculons combien de temps il vous faudra pour que vos mains se rejoignent.

Pour cela, additionnons le temps nécessaire pour parcourir la moitié de la distance restante à chaque étape.

La main droite a 2 mètres à parcourir.

Il lui faudra 1 seconde pour parcourir la moitié de ces deux mètres, puis 1/2 seconde pour parcourir la moitié du mètre restant, puis 1/4 seconde pour parcourir le 1/4 mètre restant, puis 1/8 seconde pour parcourir le 1/8 mètre restant, etc.

Appelons T le temps nécessaire pour que vos mains se rejoignent. Alors :

$$T = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots$$

On est donc amené à additionner une infinité de nombres positifs.

Appelons T_n le temps de parcours à la n -ième étape.

Par exemple, $T_1 = 1$, $T_2 = 1 + 1/2$, $T_3 = 1 + 1/2 + 1/4$. Et plus généralement, que vaut T_n ?

$T_1 = 1$, c'est-à-dire $T_1 = 1/2^0$; $T_2 = 1/2^0 + 1/2^1$; $T_3 = 1/2^0 + 1/2^1 + 1/2^2 \dots$

$$T_n = 1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^{n-1}}$$

$$T_{n+1} - T_n = 1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n} - \left(1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^{n-1}}\right) = \frac{1}{2^n} > 0$$

donc la suite est strictement croissante. On entre donc dans le cadre qui nous intéresse.

Or nous connaissons la limite de (T_n) . En effet, $T_n = \sum_{k=0}^n \left(\frac{1}{2}\right)^k = \frac{1 - \left(\frac{1}{2}\right)^{n+1}}{1 - 1/2} = 2 - \frac{1}{2^n}$, qui

tend vers 2 quand n tend vers l'infini.

En fait ce résultat se généralise et sera admis :

Théorème des suites monotones

- Toute suite croissante et majorée converge vers ℓ , sa borne supérieure.
- Toute suite décroissante et minorée converge vers ℓ , sa borne inférieure.
- Toute suite croissante non majorée diverge vers $+\infty$.
- Toute suite décroissante non minorée diverge vers $-\infty$.

— Comment démontrer qu'une fonction n'admet pas de limite grâce aux suites ?

Nous ne verrons pas ici le théorème de caractérisation des limites de fonctions à l'aide de suites dans son ensemble, mais uniquement sa réciproque qui va nous permettre de prouver, par exemple, que la fonction cosinus n'admet pas de limite en $+\infty$.

Théorème

Soit a un élément ou une extrémité d'un intervalle I , soit f une fonction de I vers \mathbb{R} , soit (u_n) une suite d'éléments de I et soit ℓ un élément de $\mathbb{R} \cup \{-\infty, +\infty\}$.

$$\text{Si } \lim_{n \rightarrow +\infty} u_n = a \text{ et } \lim_{x \rightarrow a} f(x) = \ell \text{ alors } \lim_{n \rightarrow +\infty} f(u_n) = \ell$$

La preuve n'est qu'une application directe des définitions sur les limites mais il faut distinguer de nombreux cas selon que a et ℓ sont finis ou infinis.

Parlons maintenant cosinus. Nous allons raisonner par l'absurde en supposant que \cos admet une limite ℓ en $+\infty$. Prenons deux suites (u_n) et (v_n) judicieusement choisies qui divergent toutes deux vers $+\infty$ mais qui, composées avec \cos , induisent un comportement différent. Par exemple :

$$u_n = 2n\pi \quad v_n = 2n\pi + \pi/2$$

Si \cos admet une limite ℓ en $+\infty$, notre théorème nous assure que les suites $(\cos(u_n))$ et $(\cos(v_n))$ convergent toutes deux vers ℓ , or :

$$\lim_{n \rightarrow +\infty} \cos(2n\pi) = 1 \quad \lim_{n \rightarrow +\infty} \cos(2n\pi + \pi/2) = 0$$

Donc $0 = \ell = 1$: impossible. Notre supposition de départ est donc fautive : la fonction \cos n'admet pas de limite en $+\infty$.

Nous aurons également besoin du corollaire suivant, qui est une réécriture du théorème précédent dans le cas où a est un réel :

Corollaire

Si f est définie et continue en $a \in \mathbb{R}$ et si la suite (u_n) converge vers a , alors la suite $(f(u_n))$ converge vers $f(a)$.

3.2 Suites adjacentes

— Définition et théorème

Définition : suites adjacentes

On dit que deux suites (a_n) et (b_n) sont adjacentes lorsque :

- (a_n) est croissante,
- (b_n) est décroissante,
- $\lim_{n \rightarrow +\infty} (b_n - a_n) = 0$.

Quelles conséquences tirer de cette définition ?

Tout d'abord, comme (a_n) est croissante, alors $(-a_n)$ est décroissante et donc $(b_n - a_n)$ est décroissante. Ainsi cette suite converge vers 0 en décroissant : elle est donc positive. Par conséquent :

$$\forall n \in \mathbb{N}, a_n \leq b_n$$

On a donc, pour tout n :

$$a_n \leq a_0 \leq b_0 \leq b_n$$

La suite (a_n) est donc croissante et majorée par b_0 : elle converge vers un réel a .

La suite (b_n) est quant à elle décroissante et minorée par a_0 : elle converge vers un réel b .

Par conséquent :

$$\lim_{n \rightarrow +\infty} (a_n - b_n) = \lim_{n \rightarrow +\infty} (a_n) - \lim_{n \rightarrow +\infty} (b_n) = a - b = 0$$

donc $a = b$. Nous en déduisons le théorème suivant :

Théorème des suites adjacentes

Deux suites adjacentes sont convergentes et elles ont la même limite.



Ne tombez pas dans un piège classique : l'égalité

$$\lim_{n \rightarrow +\infty} (a_n - b_n) = \lim_{n \rightarrow +\infty} (a_n) - \lim_{n \rightarrow +\infty} (b_n)$$

n'a de sens qu'à partir du moment où on a montré que les suites (a_n) et (b_n) convergeaient.

— À quoi servent les suites adjacentes ?

Prenons un exemple. Pour calculer des valeurs approchées du nombre e , la base du logarithme népérien, on peut utiliser le résultat classique que e est la limite de la suite de terme général :

$$S_n = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} = \sum_{k=0}^n \frac{1}{k!}$$



Vous prouverez l'an prochain ce résultat.

On rappelle que $n!$ (factorielle n) est égal à $1 \times 2 \times 3 \times \cdots \times n$ et que, par convention, $0! = 1$.

La suite $(S_n)_{n \in \mathbb{N}}$ est croissante, donc $S_n \leq e$ pour tout n . Mais cela ne suffit pas à dire à quelle précision S_n est une valeur approchée de e .

Pour cela, il faudrait majorer e , et c'est là que peuvent intervenir les suites adjacentes. Car si l'on trouve une suite (T_n) telle que (S_n) et (T_n) soient adjacentes, alors elles convergeront toutes les deux vers la même limite, forcément égale à e , et on aura aussi $S_n \leq e \leq T_n$ pour tout n . Ceci permettra d'obtenir une valeur approchée de e à la précision souhaitée puisque $T_n - S_n$ tend vers 0.

Il reste à trouver une suite (T_n) convenable. D'autres s'en sont chargés avant nous et ont montré que l'on pouvait prendre :

$$T_n = S_n + \frac{1}{n \cdot n!}$$

et je vous laisse le vérifier... Ce choix de T_n est particulièrement intéressant car $T_n - S_n$ tend alors très vite vers 0, ce qui permet d'obtenir une bonne précision pour des petites valeurs de n .

Par exemple, puisque $1/(10 \cdot 10!) \approx 2,76 \times 10^{-8}$, on obtient une valeur approchée de e par défaut à $3 \cdot 10^{-8}$ près, à savoir :

$$S_{10} = \frac{9\ 864\ 101}{3\ 628\ 800} = 2,71828180\dots$$

alors que, comme vous le savez tous par cœur :

$$e \approx 2,718281828459045235360287471352662497757247093699959574966\dots$$

Ces suites permettent également de prouver que le nombre e est irrationnel, mais ceci est une autre histoire.



3.3 Suites récurrentes d'ordre 1

— Étude générale

Le but de cette section n'est pas de mener une étude complète des suites récurrentes, qui sera faite en prépa, mais plutôt de proposer quelques pistes pour cerner les problèmes et pouvoir mener à bien quelques applications intéressantes.

Une relation $u_{n+1} = f(u_n)$ définit-elle toujours une suite ?

On considère une fonction f définie sur une partie D de \mathbb{R} et à valeurs réelles, ainsi qu'un élément a de D . Et la question est de savoir s'il existe une suite $(u_n)_{n \in \mathbb{N}}$ telle que :

$$u_0 = a \text{ et } \forall n \in \mathbb{N} \ u_{n+1} = f(u_n)$$

La réponse n'est pas aussi évidente qu'elle en a l'air.

Par exemple, il n'existe pas de suite $(u_n)_{n \in \mathbb{N}}$ telle que :

$$u_0 = 1 \text{ et, pour tout } n \in \mathbb{N}, u_{n+1} = -1 / \sqrt{u_n}$$

car, avec $g : \mathbb{R}_+^* \rightarrow \mathbb{R}, x \mapsto -1 / \sqrt{x}$, on a $g(1) = -1$, donc g n'est pas définie en $u_1 = g(u_0) = g(1) = -1$, donc on ne peut pas calculer u_2 et donc les termes suivants de (u_n) non plus.

Quel est le problème ? La fonction g est définie sur $]0, +\infty[$ et à valeurs dans $]-\infty, 0[$, donc l'ensemble image n'est pas inclus dans l'ensemble d'arrivée. Pire ici, les ensembles sont disjoints.

Il faudra donc travailler sur une partie de l'ensemble de définition qui soit « stable » par f , pour être sûr que tous les termes successifs de la suite puissent être « calculables ».

Définition : partie stable

Soit f une fonction définie sur une partie D de \mathbb{R} et à valeurs réelles. On dit qu'une partie S de D est **stable par f** lorsque :

$$x \in S, f(x) \in S$$

Maintenant, si a est un élément d'une partie S de D stable par f , alors on pourra définir $f(a)$, puis $f(f(a))$ puisque $f(a) \in S$, puis $f(f(f(a)))$ puisque $f(f(a)) \in S$, etc. Il suffit donc de choisir u_0 dans S .

Définition : suite récurrente

Soit f une fonction définie sur une partie D de \mathbb{R} , S une partie stable par f et un réel $a \in S$. On peut alors construire une suite (u_n) définie par :

$$\begin{cases} u_0 = a \\ \forall n \in \mathbb{N}, u_{n+1} = f(u_n) \end{cases}$$

Une telle suite ainsi définie est appelée une suite récurrente.

Quelles sont les limites possibles d'une suite récurrente ?

Un moyen de les obtenir est de « passer à la limite » l'égalité $u_{n+1} = f(u_n)$ de la façon suivante : si (u_n) converge vers ℓ , alors (u_{n+1}) aussi, ce qui nous permet de conclure que $\ell = f(\ell)$, donc ℓ est une solution de l'équation $f(x) = x$. Beaucoup de « si » sont sous-entendus !



On dit que ℓ est un point fixe de f .

- Pour que $(f(u_n))$ converge vers $f(\ell)$, il suffit que f soit continue en ℓ comme nous l'avons vu précédemment : est-ce le cas ?
- L'équation $f(x) = x$ admet-elle au moins une solution ?
- Que se passe-t-il si l'équation admet plusieurs solutions ?
- Que se passe-t-il si la limite ℓ n'est pas un point fixe de f ou si $\ell \notin S$?
- Cela donne :

Théorème : limite d'une suite récurrente convergente (admis)

Soit I un intervalle fermé de \mathbb{R} , soit f une fonction continue de I dans I , et soit (u_n) une suite d'éléments de I telle que $u_{n+1} = f(u_n)$ pour tout n . Si (u_n) est convergente, alors sa limite ℓ est un point fixe de f , c'est-à-dire vérifie $f(\ell) = \ell$.



Une suite est dite monotone si elle ne change pas de sens de variation.

Une suite est dite bornée si elle est majorée et minorée.

Il faut bien retenir que ce théorème ne permet pas de conclure qu'une suite converge. Mais si l'on sait que la suite converge, par exemple en montrant qu'elle est monotone et bornée, alors on peut avoir une idée de la limite.

Le théorème permet également de montrer que (u_n) diverge, par exemple quand f est définie et continue sur \mathbb{R} et qu'elle n'a pas de point fixe.

En fait, il faut surtout être capable d'avoir une intuition du résultat à partir d'un petit dessin. Nous verrons cela sur quelques exemples. Mais d'abord nous aurons besoin, pour assurer l'existence d'une partie stable par f ou pour étudier ses variations, de répondre à la question suivante :

Les inégalités concernant u_0 se conservent-elles ?

Commençons par étudier un exemple classique. Soit (u_n) la suite définie par :

$$\begin{cases} u_0 = -1/2 \\ u_{n+1} = \sqrt{1+u_n} \end{cases}$$

On peut commencer par déterminer les points fixes de $f : x \mapsto \sqrt{1+x}$. On montre que le seul point fixe est $\varphi = (1 + \sqrt{5})/2$ et même que :

$$\sqrt{1+x} > x \text{ si } x \in [-1, \varphi[$$

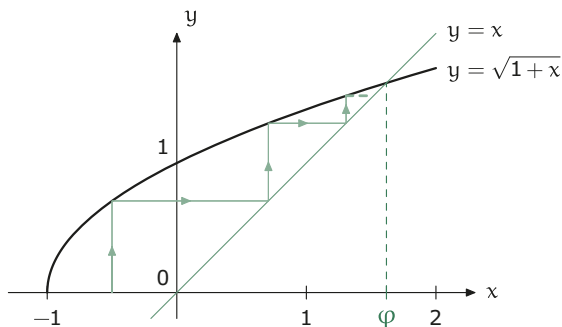
$$\sqrt{1+\varphi} = \varphi$$

$$\sqrt{1+x} < x \text{ si } x \in]\varphi, +\infty[$$



φ est plus connu sous le nom de *nombre d'or*.

On peut alors placer les premiers termes de la suite (voir figure suivante).



L'inégalité $u_n \leq \varphi$, avec $\varphi = (1 + \sqrt{5})/2$, se conserve.

En effet, si $u_n \leq \varphi$ pour un certain n , alors $\sqrt{1+u_n} \leq \sqrt{1+\varphi}$, c'est-à-dire $u_{n+1} \leq \varphi$. On a donc montré que $u_n \leq \varphi$ pour tout n .

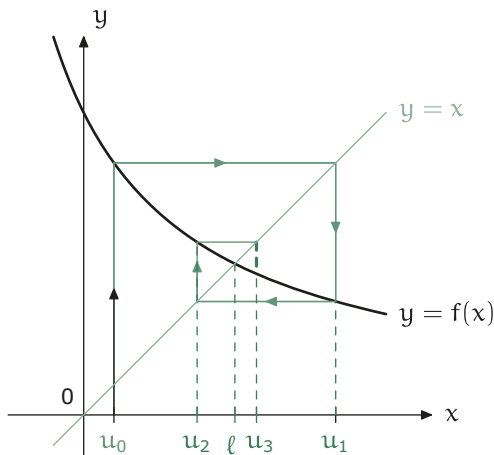
Ceci prouve que l'intervalle $[-1, \varphi]$ est stable par f . Donc, puisque nous avons montré que $\sqrt{x} \geq x$ pour tout $x \in [-1, \varphi]$ d'après l'étude de la fonction φ , nous en déduisons que $u_{n+1} - u_n = \sqrt{1+u_n} - u_n \geq 0$ et donc que (u_n) est croissante et majorée par φ . Elle est donc convergente. Enfin, la fonction f étant continue sur $[-1, \varphi]$, la suite converge vers un point fixe de f , à savoir ici φ .

On en déduit que $\lim_{n \rightarrow +\infty} u_n = \varphi$.

Plus généralement, on peut aussi parfois déterminer le sens de variation de la suite (u_n) par « conservation d'inégalités ». Supposons par exemple que f soit croissante. Alors la position de u_0 par rapport à u_1 va déterminer le sens de variation de la suite. En effet, si $u_0 \leq u_1$, alors $f(u_0) \leq f(u_1)$, c'est-à-dire $u_1 \leq u_2$, etc. : on obtient donc par récurrence $u_n \leq u_{n+1}$ pour tout n ... Et de même, $u_0 \geq u_1$ entraîne que la suite (u_n) est décroissante.

Si f est décroissante, les choses se compliquent un peu. Retenez que la suite des termes pairs et impairs sont monotones et de sens de variation contraires : en effet, si par exemple on a $u_0 \leq u_1$,

on en déduit $u_1 \geq u_2$, puis $u_2 \leq u_3$, puis $u_3 \geq u_4 \dots$. On va donc avoir $u_{2n} \leq u_{2n+1}$ et $u_{2n+1} \geq u_{2n+2}$ pour tout n . Le dessin suivant en coquille d'escargot vous permettra de visualiser la situation.



Enfin, si f n'est pas monotone, le comportement de (u_n) peut même être très compliqué, « chaotique ».

En résumé

Voici quelques idées *simples* qui pourront vous guider dans votre étude d'une suite définie par une relation du type $u_{n+1} = f(u_n)$.

- Commencez par étudier la fonction f : si par chance elle est croissante sur un intervalle stable, vous pourrez prouver des inégalités intéressantes (en utilisant la conservation de l'ordre par f) et en particulier étudier le sens de variation de la suite selon que u_0 est supérieur ou inférieur à u_1 . Mais attention, dans certains cas, la suite n'est monotone qu'à partir d'un certain rang k qui ne vaut pas forcément 0.
- Vous serez le plus souvent amené à prouver qu'une suite est monotone et bornée, donc convergente.
- Pour déterminer sa limite, on utilise le théorème bien connu. N'oubliez pas de vérifier toutes les conditions d'application, à savoir que la suite est définie par une relation du type $u_{n+1} = f(u_n)$, que f est continue de I vers I , I étant un intervalle fermé, et que la suite $(u_n)_{n \in \mathbb{N}}$ est convergente.

Cas particuliers

Suites arithmétiques

Définition : suite arithmétique

Une suite (u_n) est arithmétique lorsqu'il existe un réel b tel que, pour tout $n \in \mathbb{N}$:

$$u_{n+1} = u_n + b$$

On appelle b la raison de la suite.

Les propriétés suivantes se montrent par récurrence.

Propriétés

Étant donné une suite arithmétique de raison b :

- $\forall n \in \mathbb{N}, u_n = u_0 + nb$
- $\sum_{k=0}^n u_k = \frac{\text{nombre de termes} \times (\text{1}^{\text{er}} \text{ terme} + \text{dernier terme})}{2} = \frac{(n+1)(u_0 + u_n)}{2}$



En particulier $\sum_{k=0}^n k = \frac{n(n+1)}{2}$ pour tout $n \in \mathbb{N}$.

Suites géométriques

Définition : suite géométrique

Une suite (u_n) est géométrique lorsqu'il existe un réel $a \neq 0$ tel que, pour tout $n \in \mathbb{N}$:

$$u_{n+1} = a \cdot u_n$$

On appelle a la raison de la suite.

Les propriétés suivantes se montrent par récurrence.

Propriétés

Étant donné une suite géométrique de raison a :

- $\forall n \in \mathbb{N}, u_n = u_0 \cdot a^n$
- Si $a \neq 1$, $\sum_{k=0}^n u_k = \text{premier terme} \times \frac{1 - \text{raison}^{\text{nombre de termes}}}{1 - \text{raison}} = u_0 \frac{1 - a^{n+1}}{1 - a}$



En particulier $\sum_{k=0}^n q^k = \frac{1 - q^{n+1}}{1 - q}$ pour tout $n \in \mathbb{N}$ et tout réel $q \neq 1$.

On utilise souvent le théorème suivant :

Théorème : convergence des suites géométriques

Une suite géométrique (u_n) converge si et seulement si $|a| < 1$ ou $a = 1$

- Pour $|a| < 1$, $\lim_{n \rightarrow +\infty} u_n = 0$.
- Pour $|a| > 1$, $\lim_{n \rightarrow +\infty} |u_n| = +\infty$.
- Pour $a = -1$, la suite n'admet pas de limite.
- Pour $a = 1$, (u_n) est constante.

La clé de la démonstration vient de l'inégalité :

$$(1+x)^n \geq 1+nx$$

vraie pour tout réel positif x et tout entier naturel non nul n . Ceci nous aide à montrer le deuxième point. Pour revenir au premier, dans le cas où u_0 et a sont non nuls, on introduit la suite $(1/u_n)$ qui est également géométrique.

Suites arithmético-géométriques

Définition : suite arithmético-géométrique

Une suite (u_n) est arithmético-géométrique lorsqu'il existe deux réels a et b tels que :

$$\text{pour tout } n \in \mathbb{N}, u_{n+1} = au_n + b$$

Si $a = 1$ ou $b = 0$, on retrouve les cas précédents.

Sinon, on retrouve une situation classique, puisqu'on peut poser $u_{n+1} = f(u_n)$ avec $f(x) = ax + b$. Soit α la solution de l'équation $x = ax + b$. On obtient $\alpha = b/(1-a)$. Posons $v_n = u_n - \alpha$. Alors, après calculs, on a $v_{n+1} = av_n$ pour tout $n \in \mathbb{N}$.

Propriété : étude des suites arithmético-géométriques

Soit (u_n) une suite arithmético-géométrique. Si $a \neq 1$, la suite (v_n) définie par :

$$v_n = u_n - \frac{b}{1-a}$$

est géométrique de raison a . On peut alors en déduire u_n en fonction de n .

Remarque

Si $|a| < 1$, on montre facilement que la suite (u_n) converge vers $b/(1-a)$.

3.4 Exercices

Testez vos connaissances

Test 3.1 (Vrai ou faux ? - ♥)

1. Si une suite est convergente, alors elle est bornée.
2. Si une suite n'est pas bornée, alors elle diverge.
3. Si une suite diverge, alors elle n'est pas bornée.
4. Si $\lim_{n \rightarrow +\infty} (u_{n+1} - u_n) = 0$ alors $(u_n)_{n \in \mathbb{N}}$ est convergente.
5. La somme d'une suite divergente et d'une suite convergente est une suite divergente.
6. La somme de deux suites divergentes est divergente.
7. Si une suite d'entiers relatifs est convergente, alors elle est stationnaire (i.e. constante à partir d'un certain rang).
8. Soit (u_n) et (v_n) deux suites de réels non nuls tels que $\lim_{n \rightarrow +\infty} \frac{u_n}{v_n} = 1$, alors :

- a) $\lim_{n \rightarrow +\infty} \frac{e^{u_n}}{e^{v_n}} = 1$;
- b) $\lim_{n \rightarrow +\infty} \frac{\ln u_n}{\ln v_n} = 1$ (en supposant u_n et v_n strictement positifs pour tout entier naturel n) ;
- c) $\lim_{n \rightarrow +\infty} \frac{u_n^x}{v_n^x} = 1$ (en supposant u_n et v_n strictement positifs pour tout entier naturel n et x un réel) ;
- d) $\lim_{n \rightarrow +\infty} \frac{u_n^n}{v_n^n} = 1$.
9. $\lim_{n \rightarrow +\infty} \left[\lim_{p \rightarrow +\infty} \left(\frac{p}{p+1} \right)^n \right] = \lim_{p \rightarrow +\infty} \left[\lim_{n \rightarrow +\infty} \left(\frac{p}{p+1} \right)^n \right]$
10. Si f est croissante sur \mathbb{R} , alors la suite définie par $u_{n+1} = f(u_n)$ et $u_0 = a$ est croissante.
11. Si f est décroissante sur \mathbb{R} , alors la suite définie par $u_{n+1} = f(u_n)$ et $u_0 = a$ est décroissante.

— Entraînez-vous

Test 3.2 (Limite de somme - ♥)

On considère la suite $(s_n)_{n \in \mathbb{N}}$ de terme général :

$$s_n = \sum_{k=1}^n \frac{n}{n^2 + k}$$

En déterminant le plus petit et le plus grand terme de la somme s_n , donnez un encadrement de s_n et déduisez-en la limite de $(s_n)_{n \in \mathbb{N}}$.

Test 3.3 (Comportement asymptotique d'une suite croissante - ♥)

Soit $\lambda > 0$ et $(u_n)_{n \in \mathbb{N}}$ une suite telle que $u_0 > 0$, $u_1 > 0$ et :

$$\forall n \in \mathbb{N}, u_{n+2} = u_{n+1} + \lambda u_n$$

Montrez que la suite diverge vers $+\infty$. Vous commencerez par étudier le sens de variation de la suite, puis vous raisonnerez par l'absurde.

Test 3.4 (Gendarme caché - ♥)

Soit $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$ deux suites telles que $\lim_{n \rightarrow +\infty} (u_n^2 + u_n v_n + v_n^2) = 0$. Montrez que :

$$\lim_{n \rightarrow +\infty} u_n = \lim_{n \rightarrow +\infty} v_n = 0$$

Test 3.5 (Suite arithmético-géométrique - ♥)

$$\text{Étudiez } \begin{cases} u_0 = 6 \\ u_{n+1} = \frac{2u_n + 3}{5} \end{cases} \text{ et calculez } S_n = \sum_{k=0}^n u_k.$$

Test 3.6 (Sommes télescopiques - ♥)

Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie par
$$\begin{cases} u_0 = 0 & u_1 = 1 \\ u_{n+2} = \frac{1}{2}(u_{n+1} + u_n) \end{cases}$$

Pour tout $n \in \mathbb{N}$, posons $v_n = u_{n+1} - u_n$.

1. Montrez que la suite $(u_n)_{n \in \mathbb{N}}$ est géométrique et exprimez u_n en fonction de n .
2. Calculez, en fonction de n , la somme $S_n = \sum_{k=0}^{n-1} v_k$ et déduisez-en u_n en fonction de n .

Test 3.7 (Une incursion dans le programme de Math Spé... - ♥)

1. Montrez que 3,222222222... est un nombre rationnel que vous déterminerez.
2. Même question avec 0,3232323232323...
3. Même question avec $1 - \frac{1}{3} + \frac{1}{9} - \frac{1}{27} + \frac{1}{81} + \dots$

Test 3.8 (Suite homographique - ♥)

On définit la suite $(u_n)_{n \in \mathbb{N}}$ par :

$$u_0 = 2 \text{ et pour tout } n \in \mathbb{N} \quad u_{n+1} = \frac{2u_n - 1}{u_n + 4}$$

1. Étudiez la fonction $f : x \mapsto \frac{2x-1}{x+4}$ et déduisez-en que la suite (u_n) est bien définie pour tout $n \in \mathbb{N}$.
2. Montrez que l'équation $x = \frac{2x-1}{x+4}$ admet une seule solution, que l'on notera α .
3. Montrez que $u_n \neq \alpha$ pour tout $n \in \mathbb{N}$.
4. Pour $n \in \mathbb{N}$, on pose :

$$v_n = \frac{1}{u_n - \alpha}$$

Montrez que la suite (v_n) est arithmétique.

5. Exprimez u_n en fonction de n pour tout n .
6. Étudier la convergence de la suite (u_n) .

Test 3.9 (Paradoxe de Zénon - ♥♥♥♥)

Le paradoxe suivant a été imaginé par Zénon d'Élée (490-430 av. J.-C.). Achille fait une course avec une tortue. Il part 100 mètres derrière la tortue, mais il va dix fois plus vite qu'elle. Quand Achille arrive au point de départ de la tortue, la tortue a parcouru 10 mètres. Pendant qu'Achille parcourt ces 10 mètres, la tortue a avancé d'un mètre. Pendant qu'Achille parcourt ce mètre, la tortue a avancé de 10 cm... Puisqu'on peut réitérer ce raisonnement à l'infini, Zénon conclut qu'Achille ne peut pas dépasser la tortue.

Avec des outils plus modernes que ceux dont disposait Zénon, calculez le temps nécessaire à Achille pour dépasser la tortue en notant v sa vitesse supposée constante.

Test 3.10 (Suite récurrente - ♥♥♥♥)

Étudiez la suite (u_n) définie par $u_0 = -4$ et $u_{n+1} = \sqrt{3 + \frac{u_n}{2}}$.

Test 3.11 (Une suite convergente de nombres rationnels a-t-elle toujours une limite rationnelle ? - ♥♥♥♥)

Soit (u_n) la suite définie sur \mathbb{N} par
$$\begin{cases} u_0 = 2 \\ u_{n+1} = \frac{1}{2} \left(u_n + \frac{2}{u_n} \right) \end{cases} \text{ pour tout } n \in \mathbb{N}.$$

1. Montrez que, pour tout $n \in \mathbb{N}$, $u_n > 0$.
2. Montrez que, pour tout $n \in \mathbb{N}$:

$$u_{n+1} - \sqrt{2} = \frac{(u_n - \sqrt{2})^2}{2u_n}$$

3. En déduire que (u_n) est minorée par $\sqrt{2}$.
4. Déduisez-en que, pour tout $n \in \mathbb{N}$, $\frac{2}{u_n^2} \leq 1$.
5. Étudiez alors les variations de (u_n) et déduisez-en que cette suite est convergente.
6. Déterminez la limite ℓ de (u_n) .
7. Montrez que, pour tout $n \in \mathbb{N}$, $u_n \in \mathbb{Q}$. Est-ce que $\ell \in \mathbb{Q}$ pour autant ? Incroyable, non ?
8. Une suite convergente de nombres rationnels a-t-elle toujours une limite rationnelle ?
9. Soit (y_n) la suite définie sur \mathbb{N} par : pour tout $n \in \mathbb{N}$, $y_n = \frac{2}{u_n}$.

- a) Montrez que (y_n) est majorée par $\sqrt{2}$ puis étudiez ses variations.
- b) Prouvez que, pour tout $n \in \mathbb{N}$:

$$\sqrt{2} + 1 \leq u_n + y_n \leq \sqrt{2} + 2$$

- c) Établissez que, pour tout $n \in \mathbb{N}$:

$$0 \leq u_{n+1} - y_{n+1} = \frac{(u_n - y_n)^2}{2(u_n + y_n)} \leq \frac{(u_n - y_n)^2}{4}$$

- d) Déduisez-en que, pour tout $n \in \mathbb{N}$:

$$0 \leq u_n - y_n \leq (1/4)^{2^n - 1}$$

- e) Que peut-on dire des suites (u_n) et (y_n) ?

10. Déduisez-en une valeur approchée rationnelle de $\sqrt{2}$ à 10^{-9} près.
11. Comment qualifieriez-vous la convergence de (u_n) vers $\sqrt{2}$?

Test 3.12 (Une bille qui rebondit - ♥♥♥♥)

Vous aurez besoin d'utiliser quelques lois physiques du programme de Terminale :

- En chute libre verticale, l'altitude z suit la loi :

$$z(t) = -\frac{1}{2}gt^2 + v_0t + z_0$$

- La vitesse suit la loi $v(t) = -gt + v_0$ en coordonnées algébriques.
- Théorème de l'énergie cinétique :

$$E_{C(B)} - E_{C(A)} = \frac{1}{2}mv_B^2 - \frac{1}{2}mv_A^2 = -mg(z_B - z_A)$$

1. Une balle part d'une certaine hauteur h_0 au-dessus du sol (sans vitesse initiale et en négligeant les frottements). On modélise le rebond de la façon suivante : lorsque la balle rebondit, elle perd une certaine proportion p de son énergie cinétique (par exemple $p = 10\%$) et elle met le même temps pour monter et descendre entre deux rebonds. Étant partie de la hauteur h_0 , à quelle hauteur h_1 va-t-elle remonter ? Quelle est la durée t_0 entre les deux premiers rebonds ?
2. Combien de fois la balle rebondit-elle ? Pendant combien de temps rebondit-elle ? (Pour l'application numérique, on prendra $h_0 = 1$ m.)

Test 3.13 (Analyse numérique : résolution d'une équation par la méthode de Newton - ♥♥♥♥)

La méthode de Newton (ou méthode des tangentes) est une manière d'obtenir des approximations numériques d'un « zéro » d'une fonction (c'est-à-dire d'un nombre où la fonction s'annule). Dans cet exercice, on décrit cette méthode, puis on l'applique au calcul des premières décimales de $\sqrt{2}$. La deuxième partie de l'exercice propose de comparer cette méthode avec la méthode par dichotomie.

La méthode de Newton

1. Toutes les méthodes de recherche numérique de zéro de fonctions suivent le même principe général : à partir d'une « première bonne approximation » du zéro recherché, trouvez une approximation qui soit encore meilleure, puis recommencer...

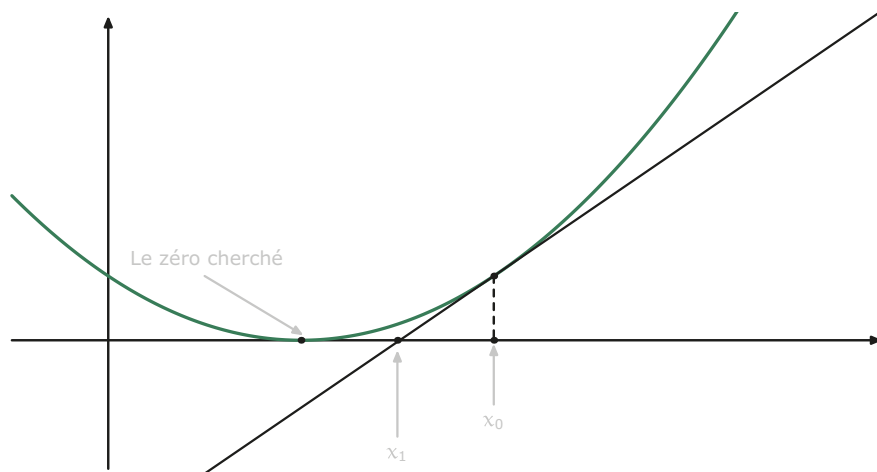
Voici comment procède la méthode de Newton. Soit f une fonction de \mathbb{R} dans \mathbb{R} .

- On part d'un nombre quelconque x_0 ;
- à partir de x_0 , on calcule un nouveau nombre x_1 de la manière suivante (voir figure suivante) : on trace la tangente au graphe de f au point d'abscisse x_0 , et on détermine le point d'intersection de cette tangente avec l'axe des abscisses. On appelle x_1 l'abscisse de ce point d'intersection ;
- et on recommence : on calcule un nouveau nombre x_2 en appliquant le procédé décrit au point 2 où l'on remplace x_0 par x_1 ;
- etc.

À partir de cette description graphique de la méthode de Newton, trouvez la formule donnant x_1 en fonction de x_0 .

Quelles hypothèses doit-on faire sur f et x_0 pour que la formule ait un sens ?

2. On veut utiliser la méthode de Newton pour calculer, à la main, une approximation décimale du nombre $\sqrt{2}$. Pour ça, on prend la fonction $f(x) = x^2 - 2$ et on part de $x_0 = \frac{3}{2}$.



- a) Notons g la fonction vérifiant $x_{n+1} = g(x_n)$. Tracez le graphe de la fonction g , représentez graphiquement les nombres x_0, x_1, x_2 et x_3 . Calculez les nombres x_1, x_2, x_3 et x_4 sous forme de fraction.
- b) Sachant que le début du développement décimal de $\sqrt{2}$ est 1,41421356237309504880168872421 dites, pour x_4 , combien on a trouvé de bonnes décimales.
- c) Il s'agit maintenant de prouver ces résultats en étudiant la suite (x_n) :
- Étudiez la fonction g sur $]0, +\infty[$ et montrez que $[\sqrt{2}, x_0]$ est stable par g .
 - Étudiez le sens de variation de (x_n) .
 - Montrez que (x_n) converge bien vers $\sqrt{2}$.

Comparaison avec la dichotomie

Voici une description de la méthode par dichotomie.

Soit f une fonction de \mathbb{R} dans \mathbb{R} , dont on connaît le signe autour de α : par exemple, f est négative avant α et positive après.

- On commence par choisir un nombre x_0 que l'on sait être plus petit que le zéro α recherché ;
- on choisit aussi un nombre x_1 plus grand que α ;
- on prend pour x_2 le milieu de l'intervalle $[x_0, x_1]$;
- le nombre x_3 est alors déterminé de la manière suivante : si α est dans l'intervalle de gauche (autrement dit si $f(x_2) < 0$), alors on choisit pour x_3 le milieu de cet intervalle ; sinon, on choisit le milieu de l'intervalle de droite ;
- et on recommence : le nombre x_3 découpe le nouvel intervalle en deux intervalles deux fois plus petits, et on choisit pour x_4 le milieu d'un de ces deux intervalles, selon la position de α ;
- etc.

On cherche à nouveau une approximation de $\sqrt{2}$. Pour ça, on va appliquer la méthode par dichotomie avec $x_0 = 1$ et $x_1 = 2$.

- Montrez rapidement (mais rigoureusement) que $\sqrt{2} \in [x_0, x_1]$.
- Calculez les nombres x_2, x_3, \dots, x_7 sous forme de fractions.
- Dites, pour x_7 , combien on a trouvé de « bonnes » décimales.

Préparez-vous aux colles

Test 3.14 (♥)

Cours

Énoncez et démontrez le théorème des suites adjacentes.

Premier exercice

Soit $(u_n)_{n \in \llbracket 1; +\infty \rrbracket}$ une suite convergeant vers ℓ et M_n la moyenne arithmétique des n premiers termes de la suite n . Montrez que (M_n) converge vers ℓ . (Il s'agit en fait du théorème de Cesàro que vous étudierez en cours.)

Deuxième exercice

Soit $\ell \in \mathbb{R}^*$ et $(a_n)_{n \in \llbracket 1; +\infty \rrbracket}$ une suite réelle telle que $(a_{n+1} - a_n)_{n \in \llbracket 1; +\infty \rrbracket}$ converge vers ℓ .

Montrez que $\lim_{n \rightarrow +\infty} \frac{a_n}{n} = \ell$. (Ce résultat classique est appelé « lemme de l'escalier ».)

Test 3.15 (♥♥)

Cours

Démontrez les critères de convergence d'une suite géométrique.

Premier exercice

Soit $a \in \mathbb{R}$. Étudiez la suite de premier terme $u_0 = a$ et telle que $u_{n+1} = \frac{1+u_n^2}{2}$ pour tout entier naturel n .

Deuxième exercice

Soit (u_n) la suite de terme général :

$$u_n = \sqrt{a_1 + \sqrt{a_2 + \cdots + \sqrt{a_n}}}$$

avec a_n la n -ième décimale de π .

Étudiez la convergence de cette suite.

Coups de pouce

Coup de pouce pour le test 3.4

Vous pourrez commencer par mettre $X^2 + Xv_n + v_n^2$ sous forme canonique.

Coup de pouce pour le test 3.7

1. Introduisez la somme des termes d'une suite géométrique.

Coup de pouce pour le test 3.15

On pourra introduire la suite (v_n) de terme général $\sqrt{9 + \sqrt{9 + \dots + \sqrt{9}}}$.

Solutions et commentaires

Solution du test 3.1

1. Vrai. C'est un résultat fondamental et la preuve est un modèle pour toute l'analyse.

L'idée est simple : soit ℓ la limite de $(u_n)_{n \in \mathbb{N}}$ et soit ε un réel strictement positif. Puisque la suite est convergente, il existe un rang N tel que, pour tout $n \geq N$, on a $u_n \in]\ell - \varepsilon; \ell + \varepsilon[$, ce qui exprime que $(u_n)_{n \in \mathbb{N}}$ est bornée à partir du rang N .

D'autre part, l'ensemble $\{u_n, 0 \leq n < N\}$ est fini, donc il admet un plus grand et un plus petit élément, ce qui exprime que la suite est bornée jusqu'au rang N . Finalement, la suite est bornée tout court.

2. Vrai. Un peu de logique : il est équivalent de prouver $A \Rightarrow B$ et contraire de $B \Rightarrow$ contraire de A . On dit que la deuxième assertion est la **contraposée** de la première.

Voir la section sur la logique du chapitre 1.

Ici, la contraposée de l'assertion à prouver est :

« Si une suite est convergente, alors elle est bornée. »

3. Faux. Il suffit de considérer la suite de terme général $(-1)^n$.

4. Faux. Considérez par exemple la suite de terme général $u_n = \ln n$.

5. Vrai. Raisonnons par l'absurde. Supposons que $(u_n)_{n \in \mathbb{N}}$ converge vers ℓ , que $(v_n)_{n \in \mathbb{N}}$ diverge et que $(u_n + v_n)_{n \in \mathbb{N}}$ converge vers L . Alors nous aboutissons à la contradiction suivante :

$$v_n = (u_n + v_n) - u_n \xrightarrow{n \rightarrow +\infty} L - \ell$$

6. Faux. Considérez les suites $u_n = n$ et $v_n = -n$.

7. Vrai. C'est assez évident intuitivement. En effet, tous les termes de la suite doivent être aussi proches que l'on veut de la limite à partir d'un certain rang, disons à moins de $1/4$. Or deux entiers situés à une distance inférieure à $1/4$ sont égaux et donc la suite sera stationnaire. On peut exprimer cette idée de manière plus formelle.

Comme la suite converge, il existe un entier N tel que, pour tout $n \in \mathbb{N}$ on a $n \geq N \Rightarrow |u_n - \ell| \leq 1/4$. Ensuite on utilise l'inégalité triangulaire :

$$\begin{aligned} |u_n - u_N| &= |u_n - \ell + \ell - u_N| \\ &\leq |u_n - \ell| + |\ell - u_N| = |u_n - \ell| + |u_N - \ell| \\ &\leq 1/4 + 1/4 = 1/2 \end{aligned}$$

8. a) Faux. Prenez $u_n = n + 1$ et $v_n = n$, alors :

$$\frac{e^{u_n}}{e^{v_n}} = e^{u_n - v_n} = e$$

b) Faux. Il y a déjà danger si v_n vaut 1. Même en admettant le contraire, il y a encore danger quand (v_n) converge vers 1. Considérez en effet les suites de terme général $u_n = 1 + 1/n$ et $v_n = 1 + 1/n^2$, alors :

$$\frac{\ln u_n}{\ln v_n} = \frac{\ln(1 + 1/n)}{1/n} \times (1/n) \times \frac{1/n^2}{\ln(1 + 1/n^2)} \times n^2 = n \times \frac{\ln(1 + 1/n)}{1/n} \times \frac{1/n^2}{\ln(1 + 1/n^2)}$$

Or $\lim_{x \rightarrow 0} \frac{\ln(1+x)}{x} = 2$ donc :

$$\lim_{n \rightarrow +\infty} \frac{\ln u_n}{\ln v_n} = \lim_{n \rightarrow +\infty} n = +\infty$$

c) Vrai. Comme $u_n / v_n \xrightarrow[n \rightarrow +\infty]{} 1$, alors $(u_n / v_n)^x \xrightarrow[n \rightarrow +\infty]{} 1$.

d) Ça n'est plus pareil : l'exposant n'est plus fixe mais dépend maintenant de n . Or attention, 1^∞ est une forme indéterminée. Par exemple ici, prenez $u_n = (1 + 1/n)$ et $v_n = (1 + 2/n)$. Un résultat classique donne $\lim_{n \rightarrow +\infty} u_n = e$ et $\lim_{n \rightarrow +\infty} v_n = e^2$.

9. Faux.

$$\lim_{n \rightarrow +\infty} \left[\lim_{p \rightarrow +\infty} \left(\frac{p}{p+1} \right)^n \right] = \lim_{n \rightarrow +\infty} 1^n = \lim_{n \rightarrow +\infty} 1 = 1$$

$$\lim_{p \rightarrow +\infty} \left(\frac{p}{p+1} \right)^n = \lim_{p \rightarrow +\infty} 0 = 0$$

Moralité : attention aux interversions de limites ! C'est loin d'être un problème anodin. Il faudra attendre d'étudier la convergence uniforme en Maths Spé...

10. Faux. Prenez par exemple $u_{n+1} = \sqrt{u_n}$ et $u_0 = 2$. Alors $u_1 = \sqrt{2} < u_0$. Maintenant, souvenez-vous du cours : f étant croissante, elle conserve l'ordre, donc une récurrence immédiate montre que $u_{n+1} < u_n$ pour tout naturel n et donc (u_n) est décroissante. D'ailleurs, si nous avions choisi $u_0 < 1$, nous aurions obtenu $u_0 < u_1$ et la suite aurait cette fois-ci été croissante.

11. Faux. Autre piège habituel. C'est le dessin en coquille d'escargot de la figure page 82.

Solution du test 3.2

Pour ce genre de suites, une première méthode (qui servira également pour les intégrales qui sont aussi des « sommes ») consiste à minorer et majorer la suite en utilisant le plus petit et le plus grand terme de la somme.

Ici, on obtient successivement :

$$\frac{n}{n^2 + n} \ll \frac{n}{n^2 + k} \ll \frac{n}{n^2 + 1}$$

$$n \times \frac{n}{n^2 + n} \leq s_n \leq n \times \frac{n}{n^2 + 1}$$

Or, chaque gendarme tendant vers 1, on obtient $\lim_{n \rightarrow +\infty} s_n = 1$.

Solution du test 3.3

Une rapide récurrence montre que la suite est à termes strictement positifs et donc $u_{n+2} - u_{n+1} = \lambda u_n > 0$. Nous en déduisons que la suite $(u_n)_{n \in \llbracket 1; +\infty \rrbracket}$ est strictement croissante. Deux choix s'offrent alors à nous : soit (u_n) converge vers un réel ℓ , soit elle diverge vers $+\infty$.

Supposons que la première affirmation soit vraie, alors on aurait nécessairement $\ell = \ell + \lambda \ell$ et donc ℓ serait nulle car $\lambda > 0$. Or (u_n) ne peut converger vers 0, car, pour tout $n > 1$, on a $u_n > u_1 > 0$. Donc c'est l'autre possibilité qu'il faut retenir : $\lim_{n \rightarrow +\infty} u_n = +\infty$.

Attention ! $u_n > 0$ ne suffit pas ! En effet, la suite de terme général $1/n$ converge vers 0. Retenez que, en passant à la limite, les inégalités strictes deviennent larges (vous le prouverez l'an prochain). Il faut donc un « tampon » de sécurité, ici u_0 .

Solution du test 3.4

Puisque $X^2 + Xv_n + v_n^2 = (X + v_n / 2)^2 + 3v_n^2 / 4$, alors :

$$0 \leq 3v_n^2 / 4 \leq (u_n^2 + u_n v_n + v_n^2)$$

Un appel aux forces de l'ordre nous permet d'établir que la suite de terme général v_n^2 tend vers 0 et, par suite, (v_n) aussi. Les rôles de (u_n) et (v_n) étant symétriques, nous en déduisons de même que $\lim_{n \rightarrow +\infty} u_n = 0$.

Solution du test 3.5

L'équation $x = \frac{2x+3}{5} = 0$ admet pour unique solution 1. Étudions la suite de terme général $v_n = u_n - 1$. Alors :

$$v_{n+1} = u_{n+1} - 1 = \frac{2u_n - 2}{5} = \frac{2}{5}(u_n - 1) = \frac{2}{5}v_n$$

La suite v_n est donc géométrique de raison $2/5$ et de premier terme $v_0 = 5$. On obtient alors $v_n = 5 \times (2/5)^n$, puis $u_n = 5 \times (2/5)^n + 1$.

On en déduit que :

$$\begin{aligned} S_n &= 5 \sum_{k=0}^n \left(\frac{2}{5}\right)^k + \sum_{k=0}^n 1 \\ &= 5 \frac{1 - (2/5)^{n+1}}{1 - 2/5} + n + 1 = \frac{25}{3} \left(1 - \left(\frac{2}{5}\right)^{n+1}\right) + n + 1 \end{aligned}$$

Solution du test 3.6

1. $v_{n+1} = u_{n+2} - u_{n+1} = \frac{1}{2}(-u_{n+1} + u_n) = -\frac{1}{2}v_n$, donc $v_n = \left(\frac{-1}{2}\right)^n$.

2. $S_n = \frac{2}{3} \left(1 - \left(-\frac{1}{2}\right)^{n+1}\right) = (u_n - u_{n-1}) + (u_{n-1} - u_{n-2}) + \dots + (u_2 - u_1) + (u_1 - u_0) = u_n - u_0 = u_n$

Solution du test 3.7

1. $\underbrace{3,222\dots 2}_{2 \text{ écrit } n \text{ fois}} = 3 + 2 \times \left(\frac{1}{10} + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^3 + \dots + \left(\frac{1}{10}\right)^n\right) = 3 + 2 \frac{1}{10} \frac{1 - (1/10)^{n+1}}{1 - 1/10} = 3 + \frac{2}{9} (1 - (1/10)^{n+1})$

Le nombre demandé s'obtient en faisant tendre n vers l'infini, et donc :

$$3,222222222222\dots = \frac{29}{9}$$

2. De même

$$0,3232323232\dots = \lim_{n \rightarrow +\infty} 32 \sum_{k=1}^n \left(\frac{1}{100}\right)^k = \frac{32}{99}$$

3. $1 - \frac{1}{3} + \frac{1}{9} - \frac{1}{27} + \frac{1}{81} + \dots = \lim_{n \rightarrow +\infty} \sum_{k=0}^n \left(-\frac{1}{3}\right)^k = \frac{3}{4}$

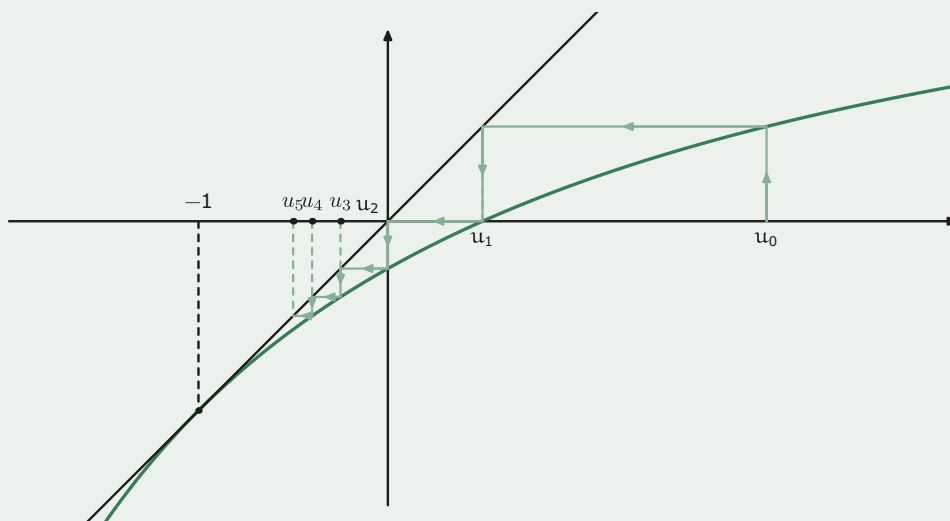
Une suite de terme général $\sum_{k=0}^n u_k$ est appelée une série numérique. Vous les étudierez en 2^e année. Lorsque (u_n) est une suite géométrique, la série associée est une série... géométrique. Remarquez qu'une somme infinie de termes strictement positifs peut avoir une limite finie. Nous en reparlerons abondamment dans les exercices 3.9 et 3.12.

Solution du test 3.8

1. Le danger vient de -4 . « Observons » la suite pour nous donner des idées. Il semble que u_n reste toujours supérieur à -1 compte tenu du u_0 choisi. Prouvons-le en étudiant la fonction f sur $]-4; +\infty[$. On obtient que :

$$f([-1; +\infty]) = [-1; 2] \subset [-1; +\infty[$$

Puisque l'intervalle $I = [-1; +\infty[$ est stable par f et que $u_0 \in I$, alors $u_n \in I$ quel que soit $n \in \mathbb{N}$. La suite (u_n) est donc bien définie pour tout $n \in \mathbb{N}$.



Remarque : On aurait pu montrer par récurrence que $u_n > -1$ pour tout $n \in \mathbb{N}$ sans parler de f , mais en travaillant uniquement sur des inégalités. Le problème est que dans ce cas on travaille sur la suite (u_n) avant de savoir si elle est ou non définie, ce qui est gênant.

2. $x = \frac{2x-1}{x+4} \Leftrightarrow x^2 + 2x + 1 = 0 \Leftrightarrow (x+1)^2 = 0$ qui admet pour unique solution -1 .

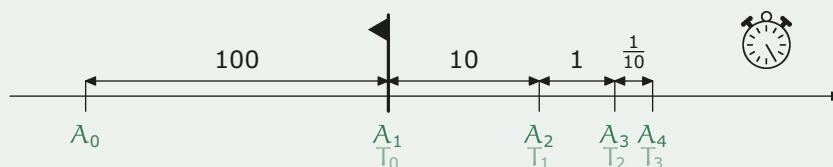
3. Voir question 2.

4. $v_{n+1} = 1/3 + v_n$ après calculs.

5. Alors $v_n = \frac{1}{3}(n+1)$ et $u_n = \frac{2-n}{1+n}$.

6. On en déduit que $\lim_{n \rightarrow +\infty} u_n = 1$.

Solution du test 3.9



Intéressons-nous d'abord à la distance Achille-tortue.

Notons d_n la distance :

$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

La suite (d_n) est donc géométrique de raison $1/10$ et de premier terme 100 , donc $d_n = 100 \times \left(\frac{1}{10}\right)^{n-1}$.

Or $|1/10| < 1$, donc $\lim_{n \rightarrow +\infty} (1/10)^{n-1} = 0 = \lim_{n \rightarrow +\infty} d_n$

Ainsi Achille va rattraper la tortue, mais au bout d'une infinité de trajets.

Intéressons-nous plutôt à la durée du trajet d'Achille pour atteindre la tortue, en supposant sa vitesse constante et égale à v .

Notons $t_1 = \frac{d_1}{v} = \frac{100}{v}$, $t_2 = \frac{d_2}{v} = \frac{1}{10} \frac{d_1}{v} = \frac{1}{10} t_1$, etc.

Donc (t_n) est une suite géométrique de raison $\frac{1}{10}$, de premier terme $t_1 = 100/v$, d'où $t_n = \frac{100}{v} \left(\frac{1}{10}\right)^{n-1}$.

La durée du trajet jusqu'à A_n est alors :

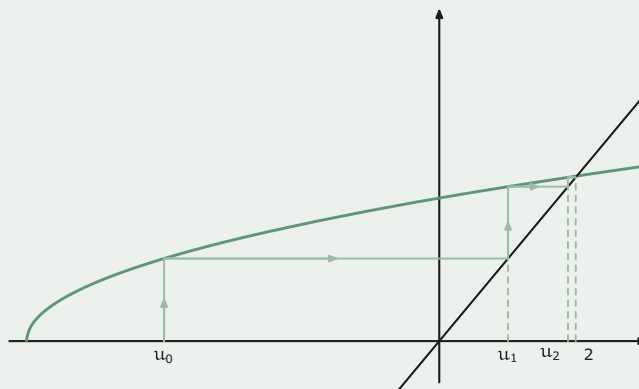
$$\begin{aligned}\tau_n &= t_1 + t_2 + \dots + t_n \\ &= \frac{100}{v} \left(1 + \frac{1}{10} + \left(\frac{1}{10} \right)^2 + \dots + \left(\frac{1}{10} \right)^{n-1} \right) \\ &= \frac{1000}{9v} \left(1 - \left(\frac{1}{10} \right)^n \right)\end{aligned}$$

Nous venons de voir qu'Achille atteindra la tortue quand n tend vers $+\infty$. Or $\lim_{n \rightarrow +\infty} \tau_n = \frac{1000}{9v}$ est un nombre fini.

Achille va donc effectuer cette infinité de trajets en un temps fini égal à $1000/9v$.

Solution du test 3.10

Traçons la représentation de la fonction associée ainsi que de la première bissectrice et on obtient les trois premiers termes de la suite.



Il semble qu'à partir du rang 1, tous les termes de la suite soient bornés par 0 et 2. Prouvons-le, ce qui assurera la définition de la suite. On étudie alors la fonction $x \mapsto \sqrt{3+x/2}$ sur $[0;2]$ et on montre que l'intervalle $[0; 2]$ est stable par f , ce qui est assez immédiat car f est croissante sur $[0; 2]$ et :

$$0 < f(0) = \sqrt{3} < f(2) = 2$$

Comme nous l'avons vu dans le cours, puisque f est croissante, la suite sera monotone, son sens de variation dépendant du signe de $u_1 - u_0$. Or $u_1 = 1 > u_0 = -4$, donc la suite est croissante.

Or elle est majorée par 2, donc elle converge. Puisque f est continue sur le fermé $[0; 2]$, la suite admettra pour limite un point fixe de f sur cet intervalle. Or l'équation $f(x) = x$ n'admet que 2 comme solution sur cet intervalle.

Finalement $\lim_{n \rightarrow +\infty} u_n = 2$.

Solution du test 3.11

1. Une rapide récurrence permet de conclure.

$$\begin{aligned}2. u_{n+1} - \sqrt{2} &= \frac{1}{2} \left(u_n + \frac{2}{u_n} \right) - \sqrt{2} = \frac{u_n}{2} - \sqrt{2} + \frac{1}{u_n} \\ &= \frac{u_n^2 - 2\sqrt{2}u_n + 2}{2u_n} = \frac{(u_n - \sqrt{2})^2}{2u_n}\end{aligned}$$

3. Ainsi $u_n - \sqrt{2} \geq 0$ pour tout $n > 0$ et $u_0 = 2 > \sqrt{2}$, donc (u_n) est minorée par $\sqrt{2}$.

4. Pour tout $n \in \mathbb{N}$, $u_n \geq 2$, donc $u_n^2 \geq 2$, i.e. $\frac{2}{u_n^2} \leq 1$ car $u_n^2 > 0$.

5. Comme $u_n > 0$, on peut étudier :

$$\frac{u_{n+1}}{u_n} = \frac{1}{2} \left(1 + \frac{2}{u_n^2} \right) \leq \frac{1}{2} (1+1) = 1$$

donc la suite (u_n) est décroissante, or elle est minorée par $\sqrt{2}$, donc elle est convergente.

6. Comme $u_{n+1} = f(u_n)$ avec $f : x \mapsto \frac{1}{2} \left(x + \frac{2}{x} \right)$ continue sur le fermé $[\sqrt{2}; +\infty[$, et que (u_n) est convergente, sa limite est nécessairement un point fixe de f .

Or $f(x) = x \Leftrightarrow x^2 = 2$ qui n'admet que $\sqrt{2}$ comme solution sur $[\sqrt{2}; +\infty[$, donc $\lim_{n \rightarrow +\infty} u_n = \sqrt{2}$.

7. Une rapide récurrence montre que $u_n \in \mathbb{Q}$ pour tout $n \in \mathbb{N}$, et pourtant la limite de u_n est irrationnelle.

8. Vous devez donc retenir qu'il faut être très prudent avant de transposer une propriété vraie pour les termes de la suite à sa limite.

9. a) Puisque $u_n \geq \sqrt{2}$, on obtient $\frac{1}{\sqrt{2}} \geq \frac{1}{u_n}$ et donc $\sqrt{2} \geq \frac{2}{u_n} = y_n$. De plus, $\frac{y_{n+1}}{y_n} = \frac{u_n}{u_{n+1}} \geq 1$, donc (y_n) est croissante.

b) $\sqrt{2} + 1 = \sqrt{2} + y_0 \leq \sqrt{2} + y_n \leq u_n + y_n \leq u_n + \sqrt{2} + u_0 + \sqrt{2} = 2 + \sqrt{2}$

c) Comme $u_{n+1} = \frac{1}{2}(u_n + y_n)$, $y_{n+1} = \frac{2}{u_{n+1}} = \frac{4}{u_n + y_n}$ et $u_n y_n = 2$, on obtient :

$$\begin{aligned} u_{n+1} - y_{n+1} &= \frac{(u_n + y_n)^2 - 8}{2(u_n + y_n)} \\ &= \frac{u_n^2 + y_n^2 + 2u_n y_n - 8}{2(u_n + y_n)} \\ &= \frac{u_n^2 + y_n^2 - 2u_n y_n}{2(u_n + y_n)} \\ &= \frac{(u_n - y_n)^2}{2(u_n + y_n)} \end{aligned}$$

Puis :

$$0 \leq \frac{(u_n - y_n)^2}{2(u_n + y_n)} \leq \frac{(u_n - y_n)^2}{2(\sqrt{2} + 1)} \leq \frac{(u_n - y_n)^2}{2 \times 2} = \frac{(u_n - y_n)^2}{4}$$

d) Une récurrence règle le problème. Occupons-nous juste de l'hérédité :

$$\begin{aligned} u_{k+1} - y_{k+1} &\leq \frac{(u_k - y_k)^2}{4} \leq \left(\frac{1}{4^{2^k - 1}} \right) \times \frac{1}{4} \\ \text{et } \left(\frac{1}{4^{2^k - 1}} \right)^2 \times \frac{1}{4} &= \frac{1}{4^{2^k \times 2 - 2}} \times \frac{1}{4} = \frac{1}{4^{2^{k+1} - 1}} \end{aligned}$$

e) Puisque (u_n) est décroissante, (y_n) croissante et $\lim_{n \rightarrow +\infty} (u_n - y_n) = 0$ (gendarmes), les deux suites sont alors adjacentes, donc elles convergent vers la même limite, $\sqrt{2}$.

10. Puisque $y_n \leq \sqrt{2} \leq u_n$ pour tout $n \in \mathbb{N}$, on a en particulier $0 \leq u_n - \sqrt{2} \leq (1/4)^{2^n-1}$, donc u_n donnera une valeur approchée de $\sqrt{2}$ à 10^{-9} près dès que $(1/4)^{2^n-1} \leq 10^{-9}$, c'est-à-dire, après calculs, dès que :

$$n \geq \frac{\ln\left(\frac{9 \ln 10}{\ln 4} + 1\right)}{\ln 2} \approx 3,99$$

Donc $u_4 = \frac{665\,857}{470\,832}$ est une approximation de $\sqrt{2}$ à 10^{-9} près (en fait 10^{-11}).

11. La convergence de (u_n) est donc extrêmement rapide puisque, au bout de 4 itérations, on obtient 11 bonnes décimales. Vous pourrez comparer avec la méthode de Newton de l'exercice **3.13**.

Solution du test 3.12

1. La balle est lâchée avec une vitesse initiale nulle. Elle touche le sol avec une vitesse V_{S_0} . Elle remonte jusqu'à la hauteur h_1 où sa vitesse v_1 est nulle.

Appliquons le théorème de l'énergie cinétique entre le lâcher et le premier contact avec le sol :

$$\frac{1}{2} m V_{S_0}^2 - 0 = -mg(0 - h_0)$$

$$\text{puis } 0 - \frac{9}{10} \left(\frac{1}{2} m V_{S_0}^2 \right) = -mg(h_1 - 0).$$

On obtient donc $\frac{9}{10}(mgh_0) = mgh_1$, d'où :

$$h_1 = \frac{9}{10} h_0$$

En chute libre et en considérant des vitesses positives et que la balle met le même temps pour monter et descendre entre deux rebonds :

$$V_{S_1} = g \frac{t_0}{2}$$

Or $\frac{1}{2} m V_{S_1}^2 = mgh_1$, donc $V_{S_1} = \sqrt{2gh_1}$ et alors :

$$t_0 = 2\sqrt{\frac{2h}{g}}$$

2. On montre facilement que :

$$t_1 = 2\sqrt{\frac{2h}{g}} \sqrt{h_2} = 2\sqrt{\frac{2h}{g}} \sqrt{\frac{9}{10} h_1} = \frac{3}{\sqrt{10}} t_0$$

Par suite :

- (h_n) est une suite géométrique de raison $9/10$ et de premier terme h_0 ;
- (t_n) est une suite géométrique de raison $3/\sqrt{10}$ et de premier terme t_0 avec t_n le temps écoulé entre le $(n+1)$ -ième et le $(n+2)$ -ième rebond.

On a $h_n > 0$ pour tout entier naturel n , mais $\lim_{n \rightarrow +\infty} h_n = 0$, donc théoriquement la balle rebondira une infinité de fois avant de s'arrêter !

Notons T_n le temps écoulé après $n+2$ rebonds :

$$\begin{aligned} T_n &= t_0 + t_1 + \dots + t_n \\ &= t_0 \left(1 + \frac{3}{\sqrt{10}} + \dots + \left(\frac{3}{\sqrt{10}} \right)^n \right) \\ &= t_0 \frac{1 - \left(3/\sqrt{10} \right)^{n+1}}{1 - 3/\sqrt{10}} \end{aligned}$$

On obtient donc $\lim_{n \rightarrow +\infty} T_n \frac{t_0 \sqrt{10}}{\sqrt{10} - 3} = \frac{6}{(\sqrt{10} - 3)} \sqrt{\frac{2h_0}{g}}$.

Pour $h_0 = 1$ m, la bille rebondit donc pendant environ 16,7 s : une infinité de rebonds en un temps fini, voici encore de quoi donner des cauchemars à Zénon...

Solution du test 3.13

Méthode de Newton

1. L'équation de la tangente vaut $y = f'(x_0) \times (x - x_0) + f(x_0)$. Or x_1 est l'abscisse du point d'intersection de la tangente avec l'axe des abscisses, donc :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

en supposant que la dérivée de f ne s'annule pas sur un voisinage de la solution.

2. a) Dans ce cas particulier, $f'(x) = 2x$ qui ne s'annule pas sur $]0, +\infty[$. On obtient :

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n} = \frac{x_n^2 + 2}{2x_n}$$

Avec par exemple le petit programme suivant :

```
def u(n: int) -> float:
    assert n >= 0, "u définie sur N"
    return 3/2 if n == 0 else u(n-1)/2 + 1/u(n-1)
```

on obtient :

```
for k in range(5):
    print(f"u({k}) = {u(k)}")

u(0) = 1.5
u(1) = 1.4166666666666665
u(2) = 1.4142156862745097
u(3) = 1.4142135623746899
u(4) = 1.414213562373095
```

Pour l'approximation :

```
for k in range(5):
    print(f"Itération {k + 1} : écart = {u(k) - sqrt(2)}")

Itération 1 : écart = 0.08578643762690485
Itération 2 : écart = 0.00245310429357137
Itération 3 : écart = 2.123901414519125e-06
Itération 4 : écart = 1.5947243525715749e-12
Itération 5 : écart = -2.220446049250313e-16
```

i. On a obtenu $g : x \mapsto \frac{x}{2} + \frac{1}{x}$. On en déduit que $g'(x) = \frac{(x - \sqrt{2})(x + \sqrt{2})}{2x^2}$ puis le tableau suivant sur $\left[\sqrt{2}; \frac{3}{2}\right]$:

x	0	1	$+\infty$
Signe de $f'(x)$	0	+	+
Variations de f			

On a bien $f\left(\left[\sqrt{2}; \frac{3}{2}\right]\right) \subset \left[\sqrt{2}; \frac{3}{2}\right]$.

ii. On montre facilement par récurrence d'une part que $x_n \in \left[\sqrt{2}; \frac{3}{2}\right]$ pour tout $n \in \mathbb{N}$ et d'autre part que la suite est décroissante.

iii. La suite (x_n) étant décroissante et minorée par $\sqrt{2}$, elle est convergente.

L'équation $g(x) = x$ admettant l'unique solution $\sqrt{2}$ sur l'intervalle fermé $\left[\sqrt{2}; \frac{3}{2}\right]$ et g étant continue sur cet intervalle, la suite (x_n) converge vers $\sqrt{2}$.

Dichotomie

1. Soit $f : x \mapsto x^2 - 2$: elle est continue et strictement croissante sur $[1, 2]$. Or $f(1) > 0$ et $f(2) < 0$, donc $\sqrt{2} \in [1, 2]$.

2. On obtient :

$$x_2 = \frac{3}{2}, x_3 = \frac{5}{4}, x_4 = \frac{11}{8}, x_5 = \frac{23}{16}, x_6 = \frac{45}{32}, x_7 = \frac{91}{64}$$

3. $x_7 \approx 1,421$: on n'a donc qu'une seule bonne décimale pour x_7 . La méthode par dichotomie semble donc bien peu performante par rapport à la méthode de Newton.

(Pour prouver la convergence de la suite définie par la méthode par dichotomie, il nous manque quelques outils mathématiques que vous étudierez en début d'année de Sup, en particulier le théorème des segments emboîtés.)

Solution du test 3.14

Premier exercice

- Étudions d'abord le cas $\ell = 0$.

Traduisons que la suite (u_n) converge vers 0.

Fixons $\varepsilon > 0$. On voudrait montrer qu'à partir d'un certain rang $|M_n| < \varepsilon$.

Tout d'abord, appliquons l'inégalité triangulaire :

$$|M_n| \leq \frac{1}{n}(|u_1| + \dots + |u_n|)$$

Comme (u_n) tend vers 0, il existe un entier n_0 tel que, pour tout entier $k \geq n_0$, on ait $|u_k| \leq \frac{\varepsilon}{2}$. Alors, on obtient successivement :

$$\begin{aligned}
 |M_n| &\leq \frac{1}{n}(|u_1| + \dots + |u_{n_0}| + |u_{n_0+1}| + \dots + |u_n|) \\
 &\leq \frac{1}{n} \left(|u_1| + \dots + |u_{n_0}| + \underbrace{\frac{\varepsilon}{2} + \dots + \frac{\varepsilon}{2}}_{n-n_0 \text{ termes}} \right) \\
 &\leq \frac{1}{n}(|u_1| + \dots + |u_{n_0}|) + \frac{1}{n}(n-n_0) \frac{\varepsilon}{2} \\
 &\leq \frac{1}{n}(|u_1| + \dots + |u_{n_0}|) + \frac{\varepsilon}{2}
 \end{aligned}$$

Or n_0 est fixé donc la somme $|u_1| + \dots + |u_{n_0}|$ ne dépend pas de n . On en déduit que :

$$\lim_{n \rightarrow +\infty} \frac{1}{n} (|u_1| + \dots + |u_{n_0}|) = 0$$

Il existe donc un entier naturel non nul n_1 tel que, pour tout entier $n \geq n_1$:

$$\frac{1}{n} (|u_1| + \dots + |u_{n_0}|) \leq \frac{\varepsilon}{2}$$

Finalement on obtient, en regroupant les deux inégalités, que :

$$\forall n \geq \max(n_0, n_1) |M_n| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

ce qui prouve que si (u_n) converge vers 0, alors (M_n) aussi.

- Étudions maintenant le cas où ℓ est un réel quelconque.

Il suffit alors, comme d'habitude, de revenir au cas précédent en considérant la suite $(u_n - \ell)$ qui tend vers 0.

Alors d'une part :

$$\lim_{n \rightarrow +\infty} \frac{1}{n} ((u_1 - \ell) + \dots + (u_n - \ell)) = 0$$

et d'autre part :

$$\begin{aligned} & \frac{1}{n} ((u_1 - \ell) + \dots + (u_n - \ell)) \\ &= \frac{1}{n} (u_1 + \dots + u_n - (\ell + \dots + \ell)) \end{aligned}$$

Ainsi $\lim_{n \rightarrow +\infty} (M_n - \ell) = 0$, c'est-à-dire $\lim_{n \rightarrow +\infty} M_n = \ell$.

Deuxième exercice

Posons $u_n = a_{n+1} - a_n$ et (M_n) la moyenne de Cesàro de cette suite :

$$\begin{aligned} M_n &= \frac{1}{n} (a_{n+1} - a_n + a_n - a_{n-1} + \dots + a_3 - a_2 + a_2 - a_1) \\ &= \frac{a_{n+1} - a_1}{n} \\ &= \frac{n+1}{n} \left(\frac{a_{n+1}}{n+1} - \frac{a_1}{n+1} \right) \end{aligned}$$

On en déduit que :

$$\frac{a_{n+1}}{n+1} = \frac{n}{n+1} \cdot M_n + \frac{a_1}{n+1}$$

D'après le théorème de Cesàro, (M_n) converge vers ℓ , $\lim_{n \rightarrow +\infty} \frac{a_1}{n+1} = 0$ et $\lim_{n \rightarrow +\infty} \frac{n}{n+1} = 1$.

On en déduit le résultat.

Solution du test 3.15

Premier exercice

La suite est définie quel que soit a et elle est à termes strictement positifs, sauf éventuellement le premier. Soit $f : x \mapsto (1+x^2)/2$; elle est croissante sur \mathbb{R}_+ (nous allons momentanément oublier a , seul terme éventuellement négatif). Nous en déduisons que (u_n) est monotone, son sens de variation dépendant du signe de $u_2 - u_1$. Or $f(x) - x = (x-1)^2/2 \geq 0$, donc la suite $(u_n)_{n \in \mathbb{R}^*}$ est croissante (récurrence immédiate sur le signe de $u_{n+1} - u_n$). De plus, f étant continue, si la suite converge vers une limite finie, ce sera une solution de $f(x) = x$, à savoir 1. Remarquons encore que les intervalles $[0, 1[$ et $]1, +\infty[$ sont stables par f :

x	$\sqrt{2}$	$\frac{3}{2}$
Signe de $g'(x)$	0	+
Variations de g	$\sqrt{2}$	$\frac{17}{12}$

Nous allons maintenant pouvoir distinguer trois cas :

- si $u_1 \in [0, 1[$ i.e. si $a \in]-1, 1[$, la suite est croissante et majorée par 1 (car $[0, 1]$ est stable par f), donc converge vers 1, unique point fixe de f ;
- si $u_1 = 1$ i.e. si $a = \pm 1$, $(u_n)_{n \in \mathbb{R}^*}$ est constante et égale à 1 ;
- si $u_1 \in]1, +\infty[$ i.e. si $a \in]-\infty, -1[\cup]1, +\infty[$, et si (u_n) convergeait vers un réel ℓ , alors on aurait $\ell \geq u_1 > 1$, ce qui contredit le fait que la seule limite possible de la suite est 1. La suite étant croissante et ne convergeant pas, elle diverge vers $+\infty$.

Deuxième exercice

Il n'est bien sûr pas réaliste de travailler directement avec les a_i . Il suffit de remarquer que chacun des a_i est un entier compris entre 0 et 9.

La suite (u_n) est donc majorée par la suite (v_n) de terme général $v_n = \sqrt{9 + \sqrt{9 + \dots + \sqrt{9}}}$.

La suite (v_n) est donc définie par la relation de récurrence $v_{n+1} = \sqrt{9 + v_n}$ avec $v_1 = 1$ ($\pi \approx 3,1\dots$).

On montre alors facilement par récurrence que la suite (v_n) est majorée par 9. Regardons uniquement l'hérédité :

En supposant qu'il existe au moins un entier n vérifiant $v_n < 9$, on obtient que $9 + v_n < 18$ et donc :

$$v_{n+1} = \sqrt{9 + v_n} < \sqrt{18} < 9$$

Or la suite (u_n) est une suite positive croissante. En effet :

$$\frac{u_{n+1}}{u_n} = \sqrt{\frac{\sqrt{a_1 + \sqrt{a_2 + \dots + \sqrt{a_n + \sqrt{a_{n+1}}}}}}{\sqrt{a_1 + \sqrt{a_2 + \dots + \sqrt{a_n + \sqrt{0}}}}}} \geq 1$$

Elle est de plus majorée par la suite (v_n) qui est elle-même majorée par 9 : la suite (u_n) est donc convergente.

« C'est ahurissant la manière dont le monde des dames de la bourgeoisie parvient à tout aplanir et à tout intégrer, à transformer la complexité et le chaos ambiants en quelque chose de charmant, d'inoffensif et d'aseptisé. »

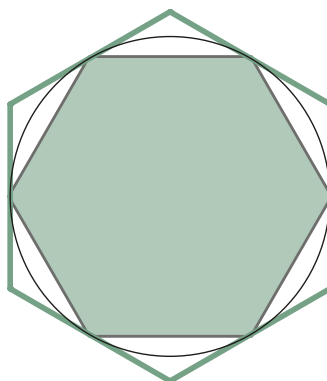
in *Bridget Jones, l'âge de raison* de Helen FIELDING

La notion d'intégrale est bien plus riche que le calcul d'aire vu en Terminale. D'ailleurs il existe plusieurs définitions différentes de l'intégrale : intégrale de RIEMANN, de LEBESGUE... Nous construirons l'intégrale de la manière la plus abordable en partant d'une intuition physique. Nous découvrirons ainsi qu'une fonction intégrable n'est pas forcément continue et n'admet pas forcément de primitive. Nous définirons donc l'intégrale et démontrerons la plupart de ses propriétés sans faire appel aux primitives, puis nous finirons malgré tout par faire le lien entre primitive et intégrale, car ce résultat est un des plus importants de l'analyse. L'approche du cours n'a plus rien à voir avec celle de Terminale. Il est essentiel de travailler chaque démonstration en profondeur pour vous mettre petit à petit dans la peau d'un préparateur.

4.1 Mise en place d'une définition

— Une vieille idée

Vers -250 avant J.-C., l'incontournable ARCHIMÈDE eut l'idée de calculer l'aire d'un disque en l'approximant par deux suites d'aires de polygones la majorant et la minorant respectivement.



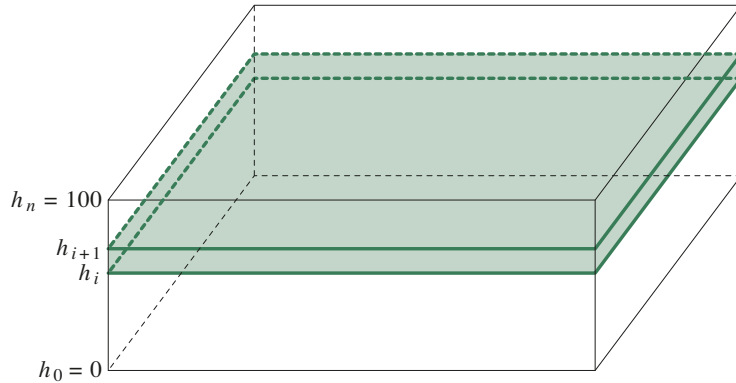
On en est resté là pendant des siècles, car il a fallu attendre le XIX^e pour oser parler de limite. Pourtant, tout était là, comme nous allons le voir en suivant les traces de Gaston DARBOUX (1842-1917).

Une approche physique

Un problème écologique

Le problème avec l'énergie éolienne, c'est le stockage de l'énergie produite. Une des possibilités envisagées consiste à la stocker sous forme d'énergie potentielle de pesanteur.

Rappel : un objet de masse m pouvant tomber d'une altitude h possède l'énergie potentielle de pesanteur $E_p = mgh$ avec $g \approx 10 \text{ m.s}^{-2}$.



On considère un réservoir en forme de pavé droit à base carrée de 1 km de côté et de 100 m de hauteur. On découpe ce pavé en n tranches comme indiqué sur la figure.

Le volume d'une tranche de couleur vaut $1\,000 \times 1\,000 \times (h_{i+1} - h_i) \text{ m}^3$. Or 1 m^3 a une masse de 1 000 kg, donc la masse de la tranche vaut :

$$m = 10^9(h_{i+1} - h_i) \text{ kg}$$

De manière évidente, l'énergie potentielle croît avec la hauteur, donc :

$$10^9(h_{i+1} - h_i) \times 10 \times h_i \leq E_{pi} \leq 10^9(h_{i+1} - h_i) \times 10 \times h_{i+1}$$

De plus, l'énergie potentielle de pesanteur est additive, c'est-à-dire que $E_p = \sum_{i=0}^{n-1} E_{pi}$.

$$\sum_{i=0}^{n-1} 10^{10}(h_{i+1} - h_i)h_i \leq E_p \leq \sum_{i=0}^{n-1} 10^{10}(h_{i+1} - h_i)h_{i+1}$$

On va considérer des tranches de hauteur constante $\frac{100}{n}$, soit $h_i = \frac{100i}{n}$. On obtient donc :

$$\sum_{i=0}^{n-1} 10^{10} \times \frac{10^2 i}{n} \times \frac{10^2}{n} \leq E_p \leq \sum_{i=0}^{n-1} 10^{10} \times \frac{10^2 i}{n} \times \frac{10^2}{n}$$

Et donc :

$$\frac{10^{14}}{n^2} \sum_{i=0}^{n-1} i \leq E_p \leq \frac{10^{14}}{n^2} \sum_{i=0}^{n-1} i + 1$$

Or un résultat classique donne $\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$ et donc $\sum_{i=0}^{n-1} i + 1 = \frac{n(n+1)}{2}$. D'où :

$$\frac{10^{14}}{2} \times \frac{n(n-1)}{n^2} \leq E_p \leq \frac{10^{14}}{2} \times \frac{(n+1)(n+2)}{n^2}$$

L'idée est bien sûr de faire tendre n vers l'infini, car plus n sera grand, plus l'épaisseur des tranches sera petite et donc plus l'erreur commise sur l'approximation inférieure et supérieure de E_p sera petite.

$$\underbrace{\frac{10^{14}}{2} \times \frac{n(n+1)}{n^2}}_{\rightarrow 5 \times 10^{13}} \leq E_p \leq \underbrace{\frac{10^{14}}{2} \times \frac{(n+1)(n+2)}{n^2}}_{\rightarrow 5 \times 10^{13}}$$

Donc, d'après le théorème des gendarmes, on obtient $E_p = 5 \times 10^{13}$ J.

Pour avoir un ordre de grandeur, la capacité du plus grand parc éolien en mer en 2018, situé au large du Kent en Angleterre, est de 630 MW, soit environ $5,5 \times 10^{13}$ J par jour. L'objectif européen est de produire 150 GW en 2030, soit 10^{16} J par jour... Il faudra malgré tout tenir compte des pertes dues à la transformation de l'énergie potentielle de pesanteur en énergie électrique par l'intermédiaire d'une turbine par exemple.

Du cas particulier au cas général

Voici une procédure qui semble efficace. Essayons de la schématiser pour essayer de la généraliser à d'autres situations¹.

- Nous voulions mesurer une grandeur (ici E_p) dépendant d'un intervalle (ici $[0, 100]$) et d'une fonction définie sur l'intervalle (ici la « hauteur ») : notons cette grandeur $\mathcal{J}([a, b], f)$.
- Nous avons alors **subdivisé régulièrement** l'intervalle $[a, b]$ en une suite de petits intervalles $[a_i, a_{i+1}]$ de même taille, avec :

$$a = a_0 < a_1 < a_2 < \dots < a_{n-1} < a_n = b$$

Notons σ_n cette subdivision.

- Nous avons pu déterminer sur chacun de ces intervalles $[a_i, a_{i+1}]$ la « borne inférieure² » (ici h_i) et la « borne supérieure³ » (ici h_{i+1}) de la fonction f : notons les respectivement m_i et M_i . Cela est possible si f est bornée sur $[a, b]$.
- Nous avons alors encadré $\mathcal{J}([a, b], f)$ entre deux sommes que nous noterons $s(\sigma_n)$ et $S(\sigma_n)$ pour obtenir :

$$s(\sigma_n) = \sum_{i=0}^{n-1} (a_{i+1} - a_i) m_i \leq \mathcal{J}([a, b], f) \leq S(\sigma_n) = \sum_{i=0}^{n-1} (a_{i+1} - a_i) M_i$$

- Nous sommes passés à la limite en utilisant le théorème des gendarmes.

Tout ceci est bien beau mais des questions se posent :

- Nous avons choisi une subdivision particulière, mais aurions-nous obtenu le même résultat avec une autre subdivision ?
- La fonction « hauteur » était assez particulière : bornée, croissante, continue, mais aurions-nous obtenu le même résultat avec une fonction moins régulière ?
- Les deux sommes admettaient des limites et en particulier la même limite : est-ce que ce sera toujours le cas ?

1. Nous nous contenterons de l'étude des fonctions d'une variable réelle.

2. Le plus grand des minorants, souvent égal au minimum.

3. Le plus petit des minorants, souvent égal au maximum.

Sommes de DARBOUX

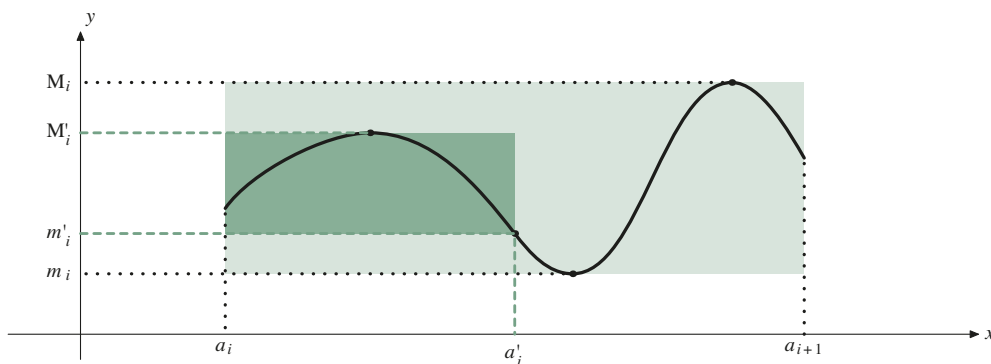
Mise en place de la définition

C'est en 1875 que notre Gaston DARBOUX national présenta sa théorie de l'intégration simplifiant les idées de RIEMANN. Nous n'entrerons pas forcément dans tous les détails mais nous donnerons les idées générales de cette théorie donnée dans une version simplifiée.

On ne considérera dans cette leçon que des SUBDIVISIONS RÉGULIÈRES pour simplifier notre propos, c'est-à-dire des subdivisions où chaque segment élémentaire a la même amplitude.

On parle de subdivision plus fine quand on intercale de nouveaux points « entre les a_i », et alors n augmente.

Un petit dessin montre que les suites $(s(\sigma_n))$ sont croissantes et les suites $(S(\sigma_n))$ sont décroissantes : si on coupe un intervalle $[a_i, a_{i+1}]$ en 2, le « nouveau » minimum est supérieur ou égal au précédent, le nouveau maximum est, lui, inférieur ou égal au précédent.



Il reste à vérifier qu'on a toujours $s(\sigma_k) \leq S(\sigma_p)$ pour tout couple d'entiers (k, p) . En effet, en réunissant les deux subdivisions, on obtient une subdivision plus fine σ_n , et, d'après ce qui précède :

$$s(\sigma_k) \leq s(\sigma_n) \leq S(\sigma_n) \leq S(\sigma_p)$$

Tout ça pour dire que SI $(S(\sigma_n) - s(\sigma_n))$ converge vers 0 quand n tend vers $+\infty$, les suites $(s(\sigma_n))$ et $(S(\sigma_n))$ sont adjacentes et convergent donc vers une même limite qui sera l'intégrale de f au sens de DARBOUX : OUF !

Théorème : critère d'intégrabilité au sens de DARBOUX

Avec les notations précédentes, une fonction f bornée sur $[a, b]$ est dite intégrable au sens de DARBOUX sur $[a, b]$ si $\lim_{n \rightarrow +\infty} (S(\sigma_n) - s(\sigma_n)) = 0$. Alors :

$$\begin{aligned} \mathcal{J}([a, b], f) &= \int_{[a, b]} f(x) dx = \int_a^b f(x) dx = \lim_{n \rightarrow +\infty} \sum_{i=0}^{n-1} (a_{i+1} - a_i) m_i \\ &= \lim_{n \rightarrow +\infty} \sum_{i=0}^{n-1} (a_{i+1} - a_i) M_i \end{aligned}$$

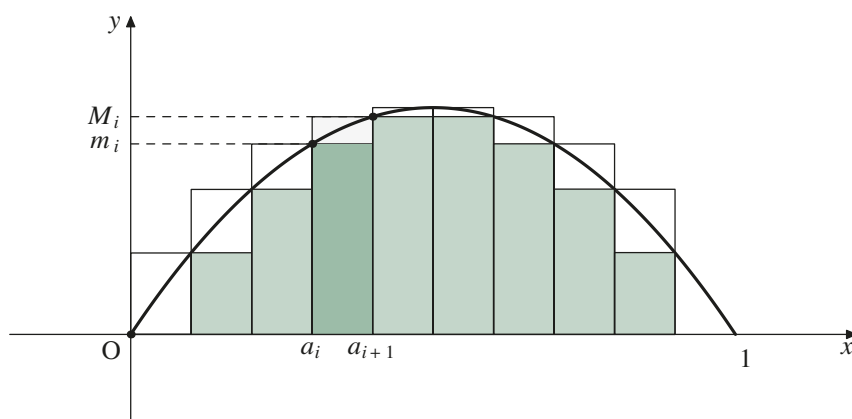
Remarque

Attention au théorème des suites adjacentes ! Le simple fait que $\lim_{n \rightarrow +\infty} (u_n - v_n) = 0$ ne suffit pas à affirmer que les deux suites convergent vers une même limite. Pouvez-vous trouver un contre-exemple ? Dans le théorème précédent, il suffit de vérifier que $\lim_{n \rightarrow +\infty} (S(\sigma_n) - s(\sigma_n)) = 0$ car par construction $(S(\sigma_n))$ est décroissante, $(s(\sigma_n))$ est croissante et $s(\sigma_k) \leq S(\sigma_p)$ pour tout couple d'entiers (k, p) .

Pour votre culture, car vous rencontrerez peut-être ces termes plus tard, on peut montrer qu'une fonction est intégrable au sens de DARBOUX si et seulement si elle est intégrable au sens de RIEMANN.

Interprétation graphique

En fait $(a_{i+1} - a_i) m_i$ représente l'aire du rectangle le plus foncé et $(a_{i+1} - a_i) M_i$ la somme des rectangles foncé et grisé. Donc $s(\sigma_n)$ et $S(\sigma_n)$ sont représentées par les sommes des aires des rectangles inférieurs et supérieurs respectivement : on retrouve alors l'idée d'ARCHIMÈDE pour le cercle.



On fait alors facilement le lien entre intégrale et aire dans le cas d'une fonction positive assez régulière.

Interprétation analytique

Une petite remarque qui s'avérera primordiale l'an prochain : on a « coincé » la fonction f entre deux fonctions « en escalier » pour lesquelles il est facile de définir une aire ou une intégrale. Si ce « coincement » est suffisamment régulier, on pourra passer des propriétés de l'intégrale des fonctions en escalier facile à calculer aux propriétés de l'intégrale de la fonction f .

4.2 Quelles sont les fonctions intégrables ?

Un exemple de référence très simple

La fonction f constante vérifiant $f(x) = 1$ pour tout $x \in [a, b]$ a une intégrale facilement calculable. En effet, pour tout $i \in \llbracket 0, n \rrbracket$, on a $m_i = M_i = 1$, donc pour tout entier n on a :

$$s(\sigma_n) = S(\sigma_n) = \sum_{i=0}^{n-1} (a_{i+1} - a_i) = a_n - a_0 = b - a$$

On a donc $\lim_{n \rightarrow +\infty} (S(\sigma_n) - s(\sigma_n)) = 0$: la fonction est bien (DARBOUX-)intégrable sur $[a, b]$ et :

$$\int_a^b 1 \cdot dx = b - a$$

Dans la suite, « intégrable » sous-entendra « DARBOUX-intégrable ».

— Existe-t-il des fonctions bornées qui ne sont pas intégrables sur $[a, b]$?

Ce genre de fonction ne se rencontre pas tous les jours dans la vie d'un lycéen.

L'exemple « le plus simple » est la fonction f définie sur $[0, 1]$ par :

$$f : x \mapsto \begin{cases} 1 & \text{si } x \in \mathbb{Q} \\ 0 & \text{sinon} \end{cases}$$

Sa représentation graphique est étrange car, « à l'œil nu », on a l'impression de voir deux traits continus parallèles. Il n'est pas possible à notre niveau de rentrer dans les détails, mais sachez seulement que de telles fonctions existent.

— Fonctions en escalier

Nous n'entrerons pas dans les détails, mais il est assez immédiat de se rendre compte que les fonctions en escalier sont intégrables sur de bons intervalles : à partir d'un moment, le pas des subdivisions devient plus petit que la largeur des marches. Cela constitue une première différence d'avec la Terminale : on peut calculer des intégrales de fonctions non continues et heureusement, car de telles fonctions se rencontrent souvent en physique.

Par exemple, la fonction partie entière est intégrable sur tout intervalle borné de \mathbb{R} .

— Fonctions monotones

Plus impressionnant :

Théorème

Les fonctions monotones sur $[a, b]$ sont intégrables sur $[a, b]$.

Avec les notations habituelles, on a :

$$S(\sigma_n) - s(\sigma_n) = \sum_{i=0}^{n-1} (a_{i+1} - a_i)(M_i - m_i)$$

On peut supposer par exemple que f est croissante sur $[a, b]$. Alors $m_i = f(a_i)$ et $M_i = f(a_{i+1})$ et donc :

$$\begin{aligned} S(\sigma_n) - s(\sigma_n) &= \sum_{i=0}^{n-1} (a_{i+1} - a_i)(f(a_{i+1}) - f(a_i)) \\ &\leq \max(a_{i+1} - a_i)_{0 \leq i \leq n-1} \sum_{i=0}^{n-1} (f(a_{i+1}) - f(a_i)) \\ &\leq \max(a_{i+1} - a_i)_{0 \leq i \leq n-1} (f(b) - f(a)) \end{aligned}$$

Puisqu'on fait tendre le nombre n de points de la subdivision vers l'infini, on peut choisir le pas maximum arbitrairement petit, par exemple égal à $\frac{1}{n(f(b) - f(a))}$. Alors :

$$S(\sigma_n) - s(\sigma_n) \leq \frac{1}{n(f(b) - f(a))} \times (f(b) - f(a)) = \frac{1}{n}$$

Nous avons donc bien $\lim_{n \rightarrow +\infty} (S(\sigma_n) - s(\sigma_n)) = 0$ et f est bien intégrable sur $[a, b]$.

Encore une classe de fonctions intégrables supplémentaires par rapport à la Terminale. Mais il faut bien retrouver ses habitudes, donc...

Fonctions continues

Théorème

Les fonctions continues sur $[a,b]$ sont intégrables sur $[a,b]$.

L'idée de la démonstration vient du résultat suivant : pour une fonction continue, plus x et x' sont proches, plus leurs images le sont aussi. Donc pour une subdivision de pas suffisamment petit, on pourra majorer chaque $M_i - m_i$ par ce qu'on veut. La démonstration étant assez technique, nous ne la verrons pas dans le détail.

4.3 Propriétés de l'intégrale

Relation de CHASLES

C'est le cœur du problème : cette relation entraîne toutes les autres.

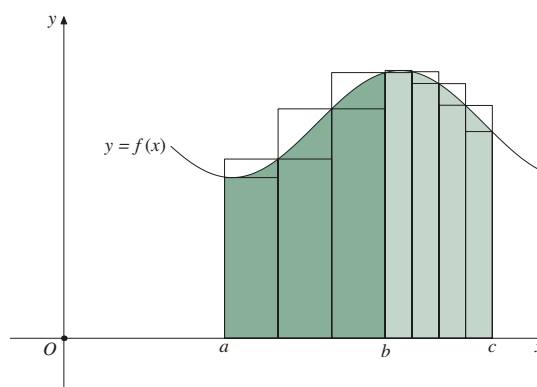
Il s'agit de montrer :

Théorème : relation de CHASLES

Soit f une fonction intégrable sur $[a,c]$, sur $[b,c]$ et sur $[a,b]$, alors :

$$\int_a^b f(x) dx + \int_b^c f(x) dx = \int_a^c f(x) dx$$

Pour s'en convaincre **intuitivement**, il suffit de se dire que si les sommes de DARBOUX convergent sur $[a,b]$ et sur $[b,c]$, alors elles vont converger sur $[a,c]$ vers la somme de ces limites en combinant les subdivisions : c'est une histoire de somme de limites finies. Nous n'entrerons pas plus dans les détails (si nous pensons « aire », le cas des fonctions positives devient assez naturel).



On peut démontrer ainsi les propriétés que vous connaissez bien : l'idée à retenir, même si vous verrez des définitions différentes l'an prochain, c'est que les propriétés de l'intégrale s'obtiennent **par passage à la limite** de sommes *discrètes*, c'est pourquoi elles ont posé problème aussi longtemps : il a fallu attendre plusieurs siècles pour avoir une définition correcte des limites. En ce qui vous concerne, vous vous contenterez de quelques mois...

Rappelons donc ces propriétés que vous pourrez chercher à démontrer :

Théorème : propriétés de l'intégrale

Avec f et g des fonctions intégrables sur $[a, b]$:

- $\int_a^a f(x) dx = 0$
- $\int_a^b f(x) dx = -\int_b^a f(x) dx$
- $\int_a^b [\lambda f(x) + \mu g(x)] dx = \lambda \int_a^b f(x) dx + \mu \int_a^b g(x) dx$ avec λ et μ des réels : c'est la linéarité de l'intégrale.
- $f(x) \geq 0 \Rightarrow \int_a^b f(x) dx \geq 0$
- $f(x) \leq g(x) \Rightarrow \int_a^b f(x) dx \leq \int_a^b g(x) dx$: c'est la croissance de l'intégrale.
- $\left| \int_a^b f(x) dx \right| \leq \int_a^b |f(x)| dx$: c'est l'inégalité triangulaire appliquée aux intégrales.

4.4 Valeur moyenne

Définition

La moyenne d'un nombre discret de valeurs est facile à obtenir ; il suffit d'additionner ces valeurs et de les diviser par leur nombre :

$$m_n(f) = \frac{f(a_0) + f(a_1) + \dots + f(a_{n-1})}{n}$$

On pense tout naturellement à passer à la limite et à remplacer la somme discrète par une intégrale. Mais attention : ceci n'est valable que si la subdivision n'est pas trop irrégulière. On pourrait imaginer en effet que la subdivision prend une infinité de valeurs entre a et $(b-a)/2$ et aucune ailleurs, alors $m_n(f)$ ne pourrait représenter une approximation convenable d'une moyenne. Nous continuerons donc à considérer des subdivisions régulières.

Nous admettons donc le résultat suivant :

Théorème : valeur moyenne

Soit f une fonction continue sur $[a, b]$ et $(a_k)_{0 \leq k \leq n-1}$ une subdivision **régulière** de u , alors :

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{k=0}^{n-1} f(a_k) = \frac{1}{b-a} \int_a^b f(x) dx$$

On appelle $\mu = \frac{1}{b-a} \int_a^b f(x) dx$ la valeur moyenne de f sur $[a, b]$.

Vous pouvez retenir également qu'intuitivement la valeur moyenne est une sorte de somme des valeurs de $f(x)$ affectées des coefficients dx , le tout divisé par la somme des coefficients

dx . Or pour calculer $\int_a^b 1 dx$, on remarque que :

$$\sum_{k=0}^{n-1} (a_{i+1} - a_i) m_i = \sum_{k=0}^{n-1} (a_{i+1} - a_i) M_i = \sum_{k=0}^{n-1} (a_{i+1} - a_i) \times 1 = b - a$$

quelle que soit la subdivision, donc :

$$\int_a^b dx = b - a$$

et on retrouve intuitivement le résultat.

— Est-ce que la valeur moyenne est une valeur prise par la fonction ?

Ça n'a rien d'évident *a priori*, puisque vous pouvez avoir 15 de moyenne en n'ayant jamais eu de note égale à 15 (14 et 16 par exemple).

C'est également faux pour une fonction en escalier valant 14 sur $[0, 1/2]$ et 16 sur $[1/2, 1]$.

Et pourtant c'est vrai si nous précisons que la fonction est CONTINUE comme nous allons le démontrer : comme quoi le passage à la limite du discret au continu présente quelques dangers !

Comme notre fonction f est continue sur $[a, b]$, elle est bornée, donc il existe deux réels m et M et deux éléments x_m et x_M de $[a, b]$ tels que :

$$m = f(x_m) \leq f(x) \leq M = f(x_M)$$

Alors, on obtient successivement :

$$\begin{aligned} \int_a^b m dx &\leq \int_a^b f(x) dx \leq \int_a^b M dx \\ m(b-a) &\leq \int_a^b f(x) dx \leq M(b-a) \end{aligned}$$

et finalement :

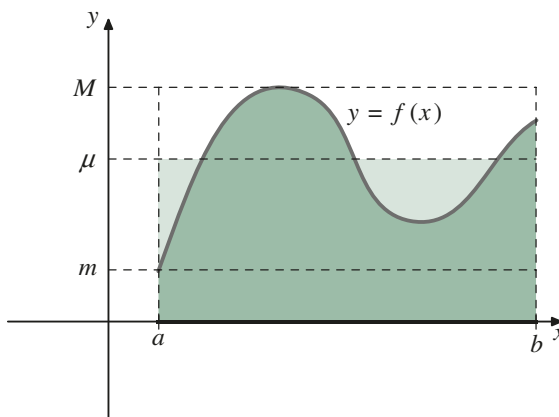
$$m \leq \mu \leq M$$

Donc μ appartient à $[f(x_m), f(x_M)]$, intervalle image de la fonction f continue sur $[a, b]$: il existe donc un réel $x_0 \in [a, b]$ tel que $f(x_0) = \mu$.

Théorème : formule de la moyenne

Soit f une fonction CONTINUE sur $[a, b]$, alors il existe un réel $x_0 \in [a, b]$ tel que :

$$f(x_0) = \mu = \frac{1}{b-a} \int_a^b f(x) dx$$



4.5 Primitive et intégrale

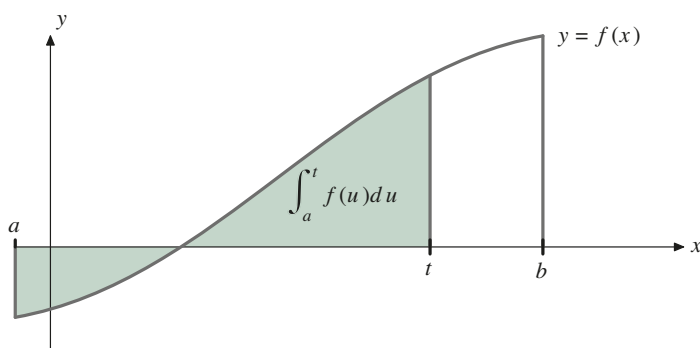
Intégrale fonction de sa borne supérieure

Considérons une fonction f que nous supposons continue sur un intervalle $[a, b]$ pour simplifier notre propos.

On peut donc définir une fonction S sur $[a, b]$ telle que :

$$S : \begin{array}{l} [a, b] \rightarrow \mathbb{R} \\ t \mapsto \int_a^t f(u) du \end{array}$$

Du point de vue graphique, on peut interpréter $S(t)$ comme l'aire algébrique du domaine coloré :



Comment retrouver f connaissant $\int_a^b f(u) du$?

Rappelons d'abord la définition d'une primitive :

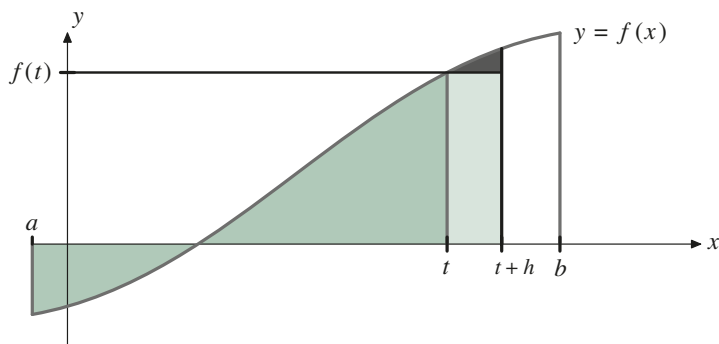
Définition : primitive

Soit f et F deux fonctions définies sur un intervalle I . Alors F est une **primitive** de f lorsque F est dérivable sur I et que $F' = f$.

Nous allons donc essayer de retrouver f connaissant S .

Approche intuitive

On fixe t dans $[a, b[$ et on considère un « petit » réel positif h . Observons ce qui se passe sur le « petit » intervalle $[t, t + h]$



On « voit » que, pour h petit, l'aire du petit « triangle » noir est « petite » devant l'aire du rectangle clair situé en dessous. Cela donne :

$$S(t+h) - S(t) = h \times f(t) + \text{aire noire} \simeq h \times f(t)$$

et donc :

$$\frac{S(t+h) - S(t)}{h} \simeq f(t)$$

Ainsi, le taux d'accroissement de S entre t et $t+h$ « ressemble » à $f(t)$ quand h est « petit ».

On « sent » donc que S est dérivable en t et que $S'(t) = f(t)$ et donc que S est une primitive de f , ce qui crée le lien bien connu entre primitive et intégrale.

Il reste à démontrer cette intuition.

Preuve de notre intuition

Nous allons utiliser la formule de la moyenne vue précédemment appliquée à la fonction f continue sur $[t, t+h]$. Cela donne qu'il existe un réel t_0 appartenant à $[t, t+h]$ tel que :

$$\frac{1}{t+h-t} \int_t^{t+h} f(u) du = \frac{1}{h} \times (S(t+h) - S(t)) = f(t_0)$$

C'est-à-dire que :

$$\frac{S(t+h) - S(t)}{h} = f(t_0)$$

Or f est continue sur $[t, t+h]$, donc quand h tend vers 0, $f(t_0)$ tend vers $f(t)$ et donc :

$$\lim_{h \rightarrow 0} \frac{S(t+h) - S(t)}{h} = S'(t) = f(t)$$

Il resterait à faire la même démonstration pour h négatif.

On obtient donc le résultat suivant :

Théorème fondamental

Soit f une fonction CONTINUE sur $[a, b]$ dans \mathbb{R} et soit

$$S : [a, b] \rightarrow \mathbb{R} \\ t \mapsto \int_a^t f(u) du$$

alors S est dérivable sur $[a, b]$ et $S' = f$.

— Comment calculer une intégrale à l'aide d'une primitive ?

Rappelons une propriété bien connue :

Théorème

Deux primitives d'une même fonction définie sur un intervalle diffèrent d'une constante.

En effet, si F et G sont deux primitives de f sur I , alors $F' = G' = f$ et donc $F' - G' = (F - G)' = 0$. La fonction $F - G$ est donc constante sur I et il existe un réel k tel que $F(x) - G(x) = k$ pour tout $x \in I$.

Soit donc F une primitive de f . Comme la fonction S est elle aussi une primitive de f , il existe donc un réel k constant tel que $S(t) = F(t) + k$ pour tout $t \in [a, b]$. Alors :

$$\int_a^b f(u) du = S(b) - S(a) = (F(b) + k) - (F(a) + k) = F(b) - F(a)$$

Théorème : intégrale et primitive

Soit F UNE PRIMITIVE d'une fonction f continue sur $[a, b]$, alors :

$$\int_a^b f(u) du = F(b) - F(a)$$

Intégration par parties

L'intégration par parties est une des plus importantes méthodes d'intégration.

Son principe peut être vu de manière assez intuitive.

Si u et v sont deux fonctions dérivables sur $[a, b]$, alors $(uv)' = u'v + uv'$,
soit encore $u'v = (uv)' - uv'$ donc :

$$\int_a^b u dv = [uv]_a^b - \int_a^b v du$$

Par exemple, pour calculer $I = \int_0^1 xe^x dx$:

$$\begin{cases} u(x) = x & u'(x) = 1 \\ v'(x) = e^x & v(x) = e^x \end{cases}$$

$$I = [xe^x]_0^1 - \int_0^1 e^x dx = [xe^x]_0^1 - [e^x]_0^1 = (e - 0) - (e - 1) = 1$$

```
In [298]: sy.integrate(x * sy.exp(x), (x, 0, 1))
Out [298]: 1
```

4.6 Exercices

Test 4.1 (Calcul de primitives - ♥)

Complétez le tableau suivant

f	Je pose $u =$	Alors $u' =$	Forme de f en fonction de u et u'	Une primitive de f est
$\frac{x+1}{(x^2+2x+2)^3}$				
$\frac{\cos(2x)}{(3+\sin(2x))^3}$				

f	Je pose $u =$	Alors $u' =$	Forme de f en fonction de u et u'	Une primitive de f est
$\frac{(\ln x)^2}{x}$				
$x\sqrt{x^2-1}$				
$16\frac{e^x}{1+2e^x}$				
$\frac{e^{1/x}}{x^2}$				
$\frac{1}{x \ln x}$				
$xe^{-x^2/2}$				
$\frac{e^x - e^{-x}}{e^x + e^{-x}}$				
$\cos(x) \sin^5(x)$				

Test 4.2 (Quiz - ♥)

- Vrai ou faux ?** L'intégrale d'une fonction continue et impaire est nulle.
- Vrai ou faux ?** Si $\int_{-2}^2 f(t) dt = 0$, alors f est impaire.
- Trouvez une fonction paire, non identiquement nulle sur $[-2, 2]$, telle que $\int_{-2}^2 f(t) dt = 0$.
- Vrai ou faux ?** Si $\lim_{x \rightarrow +\infty} f(x) = 0$, alors $\int_1^x f(t) dt$ admet une limite finie quand x tend vers $+\infty$.
- Trouvez une fonction telle que $\lim_{x \rightarrow +\infty} f(x) = 0$ et $\lim_{x \rightarrow +\infty} \int_1^x f(t) dt = +\infty$.
- Trouvez une fonction telle que $\lim_{x \rightarrow +\infty} f(x) = 0$ et $\lim_{x \rightarrow +\infty} \int_1^x f(t) dt = 32$.
- Vrai ou faux ?** Soit u un réel strictement positif, alors $\int_0^u \lfloor x \rfloor dx \in \mathbb{N}$, $\lfloor x \rfloor$ désignant la partie entière de x .
- Trouvez une fonction telle que $\left| \int_a^b f(t) dt \right| = \int_a^b |f(t)| dt$.
- Trouvez une fonction f telle que $\left| \int_a^b f(t) dt \right| < \int_a^b |f(t)| dt$.
- Trouvez une condition nécessaire et suffisante sur la fonction f continue sur $[a, b]$ pour que $\left| \int_a^b f(t) dt \right| = \int_a^b |f(t)| dt$.

11. Vrai ou faux ? $\int_2^3 xt^2 dt = \int_2^3 xt^2 dx$

12. Vrai ou faux ? $\int_2^3 xt^2 dt = \int_2^3 x^2 t dx$

13. Trouvez deux fonctions f et g continues sur $[1,2]$, distinctes, telles que $\int_1^2 f(t) dt = \int_1^2 g(u) du$.

14. Vrai ou faux ? Si f est bornée sur $[a,b]$, alors la fonction $x \mapsto \int_a^x f(u) du$ l'est aussi.

15. Vrai ou faux ? Si f est croissante sur $[a,b]$, alors la fonction $x \mapsto \int_a^x f(u) du$ l'est aussi.

16. Déterminez une fonction polynôme de degré supérieur ou égal à 2 et dont la valeur moyenne sur $[-2; 2]$ est 0.

Test 4.3 (Calcul de primitives - ♥)

1. $f_1(x) = \frac{\ln x}{x}$

2. $f_2(x) = \tan x$

Test 4.4 (Calcul d'intégrales - ♥)

1. $I_1 = \int_1^2 3(x-1)^2 \ln(x) dx$

2. $I_2 = \int_0^\pi \sin(x) e^{-x} dx$

3. $I_3 = \int_0^{\pi/4} \frac{x}{\cos^2(x)} dx$

4. $I_4 = \int_1^3 \frac{x+1}{x} (\ln(x) + x)^3 dx$

5. $I_5 = \int_{-1}^1 \frac{dx}{1 + e^{-x}}$

6. $I_6 = \int_{-1}^1 \sqrt{x^2 + 1} dx$

Test 4.5 (Primitives des puissances de cos et sin - ♥)

1. Calculez $\cos x$ en fonction de e^{ix} et e^{-ix} .

2. Dédisez-en une expression de $\cos^5 x$ comme combinaison linéaire de $\cos(kx)$ avec $k \in \llbracket 0, 5 \rrbracket$.

3. Calculez $\int_{\pi/6}^{\pi/4} \cos^5 t dt$

Test 4.6 (Fonction DARBOUX-intégrable - ♥♥)

Démontrez que la fonction $f: x \mapsto x$ est intégrable au sens de DARBOUX sur l'intervalle $[0,1]$ et retrouvez alors sa valeur bien connue.

Test 4.7 (Limites de suites définies par une intégrale - ♥♥♥)

1. À l'aide de majorations ou d'encadrements, déterminez la limite quand n tend vers $+\infty$ de :

(a) $\int_0^{\pi} \frac{\sin x}{x+n} dx$

(b) $\int_0^1 \frac{x^n}{1+x^2} dx$

(c) $\int_0^2 \frac{x^{2n}}{1+x^n} dx$

(d) $\int_1^{1+1/n} \sqrt{1+x^n} dx$

(e) $\int_0^{\pi} \frac{n \sin x}{x+n} dx$ en commençant par majorer $\int_0^{\pi} \left(\frac{n \sin x}{x+n} - \sin x \right) dx$

2. Soit $f: [0,1] \rightarrow \mathbb{R}$ dérivable et de dérivée continue. Montrez que

$$\lim_{n \rightarrow +\infty} n \int_0^1 x^n f(x) dx = f(1)$$

à l'aide d'une intégration par parties.

Test 4.8 (Transformée de Laplace de la fonction rampe - ♥♥♥)

On pose $I(t) = \int_0^t x e^{-px} dx$ où p est un paramètre réel. Calculez $I(t)$ puis $\lim_{t \rightarrow +\infty} I(t)$ en discutant selon les valeurs du paramètre p .

Test 4.9 (Transformée de Fourier du signal $s(t) = \cos(\pi t)\Pi(t)$ - ♥♥♥)

Il s'agit de calculer l'intégrale $X(s) = 2 \int_{-1/2}^{1/2} \cos(\pi t) \cos(2\pi f t) dt$ où f est un paramètre positif.

Test 4.10 (Lemme de Gronwall - ♥♥♥)

Soit $f: [0, +\infty[\rightarrow \mathbb{R}$ continue telle que, pour tout $x \geq 0$

$$0 \leq f(x) \leq \int_0^x f(t) dt$$

Montrez que f est identiquement nulle. On pourra introduire la fonction

$$g: x \mapsto e^{-x} \int_0^x f(t) dt$$

et étudier ses variations. Vous montrerez en particulier que g est identiquement nulle.

Test 4.11 (Fonction définie par une intégrale - ♥♥♥)

Soit f la fonction $x \mapsto \int_x^{3x} \frac{\cos t}{t} dt$. Étudiez f : ensemble de définition, parité, sens de variation, limites aux bornes, tracé. On obtiendra les limites avec **Python**.

Test 4.12 (Développement en série entière de $\ln(1 - x)$ - ♥♥♥♥)

Soit $x \in [0, 1[$ et $n \in \mathbb{N}$.

1. Montrez que, pour tout $t \in [0, x]$, on a

$$\left| \frac{1}{1-t} - \sum_{k=0}^n t^k \right| \leq \frac{t^{n+1}}{1-x}$$

2. Soit $x \in [-1, 0]$. Montrez que, pour tout $t \in [x, 0]$, $\left| \frac{1}{1-t} - \sum_{k=0}^n t^k \right| \leq |t|^{n+1}$.

3. Soit $x \in [-1, 1[$. Déduisez des questions précédentes que

$$\lim_{n \rightarrow +\infty} \sum_{k=0}^n \frac{x^{k+1}}{k+1} = -\ln(1-x)$$

On obtient ainsi ce qu'on appelle un développement en série entière de $\ln(1 - x)$: on « remplace » une fonction compliquée par une sorte de « polynôme infini » à coefficients entiers. Cela permet dans certains cas de simplifier des calculs (si si !). Vous verrez ça plus tard.

Tableau des dérivées usuelles

$f(x) =$	$f'(x) =$
$u + v$	$u' + v'$
$\lambda u, \lambda \in \mathbb{R}$	$\lambda u'$
uv	$u'v + uv'$
$\frac{u}{v}$	$\frac{u'v - uv'}{v^2}$
$u \circ v$	$v' \times u' \circ v$
e^u	$u' e^u$
$\sin u$	$u' \cos u$
$\cos u$	$-u' \sin u$
$\tan u$	$u'(1 + \tan^2 u) = \frac{u'}{\cos^2 u}$
$\ln u $	$\frac{u'}{u}$
$u^k, k \in \mathbb{Q}$	$ku^{k-1}u'$

Coups de pouce

Coups de pouce pour le test 4.1

Aidez-vous du tableau de la page précédente si besoin.

Coups de pouce pour le test 4.2

1. Discutez selon la nature de l'intervalle d'intégration.
2. cf. question suivante...
3. cf. question précédente...
4. Considérez $f : t \mapsto 1/t$.
5. cf. question précédente...
6. Cherchez une fonction qui « s'aplatit » plus vite sur l'axe des abscisses que $t \mapsto 1/t$.
7. Faites un dessin : la courbe représentative de la partie entière est un escalier : vous pouvez peut-être réduire la largeur de la dernière marche...
8. Le cours dit que $\left| \int_a^b f(t) dt \right| \leq \int_a^b |f(t)| dt$, c'est donc qu'il doit y avoir des cas où il y a égalité et d'autres où l'inégalité est stricte.
9. Une fonction qui n'est pas de signe constant sur $[a, b]$...
10. Il s'agit de démontrer dans le cas général le résultat observé dans les deux cas particuliers précédents : il y a égalité si, et seulement si, f est de signe constant sur l'intervalle d'intégration. Vous serez peut-être amené(e) à utiliser le fait qu'une fonction CONTINUE sur un intervalle dont l'intégrale sur cet intervalle est nulle est elle-même identiquement nulle sur cet intervalle.
11. Attention à la variable d'intégration.
12. C'est plus ressemblant qu'avant mais...
13. Pensez par exemple à la valeur moyenne.
14. L'intégration conserve l'ordre.
15. Pensez à l'intégrale comme à une somme de « bâtons » : la somme sera croissante si les bâtons pointent vers le haut...
16. Pensez à une fonction impaire...

Coups de pouce pour le test 4.3

Aidez-vous du tableau de la page précédente et retrouvez des formes du type u'/u , $u' \times u$, etc.

Coups de pouce pour le test 4.4

1. Ipp. (Intégration par parties)
2. 2 Ipp.
3. Ipp.
4. $u' \times u^3$.
5. u'/u .
6. Faites un dessin...

Coups de pouce pour le test 4.5

Pensez aux formules d'EULER et au triangle de PASCAL pour le calcul de $(a + b)^5$.

Coup de pouce pour le test 4.6

Revoyez le cours. Calculez les deux sommes de DARBOUX (s et S) pour la subdivision de $[0, 1]$ en n sous-intervalles égaux de largeur $1/n$ en vous rappelant aussi que f est strictement croissante sur $[0, 1]$.

Coups de pouce pour le test 4.7

1. Regardez bien l'intervalle d'intégration. Ici on utilise $0 \leq \sin x \leq 1$ et $x + n \geq \pi + n$ sur $[0, \pi]$.
2. Éliminer les x gênants et garder des n pour la limite.
3. Ici, c'est plus rusé. Il faut « sentir » qu'entre 1 et 2, « ça » va diverger car la fonction à intégrer « ressemble » à x^n qui est « plus fort » que n . Donc il faudrait majorer la fonction.
4. La fonction $x \mapsto \sqrt{1+x^n}$ est croissante.
5. Encore une fois, des majorations trop brutales ne nous mènent à rien. Remarquons toutefois que $\frac{n \sin x}{x+n}$ tend vers $\sin x$ quand n tend vers $+\infty$ ce qui nous incite à nous occuper de leur différence.

Coups de pouce pour le test 4.8

Ipp...

Coups de pouce pour le test 4.9

On utilise la formule

$$\cos a \cos b = \frac{1}{2} (\cos(a+b) + \cos(a-b))$$

pour linéariser l'expression à intégrer.

Coups de pouce pour le test 4.10

N'oubliez pas que la dérivée de $x \mapsto \int_0^x f(t) dt$ est $f(x)$.

Coups de pouce pour le test 4.11

Pour la dérivée, on pourra écrire

$$f(x) = \int_x^1 \frac{\cos t}{t} dt + \int_1^{3x} \frac{\cos t}{t} dt$$

par exemple.

Coups de pouce pour le test 4.12

1. Remarquons tout d'abord que $\sum_{k=0}^n t^k = \frac{1-t^{n+1}}{1-t}$ puisque $t < 1$.
2. Sur $[-1, 0[$, $|1-t| \geq 1$.
3. On « sent » qu'une intégration est passée par là. Découpons à nouveau notre problème en 2.

Solutions et commentaires

Solution du test 4.1

f	Je pose $u =$	Alors $u' =$	Forme de f en fonction de u et u'	Une primitive de f est
$\frac{x+1}{(x^2+2x+2)^3}$	(x^2+2x+2)	$2x+2$	$-\frac{1}{4}(-2)u'u^{-2-1}$	$-\frac{1}{4}u^{-2} = -\frac{1}{4(x^2+2x+2)^2}$
$\frac{\cos(2x)}{(3+\sin(2x))^3}$	$3+\sin(2x)$	$2\cos(2x)$	$-\frac{1}{4}(-2)u'u^{-2-1}$	$-\frac{1}{4}u^{-2} = -\frac{1}{4(3+\sin(2x))^2}$
$\frac{(\ln x)^2}{x}$	$\ln(x)$	$\frac{1}{x}$	$\frac{1}{3}3u'u^{3-1}$	$\frac{1}{3}u^3 = \frac{1}{3}(\ln(x))^3$
$x\sqrt{x^2-1}$	x^2-1	$2x$	$\frac{1}{3}\frac{3}{2}u'u^{3/2-1}$	$\frac{1}{3}u^{3/2} = \frac{1}{3}\sqrt{x^2-1}^3$
$16\frac{e^x}{1+2e^x}$	$1+2e^x$	$2e^x$	$\frac{8u'}{u}$	$8\ln u = 8\ln(1+2e^x)$
$\frac{e^{1/x}}{x^2}$	$\frac{1}{x}$	$-\frac{1}{x^2}$	$-u'e^u$	$-e^u = -e^{1/x}$
$\frac{1}{x\ln x}$	$\ln x$	$\frac{1}{x}$	$\frac{u'}{u}$	$\ln u = \ln \ln x $
$xe^{-x^2/2}$	$-\frac{x^2}{2}$	$-x$	$-u'e^u$	$-e^u = -e^{-x^2/2}$
$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$e^x + e^{-x}$	$e^x - e^{-x}$	$\frac{u'}{u}$	$\ln u = \ln(e^x + e^{-x})$
$\cos(x)\sin^5(x)$	$\sin x$	$\cos x$	$\frac{1}{6}(6u'u^{6-1})$	$\frac{1}{6}u^6 = \frac{1}{6}\sin^6 x$

Solution du test 4.2

1. La question est (volontairement...) mal posée. C'est faux sur un intervalle quelconque, c'est vrai si l'intervalle est centré en 0. Nous n'avons pas l'outil de changement de variable à notre disposition, mais des considérations d'aires suffisent. En effet, $\int_{-a}^a f(t) dt = \int_{-a}^0 f(t) dt + \int_0^a f(t) dt$. Or les aires algébriques de ces deux intégrales sont opposées par des considérations de symétrie, donc leur somme est nulle.

Il faut donc être vigilant lors de la lecture d'un énoncé.

2. Faux : l'énoncé de la question suivante nous met la puce à l'oreille. Il suffit de considérer une fonction paire de période 2, par exemple $f : x \mapsto \cos(\pi x)$. Vérifiez qu'elle convient.

3. cf. question précédente.

4. On pourrait le penser en effet, car l'aire comprise entre l'axe des abscisses et la courbe devient « très petite ». Et pourtant cela peut être faux, en prenant par exemple $f(t) = \frac{1}{t}$, alors $\int_1^x f(t) dt = \ln x$ qui tend vers $+\infty$ en $+\infty$. Malgré tout, ça peut être vrai, comme par exemple avec la fonction $x \mapsto \frac{1}{x^2}$. Vous découvrirez ça l'an prochain.

5. cf. question précédente.

6. Essayez $t \mapsto \frac{32}{t^2}$.

7. Faux : Prenez $u = 1,5$. Alors $\int_0^{1,5} \lfloor x \rfloor dx = \int_0^1 0 dx + \int_1^{1,5} 1 dx = 0,5 \notin \mathbb{N}$

8. Pour l'égalité, il suffit évidemment de prendre une fonction de signe constant, par exemple $t \mapsto t^2$ sur n'importe quel intervalle $[a, b]$.

9. On pense à une fonction qui n'est pas de signe constant, par exemple $t \mapsto t$ sur $[-1, 1]$

10. Essayons de démontrer notre intuition introduite dans le coup de pouce. Avec des notations simplifiées on obtient

$$\left| \int_a^b f \right| = \int_a^b |f| \Rightarrow \int_a^b |f| = \pm \int_a^b f \text{ d'où } \int_a^b (|f| - f) = 0 \text{ ou } \int_a^b (|f| + f) = 0$$

Or $|f| - f$ et $|f| + f$ sont continues et positives, donc $|f| - f = 0$ ou $|f| + f = 0$, c'est-à-dire que f est de signe constant sur $[a, b]$.

11. Faux : $\int_2^3 xt^2 dt = x \left[\frac{t^3}{3} \right]_2^3 = \frac{19x}{3}$ et $\int_2^3 xt^2 dx = t^2 \left[\frac{x^2}{2} \right]_2^3 = \frac{5t^2}{2}$

12. Faux : la première intégrale vaut $\frac{19x}{3}$ et la deuxième $\frac{19t}{3}$.

13. C'est le principe même de la valeur moyenne. Par exemple $\int_1^2 (x-1) dx = \int_1^2 \frac{1}{2} du$.

14. Vrai : soit $x \in [a, b]$. Comme f est bornée sur $[a, b]$, f est aussi bornée sur $[a, x]$, c'est-à-dire qu'il existe deux réels m et M tels que $m \leq f(t) \leq M$ pour tout $t \in [a, x]$ et donc $\int_a^x m dt \leq \int_a^x f(t) dt \leq \int_a^x M dt$, c'est-à-dire $m(b-x) \leq \int_a^x f(t) dt \leq M(b-x)$, et finalement

$$m(b-a) \leq \int_a^x f(t) dt \leq Mb$$

15. Faux : là ce n'est plus pareil. Ce qui assure la croissance de la fonction définie par l'intégrale, c'est la positivité de f . Il suffit donc de choisir une fonction croissante et négative sur $[a, b]$, par exemple $t \mapsto t-2$ sur $[0, 1]$ car on aura en notant « à la physicienne »

$$\frac{d}{dx} \int_0^x (t-2) dt = x-2 < 0 \text{ pour tout } x \in [0, 1]$$

16. La fonction $x \mapsto x^3$ est impaire donc son intégrale sur $[-2, 2]$ est nulle et par suite sa valeur moyenne aussi.

Solution du test 4.3

1. $f_1(x) = u'(x) \times u(x)$ avec $u(x) = \ln x$, donc $\int f_1(x) dx = \frac{1}{2} (\ln t)^2$.

2. $f_2(x) = \frac{u'(x)}{u(x)}$ avec $u(x) = \cos x$, donc $\int f_2(x) dx = -\ln(|\cos t|)$.

Solution du test 4.4

1. Une intégration par parties s'impose $\begin{cases} u(x) = \ln x & u'(x) = \frac{1}{x} \\ v'(x) = 3(x-1)^2 & v(x) = (x-1)^3 \end{cases}$ et on obtient

$$I_1 = \left[(x-1)^3 \ln x \right]_1^2 - \int_1^2 \frac{x^3 - 3x^2 + 3x - 1}{x} dx = \ln 2 - \int_1^2 x^2 - 3x + 3 - \frac{1}{x} = 2 \ln 2 - \frac{5}{6}$$

```
In [1]: import sympy as sy
In [2]: x = sy.symbols('x')
In [3]: f = lambda x : 3*(x - 1)**2 * sy.log(x)
In [4]: sy.integrate(f(x), (x, 1, 2))
Out[4]: -5/6 + 2*log(2)
```

2. Avec sin et exp, on « tourne en rond », donc 2 intégrations par parties s'imposent pour obtenir une équation d'inconnue I_2

$$\begin{cases} u_1(x) = e^{-x} & u_1'(x) = -e^{-x} \\ v_1'(x) = \sin x & v_1(x) = -\cos x \end{cases} \quad I_2 = \left[-e^{-x} \cos x \right]_0^\pi - \int_0^\pi e^{-x} \cos x dx = e^{-\pi} + 1 - \int_0^\pi e^{-x} \cos x dx$$

$$\begin{cases} u_2(x) = e^{-x} & u_2'(x) = -e^{-x} \\ v_2'(x) = \cos x & v_2(x) = \sin x \end{cases} \quad I_2 = e^{-\pi} + 1 - \left[e^{-x} \sin x \right]_0^\pi - I_2, \text{ d'où } I_2 = \frac{1}{2}(e^{-\pi} + 1)$$

$$3. \begin{cases} u(x) = x & u'(x) = 1 \\ v'(x) = \frac{1}{\cos^2 x} & v(x) = \tan x \end{cases} \quad I_3 = \left[x \tan x \right]_0^{\frac{\pi}{4}} - \int_0^{\frac{\pi}{4}} \tan x dx = \frac{\pi}{4} + \left[\ln |\cos x| \right]_0^{\frac{\pi}{4}} = \frac{\pi}{4} + \ln \frac{1}{\sqrt{2}} = \frac{\pi}{4} - \frac{1}{2} \ln 2$$

$$4. I_4 = \int_1^3 u'(x) \times (u(x))^3 dx \text{ avec } u(x) = \ln x + x, \text{ donc } I_4 = \left[\frac{(\ln x + x)^4}{4} \right]_1^3 = \frac{(3 + \ln 3)^4 - 1}{4}$$

$$5. \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = \frac{u'(x)}{u(x)} \text{ avec } u(x) = 1 + e^x, \text{ donc}$$

$$I_5 = \left[\ln(1 + e^x) \right]_{-1}^1 = \ln \frac{1+e}{1+\frac{1}{e}} = \ln e \left(\frac{1+e}{e+1} \right) = 1$$

6. Faites un dessin et vous verrez qu'il s'agit de l'aire du demi-disque unité supérieur, donc $I_6 = \frac{\pi}{2}$.

Solution du test 4.5

$$1. \cos x = \frac{e^{ix} + e^{-ix}}{2}$$

$$2. \cos^5 x = \frac{1}{16} \left(\frac{e^{5ix} + 5e^{3ix} + 10e^{ix} + 10e^{-ix} + 5e^{-3ix} + e^{-5ix}}{2} \right) = \frac{1}{16} (\cos(5x) + 5\cos(3x) + 10\cos x)$$

$$3. \int_{\frac{\pi}{6}}^{\frac{\pi}{4}} \cos^5 t dt = \frac{1}{16} \int_{\frac{\pi}{6}}^{\frac{\pi}{4}} (\cos(5t) + 5\cos(3t) + 10\cos t) dt = \left[\frac{\sin(5t)}{5} + \frac{5\sin(3t)}{3} + 10\sin t \right]_{\frac{\pi}{6}}^{\frac{\pi}{4}}$$

$$\int_{\frac{\pi}{6}}^{\frac{\pi}{4}} \cos^5 t dt = \frac{43\sqrt{2}}{120} - \frac{203}{480}$$

Solution du test 4.6

Soit σ_n la subdivision de $[0, 1]$ en n sous-intervalles égaux de largeur $1/n$. Pour tout $i \in \llbracket 0, n \rrbracket$ on a $a_i = \frac{i}{n}$, le minimum m_i du i^{e} intervalle $[a_{i-1}, a_i]$ valant a_{i-1} et le maximum a_i puisque la fonction est strictement croissante sur $[0, 1]$.

Alors :

$$\begin{aligned}
 s(\sigma_n) &= \sum_{i=i}^n (a_i - a_{i-1}) a_{i-1} \\
 &= \sum_{i=i}^n \frac{1}{n} \cdot \frac{i-1}{n} \\
 &= \frac{1}{n^2} \sum_{i=i}^n i-1 \\
 &= \frac{1}{n^2} \sum_{i=0}^{n-1} i \\
 &= \frac{1}{n^2} \cdot \frac{n(n-1)}{2} \\
 &= \frac{1}{2} \left(1 - \frac{1}{n} \right)
 \end{aligned}$$

De même pour la somme supérieure :

$$\begin{aligned}
 S(\sigma_n) &= \sum_{i=i}^n (a_i - a_{i-1}) a_i \\
 &= \sum_{i=i}^n \frac{1}{n} \cdot \frac{i}{n} \\
 &= \frac{1}{n^2} \sum_{i=i}^n i \\
 &= \frac{1}{n^2} \cdot \frac{n(n+1)}{2} \\
 &= \frac{1}{2} \left(1 + \frac{1}{n} \right)
 \end{aligned}$$

Ainsi $(S(\sigma_n) - s(\sigma_n)) = \frac{1}{n}$ donc $\lim_{n \rightarrow +\infty} (S(\sigma_n) - s(\sigma_n)) = 0$.

La fonction f est bien (DARBOUX-)intégrable sur $[0, 1]$ et

$$\int_0^1 x dx = \lim_{n \rightarrow +\infty} s(\sigma_n) = \lim_{n \rightarrow +\infty} S(\sigma_n) = \frac{1}{2}$$

Ouf ! On retrouve la valeur bien connue.

Solution du test 4.7

1. Comme $0 \leq \sin x \leq 1$ et $x+n \geq n$ sur $[0, \pi]$, on obtient :

$$0 \leq \int_0^\pi \frac{\sin x}{x+n} dx \leq \int_0^\pi \frac{1}{n} dx = \frac{\pi}{n} \xrightarrow{n \rightarrow +\infty} 0$$

2. En utilisant la tactique du coup de pouce et sachant que $1+x^2 \geq 1$ sur $[0, 1]$, on obtient :

$$0 \leq \int_0^1 \frac{x^n}{1+x^2} dx \leq \int_0^1 x^n dx = \frac{1}{n+1} \xrightarrow{n \rightarrow +\infty} 0$$

3. Une majoration brutale en utilisant les bornes d'intégration n'amène rien. Alors on ruse : x^{2n} « ressemble » à $x^{2n} - 1 = (x^n - 1)(x^n + 1)$ et on va pouvoir « éliminer » le dénominateur. Il y a beaucoup de guillemets car on marche à l'intuition, mais l'intuition ne descend pas du ciel : elle vient d'une bonne compréhension des phénomènes. Il faut apprendre à localiser ce qui cloche...

$$\int_0^2 \frac{x^{2n}}{1+x^n} dx \geq \int_0^2 \frac{x^{2n}-1}{x^n+1} dx = \int_0^2 (x^n-1) dx = \frac{2^{n+1}}{n+1} - 2 \xrightarrow{n \rightarrow +\infty} +\infty$$

4. La fonction $x \mapsto \sqrt{1+x^n}$ est croissante, donc on majore

$$0 \leq \int_1^{1+1/n} \sqrt{1+x^n} dx \leq \int_0^{1+1/n} \sqrt{1+(1+1/n)^n} dx \leq (1+1/n-1)\sqrt{1+(1+1/n)^n}$$

Or $\lim_{n \rightarrow +\infty} (1+1/n)^n = e$ donc $\lim_{n \rightarrow +\infty} \frac{1}{n} \sqrt{1+(1+1/n)^n} = 0$ d'où $\lim_{n \rightarrow +\infty} \int_1^{1+1/n} \sqrt{1+t^n} dt = 0$ en appliquant la règle de l'encadrement.

5. En pensant au coup de pouce, on obtient :

$$0 \leq \left| \int_0^\pi \left(\frac{n \sin x}{x+n} - \sin x \right) dx \right| = \left| \int_0^\pi \frac{x \sin x}{x+n} dx \right| \leq \int_0^\pi \frac{\pi}{n} dx = \frac{\pi^2}{n} \rightarrow 0$$

sachant que $\left| \frac{1}{x+n} \right| \leq \frac{1}{n}$, $\sin(x) \leq 1$ et $|x| \leq \pi$.

$$\text{D'où } \lim_{n \rightarrow +\infty} \int_0^\pi \frac{n \sin x}{x+n} dx = 2.$$

Solution du test 4.8

Ce calcul intervient notamment en électronique. Si p vaut 0, $I(t) = \frac{t^2}{2}$ et $\lim_{t \rightarrow +\infty} I(t) = +\infty$.

Sinon, une intégration par parties règle l'affaire.

$$\begin{cases} u(x) = x & u'(x) = 1 \\ v'(x) = e^{-px} & v(x) = \frac{-e^{-px}}{p} \end{cases} \quad I(t) = \left[\frac{-xe^{-px}}{p} \right]_0^t + \frac{1}{p} \int_0^t e^{-px} dx = \frac{-te^{-pt}}{p} - \frac{1}{p^2} (e^{-pt} - 1)$$

Si $p < 0$, $\lim_{t \rightarrow +\infty} I(t) = +\infty$, et si $p > 0$, $\lim_{t \rightarrow +\infty} I(t) = \frac{1}{p^2}$.

Solution du test 4.9

On linéarise l'expression à intégrer :

$$\begin{aligned} X(s) &= 2 \int_{-1/2}^{1/2} \frac{1}{2} (\cos((2f+1)\pi t) + \cos((2f-1)\pi t)) dt \\ &= \left[\frac{1}{(2f+1)\pi} \sin(2f+1)\pi t + \frac{1}{(2f-1)\pi} \sin(2f-1)\pi t \right]_{-1/2}^{1/2} \\ &= 2 \left(\frac{1}{(2f+1)\pi} \sin(2f+1)\frac{\pi}{2} + \frac{1}{(2f-1)\pi} \sin(2f-1)\frac{\pi}{2} \right) \end{aligned}$$

Solution du test 4.10

La dérivée de $x \mapsto \int_0^x f(t) dt$ est $f(x)$. On obtient donc

$$g'(x) = e^{-x} \left(f(x) - \int_0^x f(t) dt \right)$$

qui est négatif d'après l'énoncé. Donc g est décroissante sur \mathbb{R}^+ , or elle est positive d'après les hypothèses et $g(0) = 0$.

Donc g est identiquement nulle. On en déduit que $\int_0^x f(t) dt = 0$. Or f est elle-même positive, donc est aussi identiquement nulle.

Solution du test 4.11

- La fonction f est définie sur \mathbb{R}^* car pour tout x appartenant à \mathbb{R}^* , le segment d'extrémités x et $3x$ ne contient pas 0.
- De plus

$$f(-x) = \int_{-x}^{-3x} \frac{\cos t}{t} dt = \int_x^{3x} \frac{\cos u}{-u} (-du) = f(x)$$

en posant $u = -t$ donc f est paire.

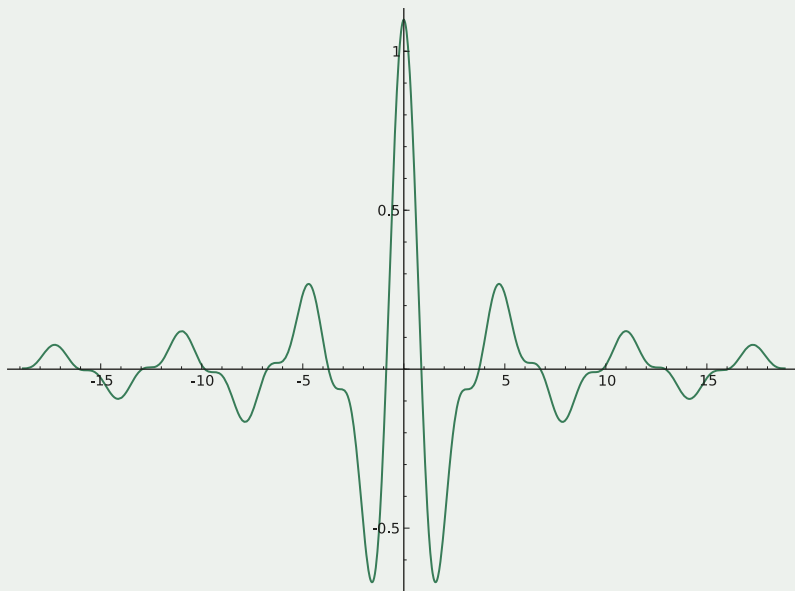
– Étudions-la sur $]0, +\infty[$, intervalle sur lequel f est dérivable.

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

en utilisant la formule $\cos p - \cos q = -2 \sin \frac{p+q}{2} \sin \frac{p-q}{2}$.

Nous en déduisons que f' est du signe contraire de $\cos x$ sur $]0, +\infty[$.

```
In [111]: t = sy.symbols('t')
In [112]: x = sy.Symbol('x', positive='True')
In [113]: f = lambda x : sy.integrate(sy.cos(t) / t, (t, x, 3*x))
In [114]: sy.limit(f(x), x, 0)
Out [114]: log(3)
In [115]: sy.limit(f(x), x, sy.oo)
Out [115]: 0
In [116]: sy.plot(f(t), (t, -6*sy.pi, 6*sy.pi))
```



Solution du test 4.12

1. $\sum_{k=0}^n t^k = \frac{1-t^{n+1}}{1-t}$ puisque $t < 1$, donc

$$\left| \frac{1}{1-t} - \sum_{k=0}^n t^k \right| = \left| \frac{t^{n+1}}{1-t} \right| \leq \frac{t^{n+1}}{1-x}$$

car $0 \leq t \leq x$.

2. Sur $[-1, 0]$, $|1-t| \geq 1$, donc on obtient cette fois-ci

$$\left| \frac{1}{1-t} - \sum_{k=0}^n t^k \right| = \left| \frac{t^{n+1}}{1-t} \right| \leq |t|^{n+1}$$

3. Découpons à nouveau notre problème en 2.

– Sur $[0, 1]$, on intègre l'inégalité obtenue :

$$\left| \int_0^x \left(\frac{1}{1-t} - \sum_{k=0}^n t^k \right) dt \right| \leq \int_0^x \left| \frac{1}{1-t} - \sum_{k=0}^n t^k \right| dt \leq \int_0^x \frac{t^{n+1}}{1-x} dt = \frac{1}{n+2} \frac{x^{n+2}}{1-x} \xrightarrow{n \rightarrow +\infty} 0$$

or

$$\int_0^x \left(\frac{1}{1-t} - \sum_{k=0}^n t^k \right) dt = \int_0^x \frac{1}{1-t} dt - \sum_{k=0}^n \int_0^x t^k dt = -\ln(1-x) - \sum_{k=0}^n \frac{x^{k+1}}{k+1}$$

donc nous en déduisons le résultat demandé.

– Sur $[-1, 0]$, seul le majorant change, mais on a cette fois-ci

$$\int_{-1}^0 (-t)^{n+1} dt = \frac{1}{n+2} \xrightarrow{n \rightarrow +\infty} 0$$

et nous obtenons le même résultat.

« We cannot predict whether a given photon will arrive at A or B. All we can predict is that out of 100 photons that come down, an average of 4 will be reflected by the front surface. Does this mean that physics, a science of great exactitude, has been reduced to calculating only the probability of an event, and not predicting exactly what will happen? Yes. That's a retreat, but that's the way it is: Nature permits us to calculate only probabilities. Yet science has not collapsed. »

Richard P. FEYNMAN – *The strange theory of light and matter* (1985)

Les probabilités sont omniprésentes dans le monde scientifique, c'est pourquoi elles ont fait leur apparition... tardive en CPGE à la rentrée 2013. Elles tiennent un rôle très important au lycée également mais, malheureusement, des notions compliquées sont souvent introduites sans preuve ni rigueur. Vous allez donc reprendre les choses au début en prépa tout en vous restreignant aux probabilités discrètes. Nous allons en faire de même dans ce chapitre : vous pouvez vous contenter d'un survol en première lecture mais il sera utile de revenir sur ce cours avec plus d'attention par la suite pour passer d'une vision lycéenne à un travail de préparationnaire... Le travail sur le cours est une nouvelle fois fondamental : chaque définition doit être comprise, chaque démonstration travaillée.

5.1 Rappels de théorie des ensembles

Ensembles finis

Il est aisé de comprendre intuitivement ce qu'est un ensemble fini. En voici une définition :

Définition : ensemble fini

On dit qu'un ensemble E est fini si, et seulement si, il existe un entier naturel n et une bijection φ de $\llbracket 1, n \rrbracket$ sur E .

Cela signifie en fait que chaque élément d'un ensemble fini E porte un numéro de dossier et qu'on connaît le nombre total de dossiers : n . Ce nombre jouant un rôle important, on lui donne un nom :

Définition : cardinal d'un ensemble fini

L'entier défini précédemment est appelé cardinal de E et noté $\text{Card}(E)$ ou $|E|$.

Si on considère l'ensemble formé des pages de cet ouvrage, il constitue un ensemble fini : quel est son cardinal ?

Par convention, l'ensemble vide est de cardinal nul : $|\emptyset| = 0$.

Parties d'un ensemble

Définition : parties d'un ensemble

Soit E un ensemble. L'ensemble formé de tous les sous-ensembles de E est appelé **ensemble des parties** de E et noté $\mathfrak{P}(E)$.

Il faudra donc faire bien attention au moment d'utiliser les symboles \in et \subseteq . Ainsi :

$$A \in \mathfrak{P}(E) \rightarrow A \subseteq E$$

Si $E_2 = \{a, b\}$ alors $\mathfrak{P}(E_2) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$.

Nous aurons également besoin de parler des éléments de E qui n'appartiennent pas à une de ses parties.

Définition : complémentaire d'une partie

Soit A une partie de E . On appelle **complémentaire** de A dans E l'ensemble des éléments de E qui n'appartiennent pas à A . On note cet ensemble \bar{A}_E , $\complement_E A$ ou plus simplement \bar{A} quand il n'y a pas d'ambiguïté.

Par exemple, le complémentaire de l'ensemble des filles dans une classe est l'ensemble des garçons de cette classe. La notation permet de faire le lien avec la négation d'une proposition : la négation de « $x \in A$ » est « $x \in \bar{A}$ ».

5.2 Une dose d'algèbre générale

Relation d'ordre

Soit A , B et C trois éléments de $\mathfrak{P}(E)$.

Considérons la relation d'inclusion (\subseteq) qui relie deux éléments quelconques de $\mathfrak{P}(E)$.

On obtient facilement que :

- $A \subseteq A$: on dit que la relation \subseteq est *réflexive* ;
- $A \subseteq B \wedge B \subseteq A \rightarrow A = B$: on dit que la relation \subseteq est *antisymétrique* ;
- $A \subseteq B \wedge B \subseteq C \rightarrow A \subseteq C$: on dit que la relation \subseteq est *transitive*.

Théorème : la relation d'ordre \subseteq

Ces trois propriétés étant vérifiées, on dit que \subseteq est une **relation d'ordre** sur $\mathfrak{P}(E)$.

Vous vérifierez que la relation \leq sur \mathbb{R} a exactement les mêmes propriétés : vous commencerez à entrevoir l'essence de l'algèbre qui est de dégager et d'étudier des structures les plus générales possibles afin de simplifier l'étude de phénomènes qui paraissent différents.

Loi de composition interne

Définition : réunion d'ensembles

La réunion de deux ensembles A et B notée $A \cup B$ est l'ensemble des éléments appartenant à A **OU** B (le « OU » correspond au « ou » inclusif).

Quelques remarques :

- comme $A \cup B$ est encore un élément de $\mathfrak{P}(E)$, on dit que \cup est une **loi de composition interne** ;

- comme $A \cup (B \cap C) = (A \cup B) \cap C$, on dit que la loi \cup est **associative** ;
- comme $A \cup B = B \cup A$, on dit que la loi \cup est **commutative** ;
- comme $A \cup \emptyset = A$, on dit que \emptyset est un **élément neutre** pour la loi \cup ;
- comme $A \cup A = A$, on dit que la loi \cup est **idempotente**.

On peut remarquer que ces propriétés sont vérifiées par l'addition sur \mathbb{N} , l'élément neutre étant 0, sauf une : laquelle ?

Définition : intersection d'ensembles

L'intersection de deux ensembles A et B notée $A \cap B$ est l'ensemble des éléments appartenant à A ET B.

Vous vérifierez que la loi \cap a les mêmes propriétés que \cup avec cette fois comme élément neutre E.

Lien avec les opérateurs logiques

Comme souvent en mathématiques, on retrouve des structures semblables dans des domaines différents.

Nous avons déjà découvert les liens entre \cup et OU puis entre \cap et ET.

On peut également faire le rapprochement entre l'implication des propositions et l'inclusion des ensembles.

Ainsi, si deux ensembles A et B vérifient $A \subseteq B$, alors il est impossible de trouver un élément x tel que $x \in A$ et $x \notin B$. Tous les autres cas sont possibles.

De même, si $A \rightarrow B$, il est impossible que A soit vraie et B soit fausse. Toutes les autres propositions sont vraies.

On fait le même rapprochement entre l'équivalence des propositions et l'égalité des ensembles. Merveilleuses mathématiques...

Partition

On a souvent besoin de découper un ensemble en tranches (souvenez-vous de votre cours de Terminale sur les probabilités totales...). Voici une définition utile :

Définition : partition d'un ensemble

*La famille $(A_i)_{i \in [1, n]}$ de parties non vides d'un ensemble E réalise une **partition** de cet ensemble si, et seulement si :*

$$\forall i \neq j \quad A_i \cap A_j = \emptyset \quad \text{et} \quad \bigcup_{i=1}^n A_i = E$$

Par exemple, une partie A de E et son complémentaire dans E réalisent une partition de E car $A \cap \bar{A} = \emptyset$ et $A \cup \bar{A} = E$.

Produit cartésien

Définition : produit cartésien

Soit $(A_i)_{i \in [1, n]}$ des parties d'un ensemble E. Le produit $\prod_{i=1}^n A_i = A_1 \times A_2 \times \dots \times A_n$ est l'ensemble des n-listes (x_1, x_2, \dots, x_n) telles que $x_1 \in A_1, \dots, x_n \in A_n$.

Soit $A = \{1, 3, 5, 7, 9\}$ et $B = \{0, 2, 4, 6, 8\}$. Alors $(1, 0)$ est un élément de $A \times B$ et $(0, 1)$ est un élément de $B \times A$. L'ordre des ensembles dans le produit et donc des éléments dans les n -listes est important !

5.3 Quelques résultats sur les cardinaux

— Cardinal de $\mathfrak{P}(E)$

Soit $E_2 = \{a, b\}$ et $E_3 = \{a, b, c\}$. En fait, $E_3 = E_2 \cup \{c\}$. Les parties de E_3 sont donc les parties de E_2 et ces mêmes parties auxquelles on adjoint c .

En effet, $\mathfrak{P}(E_3) = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \emptyset \cup \{c\}, \{a\} \cup \{c\}, \{b\} \cup \{c\}, \{a, b\} \cup \{c\}\}$.

Ainsi $|\mathfrak{P}(E_3)| = 2 \times |\mathfrak{P}(E_2)|$.

Notons de manière générale E_n un ensemble à n éléments avec $n \in \mathbb{N}$ et $E_{n+1} = E_n \cup \{\alpha\}$ avec $\alpha \notin E_n$. On montre facilement que $|\mathfrak{P}(E_{n+1})| = 2 \times |\mathfrak{P}(E_n)|$. Or $\mathfrak{P}(E_0) = \{\emptyset\}$ donc $|\mathfrak{P}(E_0)| = 1$.

Une rapide récurrence permet donc de démontrer que :

Théorème : cardinal de $\mathfrak{P}(E)$

Si $|E| = n$ alors $|\mathfrak{P}(E)| = 2^n$.

— Cardinal d'une partition

Il est assez immédiat d'obtenir le résultat suivant :

Théorème : cardinal d'une partition

Si la famille $(A_i)_{i \in \llbracket 1, n \rrbracket}$ de parties non vides d'un ensemble E réalise une **partition** de cet ensemble alors :

$$|E| = \sum_{i=1}^n |A_i|$$

Pour compter les élèves du lycée, il suffit de compter les élèves de chaque classe et d'additionner les résultats...

— Cardinaux et inclusion

Théorème

$$\forall (A, B) \in \mathfrak{P}(E) \times \mathfrak{P}(E), A \subseteq B \rightarrow |A| \leq |B|.$$

Pour le démontrer, il suffit d'utiliser une petite ruse (bientôt) habituelle : $A \cap B$ et $\overline{A} \cap B$ donc A et $\overline{A} \cap B$ forment une partition de B (car $A \subseteq B$) donc, d'après le théorème du cardinal d'une partition, ci-dessus, $|A| + |\overline{A} \cap B| = |B|$, donc $|A| \leq |B|$.

Cette application n'est pas une équivalence ! Prenez $A = \{1\}$ et $B = \{3; 4\}$.

— Cardinal d'un produit cartésien

Voici un autre résultat qui sera bien utile en probabilités :

Théorème : cardinal d'un produit cartésien

Soit $(A_i)_{i \in \llbracket 1, n \rrbracket}$ des parties d'un ensemble E .

$$\left| \prod_{i=1}^n A_i \right| = \prod_{i=1}^n |A_i|$$

En effet, pour chaque n -liste (x_1, \dots, x_n) , il y a $|A_1|$ choix pour la première composante, $|A_2|$ choix pour la deuxième, etc.

— Formule du crible de POINCARÉ

Nous n'en verrons qu'un cas particulier que vous devez déjà connaître :

Théorème : formule du crible pour deux ensembles

$$\forall (A, B) \in \mathfrak{P}(E) \times \mathfrak{P}(E), |A \cup B| = |A| + |B| - |A \cap B|.$$

Pour le démontrer, nous allons former une partition de $A \cup B$ avec $A \cap \bar{B}$, $\bar{A} \cap B$ et $A \cap B$. Mais on peut également « partitionner » A avec $A \cap \bar{B}$ et $A \cap B$ puis « partitionner » B avec $B \cap \bar{A}$ et $B \cap A$.

On obtient donc que $|A| = |A \cap \bar{B}| + |A \cap B|$ puis que $|B| = |B \cap \bar{A}| + |B \cap A|$.

On en déduit que $|A| + |B| = |A \cap \bar{B}| + |B \cap \bar{A}| + |A \cap B| + |B \cap A|$ et enfin que :

$$|A| + |B| - |A \cap B| = |A \cap \bar{B}| + |B \cap \bar{A}| + |A \cap B| = |A \cup B|$$

5.4 Dénombrement

— Nombre de permutations

Définition : permutation

Soit E un ensemble. Une permutation de E est une application bijective de E sur E . On note $\mathfrak{S}(E)$ l'ensemble des permutations de E .

Généralement, on note les permutations sous forme d'une matrice où la première ligne correspond aux éléments de E et où la deuxième ligne correspond aux images des éléments de E .

Par exemple, les permutations de $\{a, b, c\}$ sont :

$$\begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix} \begin{pmatrix} a & b & c \\ a & c & b \end{pmatrix} \begin{pmatrix} a & b & c \\ b & a & c \end{pmatrix} \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix} \begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix} \begin{pmatrix} a & b & c \\ c & b & a \end{pmatrix}$$

On peut même se contenter d'écrire (a, b, c) , (a, c, b) , (b, a, c) , (b, c, a) , (c, a, b) , (c, b, a) si l'on est sûr que la première ligne est toujours (a, b, c) .

Théorème : nombre de permutations

Il y a $n!$ permutations d'un ensemble de n éléments.

La démonstration s'effectue par récurrence. Il y a bien sûr 1! façon de permuter 1 élément. On peut donc supposer qu'il existe au moins un entier k tel qu'il y ait $k!$ permutations d'un ensemble à k éléments.

Considérons un ensemble à $k + 1$ éléments. On en choisit 1 : il y a $k + 1$ choix possibles. Ensuite, il s'agit d'ordonner les k éléments restant ce qui laisse $k!$ possibilités.

Il y a donc finalement $(k + 1) \times k! = (k + 1)!$ permutations au total.

On retrouve dans l'exemple précédent qu'il y a $3! = 6$ permutations de $\{a, b, c\}$.

On conviendra que $0! = 1$.

Nombre d'applications injectives – arrangements sans répétitions

Une source veut envoyer à chacun de ses cinq clients un fichier différent parmi les vingt-six situés dans un certain répertoire.

Combien de possibilités a la source pour envoyer un fichier à chacun des 26 clients ?

On peut modéliser cette situation : il s'agit de compter les applications injectives (i.e. telles que chaque élément de l'ensemble de départ admet au plus une image dans l'ensemble d'arrivée) de l'ensemble E des clients vers l'ensemble F des fichiers. Il ne peut y avoir répétition donc il y a 26 choix de fichiers pour le premier client, 25 pour le deuxième, etc., ce qui donne $26 \times 25 \times 24 \times 22 \times 21$ possibilités.

On peut généraliser.

Définition : arrangement sans répétition

Une application injective d'un ensemble E de cardinal p dans un ensemble F de cardinal n (avec $p \leq n$) est appelée un arrangement sans répétition de p parmi n .

Un exemple fameux est la recherche de tiercés dans l'ordre (sans ex-æquo...) : pourquoi ?

Nous pouvons formuler cette propriété plus synthétiquement. Notons A_n^p le nombre d'arrangements de p parmi n . Alors :

$$\begin{aligned} A_n^p &= n(n-1)(n-2) \cdots (n-p+1) \times \frac{(n-p)(n-p-1)(n-p-2) \times \cdots \times 2 \times 1}{(n-p)(n-p-1)(n-p-2) \times \cdots \times 2 \times 1} \\ &= \frac{n!}{(n-p)!}. \text{ D'où :} \end{aligned}$$

Théorème : nombre d'arrangements

Le nombre d'arrangements sans répétition de p parmi n est :

$$A_n^p = n(n-1)(n-2) \cdots (n-p+1) = \frac{n!}{(n-p)!}$$

— Combinaisons sans répétition

Il n'y a pas assez de gagnants au loto. Les règles en sont donc modifiées. Il s'agit maintenant de trouver 3 numéros parmi 5. Combien y a-t-il de grilles (combinaisons) possibles ?

Par exemple, on pourrait dire que j'ai cinq manières de choisir le premier numéro, quatre choix pour le deuxième et trois choix pour le troisième, donc il y a $5 \times 4 \times 3$ grilles différentes mais, dans ce cas, je compte des 3-listes ordonnées alors que les 3-listes $(1,2,3)$ et $(3,2,1)$ correspondent à la même grille ou combinaison $\{1,2,3\}$.

Posons une petite définition pour clarifier les débats. Donnons en fait un nom à une grille du loto, c'est-à-dire à un sous-ensemble (une partie) contenant p éléments d'un plus grand ensemble contenant n éléments.

Définition : combinaison sans répétition

*Soit n et p deux entiers naturels et E un ensemble contenant éléments. Un sous-ensemble de E contenant p éléments est appelé une **combinaison** de p éléments de E ou encore une **p -combinaison** d'éléments de E .*

Or ce qui nous intéresse, c'est le nombre de ces combinaisons, donc introduisons une notation :

Définition : nombre de combinaisons sans répétitions

Le nombre de p -combinaisons d'un ensemble E contenant n éléments est noté $\binom{|E|}{p}$.

Revenons à notre mini-loto. Considérons une grille quelconque (i.e. une 3-combinaison de l'ensemble des 5 numéros) : par exemple $\{2,4,5\}$. Nous avons vu dans un paragraphe précédent qu'il y a $3!$ façons d'ordonner ces nombres. Finalement, il y a $\binom{5}{3} \times 3!$ suites de 3 nombres ordonnées : c'est le nombre d'arrangements de p parmi n . Or nous en avons comptées $5 \times 4 \times 3$ tout à l'heure. Nous en déduisons finalement que :

$$\binom{5}{3} = \frac{A_5^3}{3!} = \frac{5 \times 4 \times 3}{3!}$$

Il est alors possible de généraliser la formule suivante :

Théorème : nombre de combinaisons

$$\binom{n}{p} = \frac{A_n^p}{p!} = \frac{n(n-1)(n-2)\cdots(n-(p-1))}{p!} = \frac{n!}{p!(n-p)!}$$

5.5 Triangle de PASCAL – Binôme de NEWTON

À l'aide des formules précédentes, on démontre facilement (faites-le !) le résultat suivant :

Théorème

$$\binom{n}{p} = \binom{n}{n-p}$$

On établit ensuite, toujours par le calcul, la relation suivante, dite **relation de PASCAL** même si les mathématiciens chinois l'avaient mise en évidence avant lui :

Théorème : relation de PASCAL

$$\binom{n}{p} = \binom{n-1}{p} + \binom{n-1}{p-1}$$

Il est possible de démontrer cette formule à l'aide d'un raisonnement ensembliste : pour former des groupes de p éléments dans un ensemble en contenant n , on distingue un élément n_0 quelconque :

- soit le groupe le contient et alors il y a $\binom{n-1}{p-1}$ choix des $p-1$ éléments distincts de n_0 parmi les $n-1$ restant ;

- soit le groupe ne le contient pas et il faut donc choisir p éléments parmi les $n - 1$ éléments distincts de n_0 : il y a donc cette fois $\binom{n-1}{p}$ choix.

Cette relation permet de démontrer la formule du binôme de NEWTON (que nous admettrons) :

Théorème : formule du binôme

$$\forall (a, b) \in \mathbb{R}^2, \quad (a + b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}$$

5.6 Probabilités ?

Le calcul des probabilités est très important dans toute science. Il nous permet de résoudre des problèmes tout en étant confrontés à l'incertitude ou à des systèmes très complexes.

Contrairement aux clichés, la théorie des probabilités, en tant que discipline mathématique, est dotée de toute la rigueur et de la précision possibles. Elle puise ses sources dans tous les champs des mathématiques. Il n'est dès lors pas étonnant de constater que les prix mathématiques les plus prestigieux sont attribués depuis quelque temps à des travaux liés aux probabilités.

Il ne faut pas confondre les probabilités avec la statistique qui complète un modèle inconnu à l'aide d'observations (raisonnement inductif) et fait plutôt appel à l'intuition.

Enfin, il est tout à fait possible de calculer des probabilités sans faire de statistique.

5.7 Avant la formalisation

Essayons, grâce à notre petite expérience probabiliste acquise au lycée, de dégager les grandes lignes de ce que pourrait être une théorie formalisée des probabilités.

Lors de l'étude des probabilités, on est amené à effectuer une **expérience aléatoire**, c'est-à-dire un processus dont on ne peut prévoir à l'avance le résultat, qui peut donner des résultats différents même si l'on répète l'expérience dans des conditions qui semblent identiques.

Nous allons considérer par la suite le lancement d'un dé cubique équilibré et tout et tout...

On doit d'abord connaître l'ensemble des résultats possibles : c'est l'**univers** que nous noterons par la suite Ω .

Par exemple, si je lance une fois le dé, $\Omega_1 = \{1, 2, 3, 4, 5, 6\}$.

Si je lance deux fois de suite le dé et que je note dans l'ordre le numéro des faces, j'obtiens dans ce cas $\Omega_2 = \{11, 12, 13, \dots, 66\}$

Si je lance les dés jusqu'à obtenir deux 6 à la suite, le nombre de lancers n'est maintenant pas fixé à l'avance et l'univers est *a priori* infini mais en demeurant discret : chaque événement peut en effet être « numéroté ». Plus rigoureusement, Ω_3 peut être mis en bijection avec \mathbb{N} : $\Omega_3 = \{1366, 3546213466, 55236221466, \dots\}$.

Il faut maintenant étudier la notion d'« événement »¹. Disons dans un premier temps qu'il s'agit d'une assertion vérifiable relativement au résultat d'une expérience aléatoire. Il est

1. Depuis 1990, l'Académie française préconise de mettre un accent grave au deuxième « e » d'événement.

donc « associé » à un sous-ensemble (une partie) de Ω . Un élément de Ω sera associé à un **évènement élémentaire**.

Reprenons le deuxième exemple. Voici quelques évènements :

- « Obtenir deux fois 3 » : $E_a = \{33\}$. C'est un évènement élémentaire.
- « Ne pas obtenir deux fois 3 » : $E_b = \Omega_2 \setminus E_a \subseteq \Omega_2$.
- « Obtenir deux faces dont la somme des numéros fait 11 » : $E_c = \{56, 65\} \subseteq \Omega_2$.
- « Obtenir deux faces dont le numéro est inférieur à 7 » : $E_d = \Omega_2$. Il s'agit donc de l'**évènement certain**.
- « Obtenir deux faces dont la somme des numéros fait 37 » : $E_e = \emptyset$. Il s'agit cette fois de l'**évènement impossible**.

On comprend que l'univers lui-même et l'ensemble vide sont des évènements. On voudrait que la réunion et l'intersection d'évènements soit encore un évènement et que le complémentaire de la partie associée à un évènement définisse encore un évènement : les mathématiciens (français...) disent alors que l'ensemble de tous ces évènements forment une **tribu**. Dans le reste du monde, on parle plutôt de σ -algèbre de BOOLE : les ensembles concernés vérifient en effet certaines propriétés *algébriques*.

Il reste enfin à définir un moyen de **mesurer** les probabilités : c'est une sorte de degré de certitude correspondant à une échelle graduée de 0 à 1, 0 correspondant au degré de l'évènement impossible et 1 au degré de l'évènement certain. Il est important de comprendre que ce degré de certitude est donné *a priori* et permet donc de prévoir l'issue de l'expérience avant de la réaliser.

Il faudrait que cette mesure assure une cohérence entre les évènements et les degrés.

Par exemple, si $A \subseteq B$, on aimerait que $\mathbb{P}(A) \leq \mathbb{P}(B)$. On aimerait également que l'union de deux parties de l'univers se « transforme » en somme de leur degré de certitude, mais on « sent » bien que cela ne pourra être vrai que si les évènements sont totalement **incompatibles** : $A \cap B = \emptyset \Rightarrow \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$. On retrouve ici cette notion algébrique de « transport » des opérations : la notion d'*isomorphisme*.

Considérons pour illustrer ces dernières notions le premier exemple du lancer unique d'un dé.

Notre univers Ω_1 contient 6 éléments et prenons comme tribu naturellement associée les $2^6 = 64$ parties de Ω_1 qui forment l'ensemble $\mathfrak{P}(\Omega_1)$.

Imposons pour notre modèle que la mesure du degré de certitude ne dépende pas des numéros des faces mais uniquement du cardinal de la partie considérée : chacun des six évènements élémentaires aura donc la même probabilité : $\mathbb{P}(\ll 1 \gg) = \mathbb{P}(\ll 2 \gg) = \dots = \mathbb{P}(\ll 6 \gg) = \frac{1}{6}$.

La probabilité d'obtenir un numéro pair sera donc, sachant que l'évènement est associé à $A = \{2, 4, 6\}$: $\mathbb{P}(A) = \mathbb{P}(\ll 2 \gg) + \mathbb{P}(\ll 4 \gg) + \mathbb{P}(\ll 6 \gg) = \frac{1}{2}$.

Mais si le dé est truqué ? Si l'univers est infini ? S'il n'est pas dénombrable ?

Il est temps de définir clairement et rigoureusement les notions afin de ne pas dire de bêtises...

5.8 Espace probabilisable – Espace probabilisé

Dans ce chapitre, conformément à votre futur programme de prépa, Ω est un ensemble fini. Nous le considérerons parfois dénombrable infini.

Définition : tribu

Une tribu \mathcal{T}_Ω définie sur un univers Ω est une partie de $\mathfrak{P}(\Omega)$ qui vérifie :

1. $\Omega \in \mathcal{T}_\Omega$;
2. $A \in \mathcal{T}_\Omega \rightarrow \complement_\Omega A \in \mathcal{T}_\Omega$;
3. $\forall I \subseteq \mathbb{N}, (A_i)_{i \in I} \in \mathcal{T}_\Omega^{|I|} \rightarrow \bigcup_{i \in I} A_i \in \mathcal{T}_\Omega$.



Une tribu est une partie de $\mathfrak{P}(\Omega)$: c'est donc un ensemble de sous-ensembles de Ω .

Définition : espace probabilisable

Soit \mathcal{T}_Ω une tribu définie sur l'univers Ω . On dit que $(\Omega, \mathcal{T}_\Omega)$ est un espace probabilisable (ou espace d'évènements). Les éléments de la tribu sont appelés **évènements**.



Pour une expérience aléatoire donnée, il peut y avoir de nombreux espaces probabilisables différents.

Reprenons un des exemples introduits dans la section précédente :

- $(\Omega_1, \{\emptyset, \Omega_1\})$ est un espace probabilisable : on ne pourra s'intéresser qu'aux évènements « obtenir un numéro entre 1 et 6 » et « ne pas obtenir de numéro entre 1 et 6 », ce qui est un peu sommaire ;
- $(\Omega_1, \{\emptyset, A, \complement_\Omega A, \Omega_1\})$ avec A associé à « obtenir un 6 » ;
- $(\Omega_1, \mathbb{P}(\Omega_1))$ est ce qu'on peut faire de plus précis.

Vérifiez qu'il s'agit bien d'espaces probabilisables !

Définition : axiomes de KOLMOGOROV

Soit (Ω, \mathcal{T}) un espace probabilisable. On appelle **mesure de probabilité** (ou probabilité tout court) une application \mathbb{P} définie sur \mathcal{T} vérifiant :

1. $\forall A \in \mathcal{T}, \mathbb{P}(A) \in [0, 1]$;
2. $\mathbb{P}(\Omega) = 1$;
3. $\forall (A, B) \in \mathcal{T} \times \mathcal{T}, A \cap B = \emptyset \Rightarrow \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$.

On dit alors que $(\Omega, \mathcal{T}, \mathbb{P})$ est un **espace probabilisé**.

Vous pouvez vérifier que l'espace probabilisable $(\Omega, \mathbb{P}(\Omega))$ avec $\Omega = \{1, 2, 3, 4, 5, 6\}$ muni de l'application :

$$\begin{aligned} \mathfrak{P}(\Omega) &\rightarrow [0, 1] \\ \mathbb{P} : \\ A &\mapsto \frac{|A|}{6} \end{aligned}$$

définit un espace probabilisé.

Pour l'axiome 3, on se souviendra de la formule du crible (cf. théorème de la « formule du crible » page 131).

Comme Ω est fini, il existe une certaine famille I d'entiers telle que $\Omega = \bigcup_{i \in I} \{\omega_i\}$ et pour tout

évènement A de \mathcal{T} , il existe une famille $J \subseteq I$ telle que $A = \bigcup_{j \in J} \{\omega_j\}$.

Alors, par application itérée du troisième axiome :

$$\mathbb{P}(A) = \mathbb{P}\left(\bigcup_{j \in J} \{\omega_j\}\right) = \sum_{j \in J} \mathbb{P}(\{\omega_j\})$$

Ainsi la donnée des valeurs prises par \mathbb{P} sur les événements élémentaires suffit à caractériser \mathbb{P} sur \mathcal{T} .

Dans le cas où Ω est fini et si tous les événements élémentaires ont la même probabilité (on dit qu'il y a **équiprobabilité**), alors :

$$\mathbb{P}(\omega_1) = \mathbb{P}(\omega_2) = \dots = \mathbb{P}(\omega_{|\Omega|}) = \frac{1}{|\Omega|}$$

Dans ce cas, \mathbb{P} est appelée **probabilité uniforme** sur Ω .



Toute probabilité n'est pas uniforme... Votre portable sonne : quelle est la probabilité que cela soit Monica B. qui vous fasse une déclaration d'amour ? La tribu contient deux événements en plus de l'événement certain et de l'événement impossible : « c'est Monica » et « ce n'est pas Monica ». Le rapport entre cas favorables et cas possibles est bien $\frac{1}{2}$ et pourtant...

Vous noterez enfin que, lorsque l'univers est infini dénombrable, il ne peut pas y avoir équiprobabilité : on aurait $\sum_{n \geq 1} p = p \sum_{n \geq 1} 1 \neq 1 \dots$

Théorème : principales propriétés

Soit $(\Omega, \mathcal{T}, \mathbb{P})$ un espace probabilisé. La probabilité \mathbb{P} vérifie :

1. $\mathbb{P}(\complement_{\Omega} A) = 1 - \mathbb{P}(A)$;
2. $\mathbb{P}(\emptyset) = 0$;
3. $\forall (A, B) \in \mathcal{T} \times \mathcal{T}, A \subset B \rightarrow \mathbb{P}(A) \leq \mathbb{P}(B)$;
4. $\forall (A, B) \in \mathcal{T} \times \mathcal{T}, \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$ (cas particulier de la formule du crible de POINCARÉ).

Dans la suite, quand il n'y aura pas ambiguïté, on notera $\complement_{\Omega} A$ par \bar{A} .

Les démonstrations utilisent les axiomes de KOLMOGOROV.

Définition : système complet d'événements

Une famille A_1, A_2, \dots, A_n d'événements non vides de \mathcal{T} forme un système complet d'événements si, et seulement si :

1. $\forall i \neq j, A_i \cap A_j = \emptyset$ (événements incompatibles deux à deux) ;
2. $\bigcup_{i=1}^n A_i = \Omega$.

On dit aussi que la famille (A_i) forme une **partition** de Ω .

Théorème : théorème des probabilités totales (version 1)

Soit $(A_i)_{1 \leq i \leq n}$ un système complet d'événements, alors :

$$\forall E \in \mathcal{T}, \mathbb{P}(E) = \sum_{i=1}^n \mathbb{P}(E \cap A_i)$$

La preuve s'obtient en utilisant les axiomes de KOLMOGOROV.

5.9 Probabilités conditionnelles

Dans toute cette section on travaille dans l'espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$.

— Un exemple pour comprendre

Considérons l'expérience simplissime consistant à lancer deux fois un dé à six faces. L'univers Ω est donc constitué de l'ensemble des couples (i, j) , avec i et j appartenant à l'ensemble $\llbracket 1, 6 \rrbracket$: il y a donc 36 éléments dans Ω . Intéressons-nous à la somme des deux chiffres et soit A l'événement « le total fait neuf ». On prendra $\mathcal{F} = \mathbb{P}(\Omega)$.

$$A = \{(3,6), (4,5), (5,4), (6,3)\}$$

donc :

$$\mathbb{P}(A) = \frac{4}{36} = \frac{1}{9}$$

Soit B l'événement : « on obtient 3 au premier lancer », alors :

$$B = \{(3,1), (3,2), (3,3), (3,4), (3,5), (3,6)\}$$

donc :

$$\mathbb{P}(B) = \frac{6}{36} = \frac{1}{6}$$

Jusqu'ici, tout roule. Posons-nous maintenant le problème suivant : sachant que B est réalisé, c'est-à-dire que l'on obtient 3 au premier lancer, quelle est la probabilité que le total fasse 9 ? On ne considère plus tous les éléments de Ω . Il semble alors nécessaire de définir un nouvel univers (notre nouvel ensemble des possibles) :

$$\Omega' = \{(3,1), (3,2), (3,3), (3,4), (3,5), (3,6)\} = \Omega \cap B$$

Puisque l'univers change, la probabilité aussi. L'événement A' « le total fait neuf » dans ce nouveau modèle s'écrit :

$$A' = \{(3,6)\} = A \cap B$$

et donc :

$$\mathbb{P}'(A') = \frac{1}{6} = \frac{\text{nombre d'éléments de } A \cap B}{\text{nombre d'éléments de } B} = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

— Définition

Cela nous incite à poser la définition suivante :

Définition : probabilité conditionnelle

Soit A et B deux événements d'un espace probabilisé muni d'une probabilité \mathbb{P} , avec $\mathbb{P}(B) \neq 0$. La **probabilité de A sachant B** est définie par :

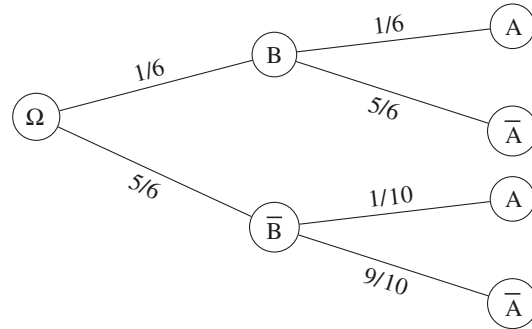
$$\mathbb{P}_B(A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

On parle de *probabilité conditionnelle*. Mais est-ce bien une probabilité ? C'est-à-dire est-ce que $(\Omega, \mathcal{F}, \mathbb{P}_B)$ est un espace probabilisé ? D'après la définition page 136 (axiomes de KOLMOGOROV), il y a trois conditions à vérifier (faites-le !...).

— Arbre

Reprenons l'exemple de la page précédente en répondant aux mêmes questions à l'aide d'un arbre *pondéré*, c'est-à-dire un arbre dont chaque branche est marquée de la probabilité (du *poids*) correspondant.

Alors la somme des probabilités de chaque « ramification » est égale à 1.



À l'aide de la formule des probabilités totales, il est aisé d'obtenir les résultats suivant :

1. Calcul de $\mathbb{P}(A)$

B et \bar{B} forment une partition. D'après la formule des probabilités totales :

$$\mathbb{P}(A) = \mathbb{P}(B \cap A) + \mathbb{P}(\bar{B} \cap A) = \mathbb{P}(B) \times \mathbb{P}_B(A) + \mathbb{P}(\bar{B}) \times \mathbb{P}_{\bar{B}}(A).$$

$$\mathbb{P}(A) = \frac{1}{6} \times \frac{1}{6} + \frac{5}{6} \times \frac{1}{10} = \frac{1}{9}.$$

2. Calcul de $\mathbb{P}(\bar{A})$

B et \bar{B} forment une partition. D'après la formule des probabilités totales :

$$\mathbb{P}(\bar{A}) = \mathbb{P}(B \cap \bar{A}) + \mathbb{P}(\bar{B} \cap \bar{A}) = \mathbb{P}(B) \times \mathbb{P}_B(\bar{A}) + \mathbb{P}(\bar{B}) \times \mathbb{P}_{\bar{B}}(\bar{A}).$$

$$\mathbb{P}(\bar{A}) = \frac{1}{6} \times \frac{5}{6} + \frac{5}{6} \times \frac{9}{10} = \frac{8}{9}.$$

3. Probabilités conditionnelles

$$\mathbb{P}_A(B) = \frac{\mathbb{P}(B \cap A)}{\mathbb{P}(A)} = \frac{\frac{1}{36}}{\frac{1}{9}} = \frac{1}{4}.$$

$$\mathbb{P}_A(\bar{B}) = \frac{\mathbb{P}(\bar{B} \cap A)}{\mathbb{P}(A)} = \frac{\frac{1}{12}}{\frac{1}{9}} = \frac{3}{4}.$$

$$\mathbb{P}_{\bar{A}}(B) = \frac{\mathbb{P}(B \cap \bar{A})}{\mathbb{P}(\bar{A})} = \frac{\frac{5}{36}}{\frac{8}{9}} = \frac{5}{32}.$$

$$\mathbb{P}_{\bar{A}}(\bar{B}) = \frac{\mathbb{P}(\bar{B} \cap \bar{A})}{\mathbb{P}(\bar{A})} = \frac{\frac{3}{4}}{\frac{8}{9}} = \frac{27}{32}.$$

— Formule de BAYES

Cette formule correspond au « retournement » d'un arbre.

Théorème : deuxième formule de BAYES (probabilité des causes)

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé et $(A_i)_{i \in I \subseteq \mathbb{N}}$ un système complet d'événements.

$$\forall B \in \mathcal{F}, \quad \mathbb{P}_B(A_k) = \frac{\mathbb{P}_{A_k}(B) \times \mathbb{P}(A_k)}{\sum_{i \in I} \mathbb{P}_{A_i}(B) \times \mathbb{P}(A_i)}$$

On utilisera souvent ce théorème dans le cas particulier de deux événements :

Théorème : première formule de BAYES

$$\mathbb{P}_B(A) = \frac{\mathbb{P}_A(B) \times \mathbb{P}(A)}{\mathbb{P}(B)}$$

On aurait pu écrire :

$$\mathbb{P}_{\text{effet}}(\text{cause}) = \frac{\mathbb{P}_{\text{cause}}(\text{effet}) \times \mathbb{P}(\text{cause})}{\mathbb{P}(\text{effet})}$$

Statistiquement, on a accès aux effets connaissant les causes (c'est le déterminisme...) mais dans la réalité, on subit les effets et on voudrait en connaître la cause.

Le cas le plus célèbre d'utilisation de la *statistique bayésienne* est le test de grossesse. Les médecins « disposent » d'un échantillon de femmes enceintes ou non (et donc de $\mathbb{P}(E)$ et $\mathbb{P}(\bar{E})$) et d'un test de grossesse. Ils font subir le test à tout l'échantillon et disposent donc de statistiques sur $\mathbb{P}_E(T)$, $\mathbb{P}_{\bar{E}}(\bar{T})$, $\mathbb{P}_{\bar{E}}(T)$ et $\mathbb{P}_E(\bar{T})$ mais ce qui intéresse le médecin et surtout la femme qui utilise le test c'est $\mathbb{P}_T(E)$ et $\mathbb{P}_{\bar{T}}(\bar{E})$...

— Indépendance

La notions d'événements indépendants est l'une des difficultés du calcul des probabilités. Comme souvent, cette notion purement mathématique renvoie, par son appellation, à une notion intuitive utilisée dans le langage courant. Il faut bien garder en mémoire que le mot du vocabulaire courant est souvent un « faux ami ». Donnons tout d'abord sa définition.

Définition : événements indépendants

Les événements A et B sont dits **indépendants** si $\mathbb{P}(A \cap B) = \mathbb{P}(A) \times \mathbb{P}(B)$.

Ainsi, dans notre exemple de dés, on a $\mathbb{P}(A) \times \mathbb{P}(B) = \frac{1}{6} \times \frac{6}{36} = \frac{1}{6} = \mathbb{P}(A \cap B)$, donc le lancer du 2^e dé est indépendant du premier, ce qui est rassurant.

Cette notion est purement abstraite et ne renvoie qu'à des propriétés mathématiques dont la principale est :

Théorème : indépendance et probabilités conditionnelles

Soit A et B deux événements tels que $\mathbb{P}(B) \neq 0$. Les événements A et B sont indépendants si et seulement si $\mathbb{P}_B(A) = \mathbb{P}(A)$.

On peut faire dire ce que l'on veut à des probabilités selon le modèle choisi.

Supposons que sur un groupe de 100 personnes, 20 portent des sous-vêtements bleus, 50 jouent au football et 10 font les deux à la fois. On met ces 100 personnes dans une boîte et on en tire une au hasard. Vérifiez que les événements « la personne tirée porte des sous-vêtements bleus » et « la personne tirée joue au football » sont indépendants.

Étudiez le même problème en considérant cette fois-ci que 15 personnes jouent au football (pourquoi pas ?)².

Veillez à ne pas confondre évènements *indépendants* et évènements *incompatibles*.

On peut montrer d'ailleurs que deux évènements incompatibles de probabilité non nulle ne sont jamais indépendants.

En effet, A et B sont incompatibles si, et seulement si, $A \cap B = \emptyset$, donc $\mathbb{P}(A \cap B) = \mathbb{P}(\emptyset) = 0$. Si $\mathbb{P}(A) \neq 0$ et $\mathbb{P}(B) \neq 0$, on a forcément $\mathbb{P}(A \cap B) \neq \mathbb{P}(A) \times \mathbb{P}(B)$.

La seule idée à retenir est que, si A et B sont indépendants, *avoir observé la réalisation de A ne modifie pas la probabilité d'une éventuelle réalisation de B*.

Ainsi, en supposant que la Française des Jeux n'utilise pas de boules truquées, on peut considérer que deux tirages successifs du loto sont indépendants.

Théorème : indépendance mutuelle

Soit $(A_i)_{i \in I \subseteq \mathbb{N}}$ une suite d'évènements de \mathcal{T} .

1. On dit que ces évènements sont **indépendants deux à deux** si, et seulement si :

$$\forall (i, j) \in I^2, (i \neq j), \mathbb{P}(A_i \cap A_j) = \mathbb{P}(A_i) \times \mathbb{P}(A_j)$$

2. On dit que ces évènements sont **mutuellement indépendants** si, et seulement si, pour toute partie J de I on a :

$$\mathbb{P}\left(\bigcap_{j \in J} A_j\right) = \prod_{j \in J} \mathbb{P}(A_j)$$

5.10 Variables aléatoires réelles finies

On travaillera en général dans l'espace probabilisé $(\Omega, \mathcal{T}, \mathbb{P})$ avec $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ un univers fini de cardinal n.

Définition

L'ensemble $\mathfrak{B}(\mathbb{R})$ est bien une tribu mais il pose quelques problèmes lors d'une étude plus approfondie des probabilités. Dans \mathbb{R} , on utilisera donc une tribu particulière qui correspond en fait à la tribu à laquelle on pense « naturellement », celle créée à partir des intervalles :

Définition : tribu borélienne

On note $\mathbf{B}(\mathbb{R})$ l'ensemble des parties engendrées (par intersections et réunions dénombrables) par les intervalles de \mathbb{R} .

Vous vérifierez qu'il s'agit d'une tribu.

Définition : variable aléatoire réelle

On appelle variable aléatoire réelle finie toute application de Ω dans \mathbb{R} vérifiant :

$$\forall I \subseteq \mathbb{R}, X^{-1}(I) \in \mathcal{T}$$

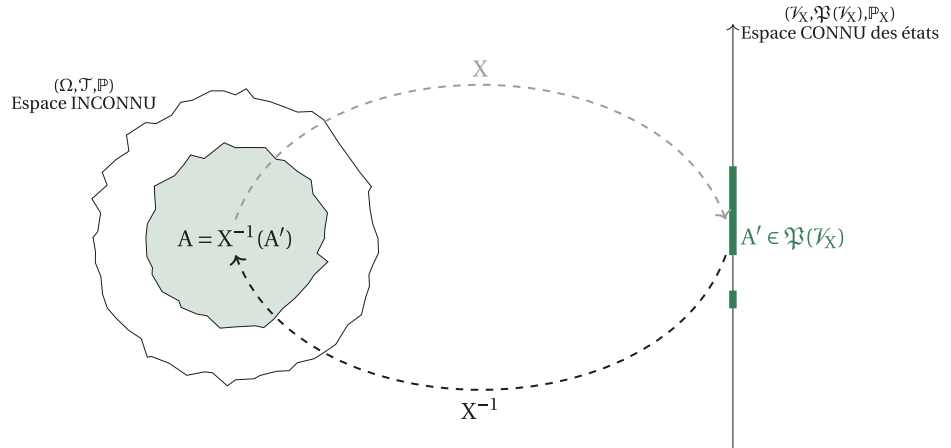
2. De façon générale, la définition probabiliste de l'indépendance est plus large que la notion intuitive.



Cette définition nous assure que $X^{-1}(I)$ est un évènement dont on peut calculer la probabilité. Une v.a. est *réelle* car X est à valeurs dans \mathbb{R} et *finie* car l'univers est fini donc l'ensemble image de X aussi.

Vous noterez qu'une *variable* aléatoire réelle (v.a.r. pour les intimes) est en fait une fonction !...

Nous noterons $\mathcal{V}_X = X(\Omega)$ l'ensemble des valeurs prises par la v.a. X .



Soit x_1, \dots, x_k les différentes valeurs prises par la fonction X .

On note $\{X = x_i\}$ l'évènement « la variable aléatoire prend la valeur x_i ». Il se note rigoureusement $\{\omega \in \Omega \mid X(\omega) = x_i\}$, ce qui se lit « l'ensemble des ω tels que $X(\omega) = x_i$ » ou encore $X^{-1}(\{x_i\})$.

La v.a.r. X induit une nouvelle probabilité (ou plutôt loi de probabilité comme nous allons le voir) \mathbb{P}_X sur $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Par exemple, $\mathbb{P}_X(\{2\}) = \mathbb{P}(\{\omega \in \Omega \mid X(\omega) = 2\})$.

La définition précédente n'est pas très parlante, il faut essentiellement retenir que l'on va s'intéresser à des évènements décrits par des nombres, les variables aléatoires nous permettent de décrire, et donc de noter, facilement des évènements à l'aide de nombres.

Si \mathcal{V}_X est réduit à un singleton, $\mathcal{V}_X = \{\alpha\}$, on dit que X est une variable certaine, on note alors $X = \alpha$.

Loi de probabilité

Définition : loi de probabilité d'une v.a.r.

Soit X une variable aléatoire réelle sur Ω . On appelle **loi de probabilité** de X la fonction \mathbb{P}_X de $\mathcal{B}(\mathbb{R})$ dans $[0,1]$ définie par :

$$\begin{aligned} \mathbb{P}_X : \mathcal{B}(\mathbb{R}) &\rightarrow [0,1] \\ B &\mapsto \mathbb{P}(\{X^{-1}(B)\}) = \mathbb{P}(\{\omega \in \Omega \mid X(\omega) \in B\}) = \mathbb{P}(X \in B) = \mathbb{P}_X(B) \end{aligned}$$

Vérifiez que \mathbb{P}_X vérifie les axiomes de KOLMOGOROV donc que \mathbb{P}_X est bien une mesure de probabilité sur $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$.

Dans la plupart des cas, on étudiera $\mathbb{P}(X = x)$, c'est-à-dire $\mathbb{P}(\{\omega \in \Omega \mid X(\omega) = x\})$.

Si $x \notin X(\Omega)$, alors $\{X = x\} = \emptyset$ et donc $\mathbb{P}(\{X = x\}) = 0$.

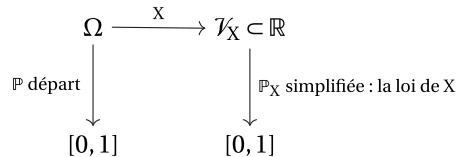
Définir la loi de probabilité d'une expérience aléatoire reviendra donc à :

- déterminer toutes les valeurs possibles x_1, \dots, x_n prises par X ;
- calculer les probabilités p_1, \dots, p_n des événements correspondants ;
- regrouper les résultats dans un tableau du type :

Valeurs prises par X	x_1	x_2	\dots	x_n
Probabilité correspondante $\mathbb{P}(\{X = x_i\})$	p_1	p_2	\dots	p_n

Vous n'oublierez pas de vérifier que $p_1 + \dots + p_n = 1$ d'après la loi des probabilités totales.

En général, on se trouve dans la situation suivante :



Remarque

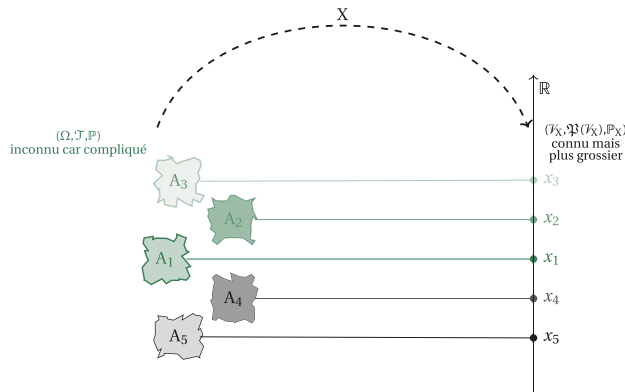
Dans la vraie vie, on connaît les valeurs prises par la v.a.r. mais on ne connaît pas bien Ω , voire pas du tout. On réfléchit donc sur les lois et non pas sur des ensembles comme au lycée. Par exemple, on peut vouloir réfléchir au nombre de plantages d'un réseau. On estime statistiquement les probabilités de tomber 0, 1, 2, etc. fois en panne dans une journée. On ne sait pas trop ce qu'est Ω mais on peut parler de la loi de X .

En fait, une variable aléatoire « transporte » les calculs de probabilités d'un univers inconnu vers des valeurs réelles connues.

La v.a.r. a pour effet de remplacer la tribu \mathcal{T} par une tribu $X^{-1}(\mathfrak{P}(\mathcal{V}_X))$ en général plus petite : elle condense l'information sur l'univers en une information plus grossière (numérique) mais c'est déjà bien car en général on ne connaît de toute manière pas ou peu l'univers de départ !...

On retiendra donc que le « praticien » travaille la plupart du temps avec des lois de probabilités et non pas des ensembles : cela simplifie son travail car il peut travailler sur des espaces complexes (l'univers de départ est souvent trop complexe voire inconnu) sans les connaître.

L'information donnée par la loi est souvent plus grossière : par exemple, on lance deux dés et on regarde la somme des deux faces. La tribu de départ contient 2^{36} événements alors que la tribu de la loi n'en contient que 2^{11} (il y a 11 sommes possibles).





À partir de la loi, on induit un système complet d'événements sur Ω : la connaissance de la loi nous donne des renseignements sur l'univers mais souvent de manière grossière (par « paquets »).

Deux v.a.r. peuvent avoir la même loi sans être égales ! On lance n fois une pièce non truquée. Considérez X le nombre de pile tombés et $n - X$ le nombre de face. Ces deux v.a.r. suivent la même loi binomiale de mêmes paramètres n et $1/2$ et pourtant elles ne sont pas égales (ni même égales presque partout ce qui signifie sauf sur un événement de probabilité nulle mais ceci est une autre histoire).

Variables aléatoires indépendantes

La notation $\mathbb{P}(X = x_i, Y = y_j)$ signifie usuellement $\mathbb{P}(\{X = x_i\} \cap \{Y = y_j\}) = \mathbb{P}(X = x_i \text{ et } Y = y_j)$. On peut donc définir un **vecteur aléatoire** $Z = (X, Y)$ tel que :

$$\mathbb{P}(Z = (x_i, y_j)) = \mathbb{P}(X = x_i, Y = y_j)$$

On peut se représenter l'espace produit à l'aide d'une représentation cartésienne : on parle alors d'événements « parallèles » à des « axes » (les Ω_i), ces « axes » étant « orthogonaux ». Un système complet d'événements est alors une « base ». Tous ces ponts de vocabulaire reliant plusieurs théories est un plaisir évident pour le mathématicien mais vous pouvez laisser cela de côté dans un premier temps...

Définition : couple de variables indépendantes

Les v.a.r. X et Y sont dites *indépendantes* si, et seulement si, pour toute partie $(A, B) \in \mathcal{V}_X \times \mathcal{V}_Y$:

$$\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A) \times \mathbb{P}(Y \in B)$$

Définition : variables aléatoires mutuellement indépendantes

Si $X_1, X_2, X_3, \dots, X_k$ sont k v.a. définies sur Ω , on dit qu'elles sont *mutuellement indépendantes* si, et seulement si, quels que soient $(I_1, I_2, \dots, I_k) \in \mathcal{V}_{X_1} \times \mathcal{V}_{X_2} \times \dots \times \mathcal{V}_{X_k}$ on a :

$$\mathbb{P}(X_1 \in I_1, X_2 \in I_2, \dots, X_k \in I_k) = \mathbb{P}(X_1 \in I_1) \times \mathbb{P}(X_2 \in I_2) \times \dots \times \mathbb{P}(X_k \in I_k)$$



L'indépendance deux à deux de variables aléatoires n'entraîne pas leur indépendance mutuelle.

Espérance mathématique

Définition : espérance mathématique

On appelle *espérance mathématique* de la variable aléatoire discrète X le nombre noté $\mathbb{E}(X)$ défini par :

$$\mathbb{E}(X) = x_1 \cdot \mathbb{P}_X(x_1) + x_2 \cdot \mathbb{P}_X(x_2) + \dots + x_n \cdot \mathbb{P}_X(x_n) = \sum_{i=1}^n x_i \cdot \mathbb{P}_X(x_i)$$

Exemple : on lance un dé honnête. On définit la variable aléatoire X qui prend la valeur 2 si le numéro du dé est pair et 1 sinon.

Notons ω_i l'événement « le numéro de la face est i », alors $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}$ et

i	1	2	3	4	5	6
$\mathbb{P}(\omega_i)$	1/6	1/6	1/6	1/6	1/6	1/6
$X(\omega_i)$	1	2	1	2	1	2

D'après le théorème précédent, on a

$$\mathbb{E}(X) = 1 \times \frac{1}{6} + 2 \times \frac{1}{6} + 1 \times \frac{1}{6} + 2 \times \frac{1}{6} + 1 \times \frac{1}{6} + 2 \times \frac{1}{6} = \frac{3}{2}$$

Nous aurions pu procéder autrement pour définir la loi de probabilité :

Valeurs x_i prises par X	1	2
$\mathbb{P}_X(x_i)$	1/2	1/2

alors, d'après la définition, $\mathbb{E}(X) = 1 \times \frac{1}{2} + 2 \times \frac{1}{2} = \frac{3}{2}$.

Théorème : espérance de $f \circ X$

Soit X une v.a.r. Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ telle que $f \circ X$ soit une v.a.r. alors :

$$\mathbb{E}(f \circ X) = \sum_{n \in \mathbb{N}} f(x_n) \cdot \mathbb{P}(X = x_n) = \sum_{n \in \mathbb{N}} f(x_n) \cdot \mathbb{P}_X(x_n)$$

Variance

La variance mesure l'« écart » par rapport à l'espérance en faisant la somme des carrés des « distances » entre chaque valeur prise par la variable aléatoire et l'espérance pondérées par la probabilité correspondante, ce qui donne :

Définition : variance

$$\mathbb{V}(X) = \sum_{i=1}^n (x_i - \mathbb{E}(X))^2 \cdot \mathbb{P}(X = x_i) = \sum_{i=1}^n (x_i - \mathbb{E}(X))^2 \cdot \mathbb{P}_X(x_i)$$

On a choisi d'utiliser les carrés de manière arbitraire pour ne pas avoir de problèmes de signes ; on aurait pu choisir une autre méthode mais celle-ci a l'avantage de rappeler la distance euclidienne bien connue. La variance est en ce sens homogène au carré d'une distance. On va donc définir une distance proprement dite en en prenant la racine carrée : c'est ce qu'on appelle l'**écart-type**.

Définition : écart-type

$$\sigma(X) = \sqrt{\mathbb{V}(X)}$$

Vous pouvez obtenir espérance, variance et écart-type très simplement à l'aide des modules statistiques de vos calculatrices. Il suffit de rentrer les valeurs prises par la variable aléatoire en liste 1, les probabilités correspondantes en liste 2.

Linéarité de l'espérance

À partir de variables aléatoires existantes, on peut en créer de nouvelles.

Avec des notations usuelles on obtient :

- $aX + b : \omega_i \mapsto aX(\omega_i) + b$ avec a et b des réels ;
- $X + Y : \omega_i \mapsto X(\omega_i) + Y(\omega_i)$.

On peut alors démontrer les propriétés suivantes :

Théorème : linéarité de l'espérance

- $\mathbb{E}(aX + b) = \sum_{i=1}^n (ax_i + b) \cdot \mathbb{P}_X(x_i) = a\mathbb{E}(X) + b$
- $\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$

Les démonstrations sont similaires. Notons $Y = aX + b$.

Pour tout $\omega_i \in \Omega$ on a $Y(\omega_i) = aX(\omega_i) + b$ et donc $y_i = ax_i + b$ avec les notations usuelles donc $\{\omega \in \Omega | Y(\omega) = y_i\} = \{\omega \in \Omega | aX(\omega) + b = ax_i + b\} = \{\omega \in \Omega | X(\omega) = x_i\}$.

Alors :

$$\begin{aligned} \mathbb{E}(Y) &= \sum_{i=1}^n y_i \mathbb{P}(Y = y_i) = \sum_{i=1}^n (ax_i + b) \mathbb{P}(X = x_i) \\ &= a \sum_{i=1}^n x_i \mathbb{P}(X = x_i) + b \sum_{i=1}^n \mathbb{P}(X = x_i) \end{aligned}$$

On en déduit que $\mathbb{E}(Y) = a\mathbb{E}(X) + b$.

— Théorème de KÖNIG-HUYGENS

Johann KÖNIG (1712-1757) était un mathématicien allemand, élève de Jean BERNOULLI, et Christian HUYGENS (1629-1695) était quant à lui néerlandais.

Théorème de König-Huygens

$$\mathbb{V}(X) = \mathbb{E}\left((X - \mathbb{E}(X))^2\right) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$$

La démonstration est demandée au test 5.26.

— 5.11 Quelques lois discrètes classiques

— Loi uniforme (discrète)

C'est la loi qui correspond au tirage équiprobable d'un jeton blanc dont on note le numéro dans une urne qui contient n jetons blancs numérotés.

Définition : loi uniforme

On dit que $X : \Omega \rightarrow \llbracket 1, n \rrbracket$ suit une **loi uniforme** sur $\llbracket 1, n \rrbracket$ et on note $X \rightsquigarrow \mathcal{U}(\llbracket 1, n \rrbracket)$ si :

$$X(\Omega) = \llbracket 1, n \rrbracket \quad \text{et} \quad \forall k \in \llbracket 1, n \rrbracket, \mathbb{P}(X = k) = \frac{1}{n}$$

Calculons l'espérance et la variance de $X \rightsquigarrow \mathcal{U}(\llbracket 1, n \rrbracket)$.

$$\mathbb{E}(X) = \sum_{k=1}^n k \mathbb{P}(X=k) = \sum_{k=1}^n k \frac{1}{n} = \frac{1}{n} \sum_{k=1}^n k = \frac{1}{n} \frac{n(n+1)}{2} = \frac{n+1}{2}.$$

De manière analogue, on obtient que $\mathbb{V}(X) = \frac{n^2-1}{12}$, sachant que $\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$.

Loi de BERNOULLI : to be or not to be

On place des jetons blancs et des noirs dans une urne. On suppose que toutes les conditions sont réunies pour effectuer le tirage d'un jeton dans le cadre d'équiprobabilité et que la proportion de jetons blancs est p . Que dire de la variable aléatoire qui vaut 1 si le jeton tiré est blanc et 0 s'il est noir ?

Définition : loi de BERNOULLI

On dit que la v.a.r. finie X suit la **loi de BERNOULLI de paramètre p** si :

- $p \in]0 ; 1[$, $X(\Omega) = \{0, 1\}$;
- $\mathbb{P}(X=1) = p$.

On note alors $X \rightsquigarrow \mathcal{B}(1, p)$.

On dit aussi que X est une **variable aléatoire de BERNOULLI**. Sa loi de probabilité est simple à décrire :

x_i	0	1
$\mathbb{P}(X = x_i)$	$1 - p$	p

Il est simple de déterminer son espérance et sa variance :

- $\mathbb{E}(X) = 0 \times \mathbb{P}(X=0) + 1 \times \mathbb{P}(X=1) = p$;
- $\mathbb{V}(X) = (0-p)^2 \times \mathbb{P}(X=0) + (1-p)^2 \times \mathbb{P}(X=1)$
 $= p^2(1-p) + (1-p)^2 p = p(1-p)$.

Loi binomiale

Dans les mêmes conditions que précédemment, on effectue n tirages d'un jeton avec remise. À chaque tirage on associe une v.a.r. indépendante des autres qui suit la loi de BERNOULLI de paramètre p . Les issues élémentaires ω de ces n tirages sont des « mots » de n lettres, chaque lettre étant un B (pour blanc) ou un \bar{B} . On définit alors la variable aléatoire X à valeurs dans $\{0, 1, 2, \dots, n\}$, donnant le nombre de jetons blancs de ces issues élémentaires, c'est-à-dire que X est la somme des v.a.r. finies élémentaires suivant la loi de BERNOULLI de manière indépendante.

Définition : loi binomiale

On dit que la v.a.r. finie X suit la **loi binomiale de paramètres n et p** si $n \in \mathbb{N} \setminus \{0\}$, $p \in]0 ; 1[$, $X(\Omega) = \llbracket 0, n \rrbracket$ et si l'expérience est une suite de expériences de BERNOULLI de paramètre p menées indépendamment.

On note alors $X \rightsquigarrow \mathcal{B}(n, p)$.

On s'intéresse maintenant à l'évènement $(X = k)$, c'est-à-dire l'ensemble des mots de n lettres écrits avec k B et $n - k$ \bar{B} . Notons B_j l'évènement « obtenir un jeton blanc au j^e tirage ». Chaque tirage d'une boule étant indépendant des autres, on a :

$$\mathbb{P}(\omega) = \mathbb{P}(B_1) \times \cdots \times \mathbb{P}(B_k) \times \mathbb{P}(\overline{B_{k+1}}) \times \cdots \times \mathbb{P}(\overline{B_n}) = p^k (1-p)^{n-k}$$

Il reste à déterminer combien il y a de tels mots. Ce sont les anagrammes (permutations distinctes) du mot de n lettres $BB \dots B\bar{B}\bar{B} \dots \bar{B}$. Il y en a $n!$ qu'il faut diviser par les $k!$ permutations des B et les $(n-k)!$ permutations des \bar{B} , c'est-à-dire C_n^k .

Finalement, on obtient :

Théorème

Pour toute v.a.r. finie $X \rightsquigarrow \mathcal{B}(n, p)$, on a, pour tout $k \in X(\Omega)$:

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Vous démontrerez que $\mathbb{E}(X) = np$ et $\mathbb{V}(X) = np(1-p)$.

Loi hypergéométrique

Modèle : on considère une urne contenant N boules ($N \in \mathbb{N}^*$), la proportion de boules blanches dans l'urne étant p . On tire simultanément n boules (les boules sont choisies au hasard) et X est la v.a. qui indique le nombre de boules blanches tirées. On dit que X suit la loi hypergéométrique de paramètres N, n et p notée $\mathcal{H}(N, n, p)$.

On retiendra :

- X peut prendre toute valeur entière de l'intervalle $\{\max\{0, n - N(1-p)\}, \min\{n, Np\}\}$
- $k \in \mathcal{V}_X, \mathbb{P}(\{X = k\}) = \frac{\binom{Np}{k} \binom{N(1-p)}{n-k}}{\binom{N}{n}}$
- $\mathbb{E}(X) = np, \mathbb{V}(X) = np(1-p) \frac{N-n}{N-1}$

Loi géométrique

Modèle : on répète indéfiniment une épreuve de BERNOULLI dont la probabilité de succès est p . Les répétitions étant mutuellement indépendantes, X désigne le rang de l'essai donnant pour la première fois un succès. On dit que X suit la loi géométrique de paramètre p et on écrit $X \rightsquigarrow \mathcal{G}(p)$. On retiendra :

$$\begin{aligned} \mathcal{V}_X &= \mathbb{N}^* \\ k \in \mathbb{N}^*, \mathbb{P}(\{X = k\}) &= p(1-p)^{k-1} \\ \mathbb{E}(X) &= \frac{1}{p}, \mathbb{V}(X) = \frac{1-p}{p^2} \end{aligned}$$

La loi géométrique modélise donc le **temps d'attente jusqu'au premier succès**.

Problème du collectionneur

Voici une application classique de la loi géométrique : en Syldavie, le petit Ivan collectionne les images du chocolat Wonka. Il y a images différentes. Lorsqu'il aura obtenu les images, il aura droit de visiter les usines Wonka. À chaque fois qu'il achète une tablette, la probabilité d'obtenir une image quelconque est $1/n$. Soit T_n le temps d'attente (en nombre de tablettes achetées) pour obtenir les n images. Calculez $\mathbb{E}(T_n)$.

Soit X_i la variable aléatoire correspondant au temps d'attente pour passer de $i - 1$ images différentes à i images différentes. Quand on est dans l'état $i - 1$, on dispose de $i - 1$ images différentes. Si on tombe sur l'une des $i - 1$ images dont on dispose (avec la probabilité $\frac{i-1}{n}$), on reste dans l'état $i - 1$, sinon, on passe à l'état i (avec la probabilité $\frac{n-(i-1)}{n}$).

Ainsi, $X_i \rightsquigarrow \mathcal{G}(p_i)$ avec $p_i = \frac{n-(i-1)}{n}$. On en déduit que $\mathbb{E}(X_i) = \frac{1}{p_i} = \frac{n}{n+1-i}$.

Or, $T_n = X_1 + X_2 + \dots + X_n$, donc $\mathbb{E}(T_n) = \mathbb{E}(X_1) + \mathbb{E}(X_2) + \dots + \mathbb{E}(X_n)$:

$$\mathbb{E}(X_n) = \sum_{i=1}^n \frac{1}{p_i} = n \sum_{i=1}^n \frac{1}{n+1-i} = n \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{2} + 1 \right)$$

Par exemple, avec $n = 10$:

Il faut en moyenne acheter 30 tablettes pour obtenir les dix images.

5.12 Exercices

— Dénombrements

Test 5.1 (Octets - ♥)

Combien y a-t-il d'octets commençant par 1 ou se terminant par 00 ?

Test 5.2 (Téléphone - ♥)

1. On suppose que les numéros de téléphone sont de la forme 0Z-YX-XX-XX-XX avec X un nombre entier entre 0 et 9, Y un nombre entier entre 2 et 9 et Z un entier entre 1 et 9. En France (données du quatrième trimestre 2011), il y a environ 40 millions d'abonnements à un « fixe » (Z étant compris entre 1 et 5) et 69 millions à un « mobile » (Z valant 6 ou 7).

Est-on tranquille ? Pourquoi existe-t-il des numéros de mobile commençant par 07 ?

2. Supposons qu'en Syldavie il y ait 109 millions d'abonnés à un téléphone fixe et que le Y soit fixé par opérateur et qu'il n'y ait que deux opérateurs. De combien de zones a-t-on besoin ?

Test 5.3 (Nombre d'applications - ♥♥)

Combien y a-t-il d'applications d'un ensemble E de cardinal n vers un ensemble F de cardinal p ?

Test 5.4 (Barbie - ♥♥♥)

Un étudiant de Math Sup du lycée Ottokar de Klow en Syldavie a reçu pour Noël sept Barbie et sept séries de dix robes de couleurs différentes. Il habille chaque Barbie d'une de ces robes.

1. Combien y a-t-il d'habillages possibles selon la couleur ?
2. Combien y a-t-il d'habillages de sorte que chaque Barbie ait une robe de couleur différente des autres ?

3. Combien y a-t-il d'habillages de sorte qu'au moins deux Barbie aient une robe de la même couleur ?
4. Combien y a-t-il d'habillages de sorte qu'exactement deux Barbie aient une robe de la même couleur ?

Test 5.5 (Jus - ♥♥)

Lors de la soirée d'intégration des Math Sup du lycée Ottokar de Klow, 9 étudiants ont bu un verre d'eau, 25 un verre de jus de tomate, 14 un verre de jus de betterave, 7 un verre de jus de tomate et un verre d'eau, 4 un verre d'eau et un verre de jus de betterave, 10 un verre de jus de betterave et un verre de jus de tomate. Les étudiants de Math Sup s'occupaient des boissons et ceux de BCPST des entrées or ces derniers ont perdu leur liste et ne savent plus combien d'étudiants ont participé à la soirée. Les étudiants de Math Sup, connaissant les statistiques sur les consommations de boissons, pourront-ils corriger la gaffe des BCPST sachant que tout le monde a bu et que lors de la valse, tous les participants dansaient en couple ?

Test 5.6 (k-listes - ♥♥)

Soit E un ensemble fini non vide de cardinal n .

- a) Combien y a-t-il de parties de E formées de k éléments ?
- b) Combien y a-t-il de k -uplets d'éléments de E ?
- c) Combien y a-t-il de k -uplets d'éléments deux à deux distincts de E ?
- d) Combien y a-t-il de k -uplets d'éléments deux à deux distincts de E , tel que le premier élément est le plus petit et le dernier élément est le plus grand ?
- e) Combien y a-t-il de k -uplets d'éléments de E ordonnés dans l'ordre strictement croissant ?

Test 5.7 (Poker - ♥♥)

Une main au poker est constituée de 5 cartes tirées d'un jeu de 32 cartes. Combien y a-t-il de mains contenant des carrés (XXXXY) ? des fulls (XXXYY) ? des brelans (XXXYZ) ? des doubles paires (XX YYZ) ? des paires (XXYZA) ?

Deux lettres identiques (par exemple XX) correspondent à deux cartes de même hauteur (par exemple deux dames).

Test 5.8 (Mélange de cartes - ♥♥♥)

On dispose d'un jeu de 52 cartes et on le mélange en *queue d'aronde* : on divise le jeu en deux parties de 26 cartes et on intercale les cartes de la moitié gauche dans la moitié droite sans modifier l'ordre de chaque partie bien sûr. Est-ce que quatre mélanges en queue d'aronde successifs permettent de fournir un ordre aléatoire des cartes ?

Il faudrait préciser d'abord cette notion : combien y a-t-il d'ordres possibles des cartes ?

Existe-t-il alors des ordres qui ne pourront être obtenus par notre succession de mélanges ?

Pour répondre à cette question, il faudrait calculer le nombre d'ordres qu'on peut obtenir par la succession de quatre mélanges... Imaginez que vous disposiez de 26 cartes bleues et 26 rouges. Vous les mélangez avec la méthode de la queue d'aronde : pouvez-vous reconstituer le jeu initial ? Combien y a-t-il alors d'ordres possibles après le premier mélange ?

Test 5.9 (Suissesses multinomiales - ♥♥♥♥)

Combien existe-il d'anagrammes au mot « suissesses » ?

Pour répondre à cette question on définit un nouveau type de coefficient :

Définition : coefficient multinomial

Soit k_1, k_2, \dots, k_p des entiers naturels tels que $\sum_{i=1}^p k_i = n$. On note alors :

$$C_n^{k_1, k_2, \dots, k_p} = \frac{n!}{k_1! k_2! \dots k_p!}$$

1. Quel est le coefficient multinomial égal à $\binom{n}{p}$?
2. Comment interpréter ces coefficients en termes combinatoires ? Quel est le rapport avec les Suissesses ?
3. Il existe un pendant multinomial à la formule du binôme :

Théorème : formule du multinôme

$$(x_1 + x_2 + \dots + x_p)^n = \sum_{k_1 + k_2 + \dots + k_p = n} C_n^{k_1, k_2, \dots, k_p} x_1^{k_1} x_2^{k_2} \dots x_p^{k_p}$$

Par exemple, quel est le coefficient de $x^2 y^5 z$ dans $(x + y + z)^8$?

Test 5.10 (L'âge du capitaine - ♥♥)

Le capitaine des pompiers de New York est marié et a quatre enfants dont un mange un yaourt aux fruits tous les matins. Il ne fume pas, aime regarder des films de gladiateurs et réside à l'angle de la 1^{re} avenue et de la 33^e rue. La caserne se trouve à l'angle de la 9^e avenue et de la 40^e rue. Il s'y rend tous les jours à pied en sifflant « *Im schönsten Westertal* » et sans perdre de temps (i.e. dans le sens des numéros croissants aussi bien pour les rues que pour les avenues). Sachant qu'il a commencé à travailler le jour de ses 18 ans, et sachant qu'il n'est jamais passé deux fois par le même chemin à l'aller, qu'il boîte légèrement de la jambe droite après avoir participé au championnat d'Écosse de lancement d'enclume, qu'il est sourd de l'oreille gauche depuis qu'il a plongé dans un lac gelé pour sauver son petit frère John qui était en train de se noyer après que la glace sur laquelle il patinait craqua, quel est l'âge (maximum) du capitaine ?

— Probabilités

Test 5.11 (Formule du crible de POINCARÉ : cas de trois événements - ♥)

Déterminez $\mathbb{P}(A \cup B \cup C)$ en fonction des probabilités de $A, B, C, A \cap B, A \cap C, B \cap C$ et $A \cap B \cap C$.

Test 5.12 (Définition d'une probabilité - ♥♥♥♥)

Soit $\Omega = \{\omega_1, \omega_2, \dots\}$ un ensemble infini dénombrable, les ω_i étant distincts deux à deux. On

suppose que, pour tout entier naturel non nul k , $\mathbb{P}(\{\omega_k\}) = \frac{\lambda}{37^k}$ avec λ un réel.

Existe-il une valeur de λ telle que $(\Omega, \mathbb{P}(\Omega), \mathbb{P})$ soit un espace probabilisé ?

Test 5.13 (Problèmes historiques - ♥♥)

Tout a commencé au XVII^e siècle, lorsque les princes demandèrent aux mathématiciens de les aider à gagner au jeu...

1. Le duc de Toscane demande à GALILÉE pourquoi, en lançant trois dés, on obtient plus souvent un total de 10 qu'un total de 9, alors qu'il y a dans les deux cas exactement 6 façons d'obtenir ces résultats.
2. Le chevalier de MÉRÉ soutient à PASCAL que les deux jeux suivants sont favorables au joueur : obtenir au moins un 6 en lançant 4 fois de suite un dé, et obtenir au moins un double 6 en lançant 24 fois de suite 2 dés.

Vous simulerez ces expériences à l'aide de **Python** et vous démontrerez les résultats observés.

Test 5.14 (Formule de B. et Monica B. - ♥)

En Syldavie, 70 % des habitants sont de sexe masculin et 60 % des hommes et 25 % des femmes ont une photo de Monica B. dans leur portefeuille. Un extra-terrestre enlève au hasard une personne syldave et découvre une photo de Monica B. dans son portefeuille : quelle est la probabilité que la personne enlevée soit une femme ?

Test 5.15 (Marylin et les chèvres - ♥♥♥♥)

Dans le numéro du 9 septembre 1990 de *Parade*, Marilyn VOS SAVANT, connue pour avoir un QI de 186, soit l'un des plus élevés au monde, répondit au courrier suivant : « *Suppose you're on a game show, and you're given the choice of three doors. Behind one door is a car, behind the others, goats. You pick a door, say number 1, and the host, who knows what's behind the doors, opens another door, say number 3, which has a goat. He says to you, "Do you want to pick door number 2?" Is it to your advantage to switch your choice of doors?* »

Craig. F. Whitaker Columbia, MD

Cette lettre décrit en fait le déroulement d'un jeu télévisé des années 1970, *Let's make a deal*, animé par Carol MERRIL et **Monty HALL** : c'est en référence à ce dernier qu'on donna son nom à ce problème célèbre (plus tard, un haut représentant de la culture française reprit ce jeu : ce fut Lagaf et son *Bigdil...*).

Marilyn répondit qu'il valait mieux pour le candidat changer de porte quand le présentateur l'y invitait. Qu'en pensez-vous ?

Vous pourrez également proposer une simulation avec **Python**.

Test 5.16 (Duel - ♥♥♥♥)

Alexandre P. et Évariste G. s'affrontent en duel au pistolet suivant les règles suivantes :

- Ils tirent chacun leur tour. Le premier qui atteint l'autre a gagné...
- Lorsqu'il tire, Alexandre atteint Évariste avec la probabilité a ($0 < a < 1$) et il le rate avec la probabilité $\bar{a} = 1 - a$.
- Lorsqu'il tire, Évariste atteint Alexandre avec la probabilité b ($0 < b < 1$) et il le rate avec la probabilité $\bar{b} = 1 - b$.
- Alexandre tire le premier. Ainsi, Alexandre (resp. Évariste) n'effectue que des tirs de rang impair (resp. pair).

On considère, pour tout entier $n \geq 1$, les événements A_{2n-1} : « Alexandre gagne à l'issue du tir numéro $2n - 1$ », B_{2n} : « Évariste gagne à l'issue du tir numéro $2n$ ».

1. Calculez, en fonction de a et b , les probabilités des événements A_1 , B_2 et A_3 . Plus généralement, calculez $\mathbb{P}(A_{2n-1})$ et $\mathbb{P}(B_{2n})$.
2. Pour $n \geq 1$, on note C_n (resp. D_n) l'évènement : « Alexandre (resp. Évariste) gagne à un tir dont le numéro est entre 1 et $2n - 1$ (resp. entre 2 et $2n$) ». Calculer $\mathbb{P}(C_n)$ et $\mathbb{P}(D_n)$.
3. Calculer $\alpha = \lim_{n \rightarrow +\infty} \mathbb{P}(C_n)$ et $\beta = \lim_{n \rightarrow +\infty} \mathbb{P}(D_n)$. Vérifiez que $\alpha + \beta = 1$. Comment interpréter ce résultat ?

Test 5.17 (To binomiale or not to binomiale ? - ♥)

Dites si les variables aléatoires X définies ci-dessous suivent une loi binomiale ? Donnez quand c'est possible $X(\Omega)$, les paramètres de la loi ainsi que l'espérance.

1. Dans une classe on tire au sort et sans remise 5 élèves. X est le nombre d'élèves travaillant pour les services secrets bordures dans le lot tiré au sort.
2. Dans un sac de 20 billes contenant 7 noires et 13 blanches, on tire avec remise 3 d'entre elles, X étant le nombre de billes noires obtenues.
3. On lance 10 dés, X est le nombre de « 5 » obtenus.
4. Un circuit comprend 32 lampes en série. Pour chacune d'elle, la probabilité qu'elle fonctionne est de $3/100$. X est le nombre de lampes qui s'allument lorsqu'on appuie sur l'interrupteur.
5. Même question avec cette fois des lampes en parallèle.

Test 5.18 (Lois usuelles - ♥)

Dans chacune des expériences aléatoires ci-dessous, reconnaître une loi usuelle pour la variable aléatoire X . Donner alors son espérance.

1. On choisit une carte au hasard dans un jeu de 32 cartes numérotées.
 X = « hauteur de la carte choisie »
2. Un dieu de l'Olympe tue chaque jour un Achéen (avec probabilité $2/3$) ou un Troyen (avec probabilité $1/3$). On suppose que le choix de la victime est indépendant de celle des autres. X = « nombre de Troyens tués sur 10 jours consécutifs ».
3. Sur la plaine d'Ilium se trouvent 15 Achéens et 30 Troyens. Un dieu de l'Olympe foudroie chaque matin un Achéen ou un Troyen, pris parmi les soldats de la plaine.
 X = « nombre de Troyens tués sur 10 jours consécutifs ».
4. On tire une à une les cartes d'un jeu de 32 cartes, jusqu'à obtenir le valet de pique.
 X = « nombre de tirage effectuées ».
5. On lance un dé à 6 faces. X = « chiffre obtenu ».
6. Monica a dans sa poche 7 allumettes et 12 pièces de monnaie. Elle met la main dans sa poche et en sort 5 objets. X = « nombre de pièces obtenues ».
7. Monica fait passer une audition pour trouver des partenaires pour son prochain film. 3000 candidats passent l'audition. Chaque candidat réussit son entretien avec probabilité $2/3$. X = « nombre de candidats sélectionnés par Monica ».
8. James dort chez son amie Monica, qui lui a prêté son appartement. Il dispose d'un trousseau de 7 clés, et ne sait pas laquelle ouvre la porte de l'appartement. Il les essaye une par une, en mettant de côté chaque clé essayée. X = « nombre d'essais nécessaires pour trouver la bonne clé ».

Test 5.19 (Loi hypergéométrique - ♥)

Justifiez la valeur donnée dans le cours page 148 (section « Loi hypergéométrique ») de $\mathbb{P}(\{X = k\})$ lorsque $X \rightsquigarrow \mathcal{H}(N, n, p)$.

Test 5.20 (HEC 2016 - ♥)

On tire avec remise une boule d'une urne contenant des boules numérotées de 1 à n . Quelle est la variable aléatoire V_n dont la fonction **Python** suivante calcule la simulation ?

```
import numpy.random as rd
def V(n: int) >
int:
A = []
compt = 0
while len(A) < n:
u = int(n * rd.rand() + 1)
i = [ind for ind, el in enumerate(A) if el == u] # liste des
# positions de u dans A
if len(i) == 0:
A.append(u)
compt = compt + 1
return compt
```

Test 5.21 (Centrale 2017 - ♥)

Un pion se trouve à l'instant 0 sur la case 0 d'un parcours linéaire dont les cases sont numérotées par les entiers consécutifs. À chaque étape il avance d'un nombre strictement positif aléatoire de cases. Pour $n \in \mathbb{N}^*$ on note Y_n le nombre de cases dont il avance à la n -ième étape. Ainsi $S_n = Y_1 + Y_2 + \dots + Y_n$ est sa position à l'instant n avec $S_0 = 0$ par convention. On suppose que les variables $(Y_n)_{n \in \mathbb{N}^*}$ suivent toutes la même loi de probabilité et sont indépendantes.

On suppose que $Y_1 - 1$ suit une loi de BERNOULLI de paramètre p et on choisit un entier $k \in \mathbb{N}^*$.

1. Écrire une fonction en **Python** prenant en argument les paramètres p et k et simulant l'expérience jusqu'à ce que le pion dépasse (au sens large) la position k . La fonction renverra 1 si le point a atterri sur la case k et 0 s'il l'a dépassée sans s'y arrêter.

Pour simuler une loi de BERNOULLI, on pourra utiliser l'aide documentaire fournie dont voici un extrait :

Les fonctions d'échantillonnage et de génération de valeurs pseudo-aléatoires sont regroupées dans la bibliothèque `numpy.random`.

```
import numpy.random as rd
```

La fonction `binomial` permet de simuler une variable aléatoire suivant une loi binomiale de paramètres n et p . Elle permet donc également de simuler une variable aléatoire suivant une loi de BERNOULLI de paramètre p en prenant simplement $n = 1$. Cette fonction prend un troisième paramètre optionnel qui correspond, comme pour les fonctions précédentes, au nombre de valeurs à obtenir.


```
In [1]: rd.binomial(10, 0.3, 7)
Out[1]: array([4, 3, 4, 1, 4, 3, 3])

In [2]: rd.binomial(1, 0.6, 20)
Out[2]: array([0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1,
               1, 1, 1, 1, 1])
```

2. Pour un entier k assez grand et des valeurs de p de votre choix, calculez sur une centaine d'essais la proportion de tentatives pour lesquelles le pion atteint la position k exactement. Comparez avec $1/\mathbb{E}(Y_1)$.

Test 5.22 (Roulette - ♥♥)

Une roulette contient 36 cases numérotées de 1 à 36 dont 18 sont rouges et 18 sont noires, plus une case numérotée 0 de couleur verte.

Un joueur qui mise sur la couleur rouge ou noire gagne deux fois sa mise si la couleur choisie sort.

Un joueur qui mise sur un numéro de 1 à 36 gagne 36 fois sa mise si le numéro sort.

Il est interdit de miser sur le zéro.

1. Un joueur mise $a \in$ sur une couleur. Soit C la variable aléatoire correspondant au gain associé. Trouvez la loi de C puis calculez $\mathbb{E}(C)$ et $\sigma(C)$.
2. Un joueur mise $a \in$ sur un numéro. Soit N la variable aléatoire correspondant au gain associé. Trouvez la loi de N puis calculez $\mathbb{E}(N)$ et $\sigma(N)$.
3. Vaut-il mieux miser sur une couleur ou un numéro ?

Test 5.23 (« C'est sans danger... » - ♥♥♥)

Afin d'effectuer des économies dans le budget de l'État, le premier ministre syldave a supprimé la faculté de médecine. Dans le but de promouvoir les liens sociaux, les dentistes sont maintenant choisis parmi les anciens boxeurs aveugles et parkinsoniens. Ils arrachent les dents de leurs patients au hasard.

Suite à quelques plaintes, une étude statistique a été menée dans le cabinet d'un de ces nouveaux dentistes. Les syldaves venant le consulter ont toujours une seule dent de malade parmi les trente-deux qu'ils possèdent encore avant l'intervention des tenailles ou des poings, c'est selon (une étude précédente a en effet montré que les syldaves ne retournent jamais chez leur dentiste après une première consultation).

L'étude a été effectuée sur n patients. On note X la variable aléatoire égale au nombre de dents malades extraites à bon escient.

1. Déterminez la loi de probabilité de la variable aléatoire X . Calculez la probabilité pour qu'aucune dent malade n'ait été extraite.
2. Combien de patients doit-il traiter pour extraire au moins une dent malade avec une probabilité supérieure à 0,9 ?
3. Le dernier patient est prêt à se sacrifier pour la gloire des statistiques syldaves et se laisse arracher les dents une à une tant que la dent malade n'a pas été extraite. On note Y le nombre de dents saines que ce vaillant patriote voit tomber de ses mâchoires endolories.

Calculez la probabilité pour qu'il reparte complètement édenté, puis $\mathbb{E}(Y)$ et $\sigma(Y)$.

Test 5.24 (Dé truqué - ♥)

On considère 6 dés, cinq étant équilibrés. Le dernier est pipé de manière à ce que lorsque l'on lance ce dé, chacun des chiffres apparaît avec une probabilité proportionnelle à ce chiffre.

1. Donner la loi, l'espérance et la variance de la variable aléatoire égale au chiffre donné par le dé truqué lorsqu'on le lance.
2. On effectue n tirages successifs et indépendants d'un dé parmi les six.
Quelle est la loi suivie par la variable aléatoire égale au nombre de fois où on a tiré le dé truqué ? Combien de tirages doit-on effectuer pour que la probabilité d'avoir obtenu le dé truqué parmi ceux tirés soit supérieure ou égale à $1/2$?
3. On effectue n tirages ($n \leq 6$) successifs sans remise d'un dé parmi les six. Quelle est la loi suivie par la variable aléatoire égale au nombre de fois où est tiré le dé truqué ? Combien de tirages doit-on effectuer pour que la probabilité d'avoir obtenu le dé truqué parmi ceux tirés soit supérieure ou égale à $1/2$?

Test 5.25 (File d'attente - ♥♥)

On constitue une file d'attente en attribuant au hasard des numéros d'ordre à n personnes dont vous et Monica B. Quelle est la probabilité que vous soyez distant de p places de Monica (c'est-à-dire séparés par $p - 1$ personnes) ? Quelle est la distance la plus probable entre vous et Monica ?

Test 5.26 (Théorème de KÖNIG-HUYGENS - ♥♥)

Démontrez le théorème de la page 146...

Test 5.27 (Espérance d'une loi géométrique - ♥♥)

Notons S la somme définie par :

$$\forall q \in \mathbb{R}, 0 \leq q < 1 \Rightarrow S = \sum_{n \geq 0} q^n$$

Pour $0 \leq q < 1$, notons $T = \sum_{n \geq 1} nq^{n-1}$ en supposant que cette écriture a un sens.

Calculer $(1 - q)T$ en fonction de S et déduisez-en T puis l'espérance d'une loi géométrique.

Coups de pousse

Coup de pousse pour le test 5.1

Un **octet** est une 8-liste d'éléments de $\{0,1\}$.

Coup de pousse pour le test 5.2

Utiliser le principe multiplicatif : un numéro de téléphone est une 10-liste.

Coup de pousse pour le test 5.3

Une application est une fonction telle que tout élément de l'ensemble de départ E admet une image et une seule dans F . Les informaticiens parlent eux de fonction totale.

Coup de pousse pour le test 5.4

Modéliser un habillage par une 7-liste.

Coup de pousse pour le test 5.5

On note E , T et B l'ensemble des buveurs d'eau, de jus de tomate et de jus de betterave respectivement. Utilisez la formule du crible à répétitions avec $E \cup (T \cup B)$ ainsi que la distributivité de l'intersection sur la réunion.

Coup de pousse pour le test 5.6

Pour les dénombrements avec ordre final imposé, commencer par choisir les éléments de façon non ordonnée.

Coup de pousse pour le test 5.7

Choisir une carte c'est choisir sa hauteur et sa couleur : pensez à des combinaisons.

Coup de pousse pour le test 5.8

Le jeu est mélangé une fois. Numérotons les cartes de 1 à 52. Pour recréer la situation initiale (rouges d'un côté, bleues de l'autre), il faut donc créer un groupe de 26 cartes parmi les 52.

Coup de pousse pour le test 5.9

Souvenez-vous que $n!$ est le nombre de permutations d'une n -liste.

Coup de pousse pour le test 5.10

On peut modéliser un trajet comme un mot constitué de r (pour rue) et de a (pour avenue) : il s'agit de chercher les anagrammes d'un tel mot...

Coup de pousse pour le test 5.11

La démonstration de la formule associée pour les cardinaux a déjà été faite à l'exercice 5.5.

Coup de pousse pour le test 5.12

Relisez la partie du cours concernant les axiomes de KOLMOGOROV.

Étudier la limite quand n tend vers l'infini de $\sum_{i=1}^n q^i$ lorsque $|q| < 1$.

Coup de pousse pour le test 5.13

Il **semble** y avoir 6 façons d'obtenir 9 ou 10. Pour modéliser la situation, numérotez les dés. Un lancer correspond donc à un triplet (ordonné !) de chiffres entre 1 et 6.

Coup de pousse pour le test 5.14

La formule de BAYES :

$$\mathbb{P}_{\text{effet}}(\text{cause}) = \frac{\mathbb{P}_{\text{cause}}(\text{effet}) \times \mathbb{P}(\text{cause})}{\mathbb{P}(\text{effet})}$$

Coup de pousse pour le test 5.15

On peut avoir deux idées contradictoires :

- le présentateur ayant ouvert une porte, le candidat a une chance sur deux de tomber sur la voiture ;
- au départ, le candidat a une probabilité de $\frac{1}{3}$ de tomber sur la voiture. Ensuite, s'il garde son choix initial, il a toujours la même probabilité $\frac{1}{3}$ de gagner donc s'il change d'avis, la probabilité qu'il gagne est donc $1 - \frac{1}{3} = \frac{2}{3}$.

Qui a raison ? Commencez par dresser un arbre et à considérer que les choix des portes sont équiprobables.

Coup de pousse pour le test 5.16

Un arbre permet de visualiser la situation.

Il faut savoir exprimer la somme des termes d'une suite géométrique.

Coup de pousse pour le test 5.17

Rappel : on dit que la v.a.r. finie X suit la **loi binomiale de paramètres n et p** si $n \in \mathbb{N} \setminus \{0\}$, $p \in]0; 1[$, $X(\Omega) = \llbracket 0, n \rrbracket$ et si l'expérience est une suite de n expériences de BERNOULLI de paramètre p menées indépendamment.

Coup de pousse pour le test 5.18

Étudier l'indépendance des répétitions pour distinguer loi binomiale et loi hypergéométrique.

Coup de pousse pour le test 5.19

Pensez au poker...

Coup de pousse pour le test 5.20

On tire une boule tant que les n numéros n'ont pas été tirés.

Coup de pousse pour le test 5.21

1. On obtient Y_1 avec `1+rd.binomial(1, p, 1)[0]`. Comme `binomial` renvoie un tableau numpy, le crochet permet d'en extraire le premier et seul élément.

2. 1 pour succès et 0 pour échec : la somme des valeurs obtenues donne donc le nombre de succès.

Coup de pouce pour le test 5.22

La variable aléatoire C ne prend que deux valeurs : a et $-a$. Étudiez sa loi de probabilité.

Coup de pouce pour le test 5.23

Pour chaque patient, l'extraction d'une dent est une épreuve de BERNOULLI de paramètre $\frac{1}{32}$.

Coup de pouce pour le test 5.24

1. Noter p la probabilité d'obtenir « 1 » avec le dé pipé.
2. Remarquer que les tirages sont indépendants.
3. Remarquer que les tirages ne sont pas indépendants.

Coup de pouce pour le test 5.25

Commencez par compter le nombre de files telles que vous soyez à la distance p de Monica en ayant un numéro plus petit qu'elle et souvenez-vous des sommes usuelles : somme des n premiers entiers, des n premiers carrés d'entiers...

Coup de pouce pour le test 5.26

Souvenez-vous de $(a + b)^2 = \dots$ et de la linéarité de l'espérance.

Coup de pouce pour le test 5.27

Nous avons déjà démontré que :

$$\forall q \in \mathbb{R}, |q| < 1 \Rightarrow S = \sum_{n \geq 0} q^n = \frac{1}{1-q}$$

Solutions et commentaires

Solution du test 5.1

Soit U l'ensemble des octets commençant par 1 et Z l'ensemble des octets se terminant par 00. D'après la formule du crible :

$$|U \cup Z| = |U| + |Z| - |U \cap Z|$$

$|U| = 2^7$: un bit est fixé et il reste 2 possibilités pour chacun des 7 bits restant.

$|Z| = 2^6$: deux bits sont fixés et il reste 2 possibilités pour chacun des 6 bits restant.

Un élément de $U \cap Z$ est un octet commençant par 1 et finissant par 00 : il reste 2 possibilités pour chacun des 5 bits restant. On a donc $|U \cap Z| = 2^5$.

Finalement :

$$|U \cup Z| = 2^7 + 2^6 - 2^5 = 2^5(4 + 2 - 1) = 5 \times 32 = 160$$

Solution du test 5.2

1. Il y a 5 zones « fixes ». Il y a donc $5 \times 8 \times 10^7$ soit 400 millions de numéros possibles : tout va bien.

Il y a 2 zones « mobiles ». Il y a donc $2 \times 8 \times 10^7$ soit 160 millions de numéros possibles. En ne créant pas de numéros commençant par 07, on aurait pu risquer la surchauffe : 80 millions de numéros pour 69 millions d'abonnés.

2. Appelons z le nombre minimum de zones nécessaires. Cela permet de créer $z \times 2 \times 10^7$ soit $20z$ millions de numéros. On doit donc avoir $20z \geq 109$: il faut au moins 6 zones distinctes.

Solution du test 5.3

On peut caractériser une application par une n -liste (on ordonne par exemple les éléments de E dans l'ordre croissant si c'est possible) formée de n éléments à choisir parmi les p images possibles.

Il y a donc p^n listes différentes caractérisant chacune des applications.

Il y a donc $|F|^{|E|}$ applications de E dans F .

Solution du test 5.4

1. Il y a 10 possibilités pour choisir la robe de chacune des 7 Barbie. Il y a donc autant d'habillages possibles que de sept-listes d'un ensemble à 10 éléments, c'est-à-dire 10^7 .

2. Le nombre d'habillages cherché est le nombre de sept-listes d'éléments deux à deux distincts d'un ensemble à 10 éléments, c'est-à-dire le nombre d'arrangements de 7 éléments de l'ensemble des robes :

$$A_{10}^7 = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 = 604\,800$$

3. Cette fois il y a 7 robes. Il y a donc $7^7 = 823\,543$ habillages possibles et $A_7^7 = 7! = 5\,040$ habillages de sorte que chaque Barbie ait une robe différente des autres (cela revient en fait à permuter les robes).

4. L'ensemble cherché est le complémentaire de l'ensemble des habillages de sorte que chaque Barbie ait une robe différente des autres. Le nombre cherché est donc :

$$10^7 - A_{10}^7 = 9\,395\,200$$

5. Pour un habillage ayant exactement deux Barbie portant la même robe, il faut :

- choisir deux robes parmi 7 (sans ordre car elles porteront la même robe) : il y a $\binom{7}{2} = 21$ choix possibles ;
- ensuite choisir une robe parmi 10 pour ces deux Barbie : il y a 10 choix possibles ;
- puis choisir une robe différente parmi 9 pour chacune des 5 Barbie restantes : il y a A_9^5 choix possibles.

Le nombre cherché est donc :

$$21 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 = 3\,175\,200$$

Solution du test 5.5

Notons x le nombre d'étudiants ayant bu un verre de chaque type de boisson.

$$\begin{aligned} |E \cup (B \cup T)| &= |E| + |B \cup T| - |E \cap (B \cup T)| \\ &= |E| + |B| + |T| - |B \cap T| - |(E \cap B) \cup (E \cap T)| \\ &= |E| + |B| + |T| - |B \cap T| - (|E \cap B| + |E \cap T| - |E \cap B \cap T|) \\ &= |E| + |B| + |T| - |B \cap T| - |E \cap B| - |E \cap T| + |E \cap B \cap T| \\ &= 9 + 14 + 25 - 10 - 4 - 7 + x \\ &= 27 + x \end{aligned}$$

Or $E \cap B \cap T \subseteq E \cap B$ donc $|E \cap B \cap T| = x \leq |E \cap B| = 4$.

De plus $|E \cap \overline{(T \cup B)}| = x - 2 \geq 0$ donc $x \geq 0$.

Enfin, le nombre de participants est pair : il vaut donc 30.

Solution du test 5.6

1. D'après le cours, il y a $\binom{n}{k}$ parties de E formées de k éléments.

2. D'après le cours, il y a n^k k -uplets de E .

3. D'après le cours, il y a $A_n^k = \frac{n!}{(n-k)!}$ k -uplets d'éléments de E deux à deux distincts.

4. On commence par choisir les k éléments deux à deux distincts, sans tenir compte de l'ordre des choix : $\binom{n}{k}$ choix possibles.

Ensuite on choisit le plus petit de ces éléments : 1 seul choix possible (car les k éléments sont 2 à 2 distincts), et on le place en première position du k -uplet. Puis on choisit le plus grand de ces éléments : 1 seul choix possible, et on le place en dernière position du k -uplet. Dernière étape : on place les $k-2$ éléments restants dans les $k-2$ positions restantes du k -uplet : $(k-2)!$ choix possibles.

Donc au total, il y a $\binom{n}{k} \times (k-2)! = \frac{n!}{k(k-1)(n-k)!}$ k -uplets d'éléments de E deux à deux distincts, tels que le premier élément est le plus petit et le dernier le plus grand.

5. On commence là aussi par choisir les k éléments deux à deux distincts, sans tenir compte de l'ordre des choix : $\binom{n}{k}$ choix possibles.

Ensuite on les met dans l'ordre strictement croissant : 1 seule possibilité.

Donc au total, il y a $\binom{n}{k} \times 1 = \binom{n}{k}$ k -uplets d'éléments de E dans l'ordre strictement croissant.

Solution du test 5.7

On modélise la distribution d'une main de 5 cartes par le choix d'une combinaison de 5 cartes parmi 32 (car l'ordre dans lequel les cartes sont distribuées n'a pas d'influence sur le jeu du joueur). L'univers Ω est donc l'ensemble des combinaisons de 5 cartes parmi 32, et $|\Omega| = \binom{32}{5} = 201\,376$.

Remarquons ensuite que dans un jeu de 32 cartes, il y a 4 couleurs (trèfle, carreau, coeur et pique) et 8 hauteurs dans chaque couleur (7, 8, 9, 10, valet, dame, roi, as).

Toutes les figures considérées dans cet exercice dépendent de la hauteur des 5 cartes de la main du joueur. Pour les dénombrements, nous allons donc commencer par choisir la hauteur des 5 cartes, puis leur couleur.

1. On note A l'évènement : « obtenir une seule paire ».

$$\begin{array}{ccccccc}
 \binom{8}{1} & \times & \binom{4}{2} & \times & \binom{7}{3} & \times & \binom{4}{1}^3 \\
 \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 \text{une} & & 2 \text{ cartes} & & 3 \text{ hauteurs} & & \text{une carte} \\
 \text{hauteur} & & \text{dans cette} & & \text{dans les 7} & & \text{dans chaque} \\
 \text{parmi 8 cartes} & & \text{hauteur} & & \text{restantes} & & \text{hauteur}
 \end{array}$$

Remarquons que le choix des deux cartes de la paire se fait de $\binom{4}{2}$ façons différentes et non pas de $\binom{4}{1}^2$ façons, car obtenir « roi de coeur, roi de trèfle » donne la même paire qu'obtenir « roi de trèfle, roi de coeur ». Il ne faut donc le compter qu'une seule fois.

On a donc :

$$|A| = \binom{8}{1} \binom{4}{2} \binom{7}{3} \binom{4}{1}^3 = 107\,520$$

maines avec une paire possibles.

2. On note B l'évènement : « obtenir deux paires ».

On adopte la même méthode. Il faut choisir la hauteur de chacune des 2 paires : $\binom{8}{2}$ possibilités (car « paire de roi, paire de 7 » donne la même main que « paire de 7, paire de roi ») ; la hauteur de la cinquième carte : $\binom{6}{1}$ possibilités ; la couleur des deux cartes de la première paire : $\binom{4}{2}$ possibilités ; la couleur des deux cartes de la seconde paire : $\binom{4}{2}$ possibilités ; la couleur de la cinquième carte : $\binom{4}{1}$ possibilités.

Au total :

$$|B| = \binom{8}{2} \times \binom{4}{2}^2 \times \binom{6}{1} \times \binom{4}{1} = 24\,192$$

maines avec deux paires.

3. On note C l'évènement : « obtenir un brelan ».

Il faut choisir la hauteur des trois premières cartes : $\binom{8}{1}$ possibilités ; les hauteurs des deux autres cartes : $\binom{7}{2}$ possibilités ; la couleur des trois premières cartes : $\binom{4}{3}$ possibilités ; la couleur de la quatrième carte : $\binom{4}{1}$ possibilités ; la couleur de la cinquième carte : $\binom{4}{1}$ possibilités.

Au total :

$$|C| = \binom{8}{1} \times \binom{4}{3} \times \binom{7}{2} \times \binom{4}{1}^2 = 10\,752$$

4. On note D l'évènement : « obtenir un carré ».

Il faut choisir la hauteur des quatre premières cartes : $\binom{8}{1}$ possibilités ; la hauteur de la cinquième carte : $\binom{7}{1}$ possibilités, la couleur des quatre premières cartes : $\binom{4}{4}$ possibilité ; la couleur de la cinquième carte : $\binom{4}{1}$ possibilités.

Au total :

$$|D| = \binom{8}{1} \times \binom{4}{4} \times \binom{7}{1} \times \binom{4}{1} = 224$$

5. On note E l'évènement : « obtenir un full ».

Il faut choisir la hauteur des trois premières cartes : $\binom{8}{1}$ possibilités ; la hauteur des deux autres cartes : $\binom{7}{1}$ possibilités ; la couleur des trois premières cartes : $\binom{4}{3}$ possibilités ; la couleur des deux cartes de la paire : $\binom{4}{2}$ possibilités.

Au total :

$$|E| = \binom{8}{1} \times \binom{4}{3} \times \binom{7}{1} \times \binom{4}{2} = 1\,344$$

Solution du test 5.8

Il y a 52! façons d'ordonner les 52 cartes.

Un mélange correspond à la création d'un groupe de 26 cartes parmi les 52 (les 26 rouges par exemple).

Il y a donc $\binom{52}{26}$ ordres possibles après un mélange, donc $\left(\binom{52}{26}\right)^4$ ordres possibles après quatre mélanges.

On doit donc comparer 52! et $\left(\binom{52}{26}\right)^4$. On peut utiliser **Python** :

```
In [81]: float(sy.factorial(52) / (sy.binomial(52, 26) ** 4)
Out [81]: 1333543042.3319275
```

Le rapport est d'environ un milliard : il existe donc de nombreux ordres non atteignables par quatre mélanges successifs.

Mais...

```
In [82]: float(sy.factorial(52) / (sy.binomial(52, 26) ** 5)
Out [82]: 2.6890365125182324e-06
```

Avec cinq mélanges successifs, on est en dessous de 1 : cinq mélanges successifs permettent donc d'obtenir un ordre aléatoire.

Solution du test 5.9

1. $\binom{n}{p} = \frac{n!}{p!(n-p)!} = C_n^{n, (n-p)}.$

2. Cela permet de compter les permutations d'une n -liste contenant des répétitions : k_1 éléments e_1 , k_2 éléments e_2 , etc. Le nombre d'anagrammes du mot « suissesess » vaut donc $C_{10}^{6,1,1,2} = C_{10}^{6,2} = \frac{10!}{6!2!} = 2\,520$.

3. Le coefficient de x^2y^5z dans $(x+y+z)^8$ vaut $C_8^{2,5} = \frac{8!}{2!5!} = 168$.

Solution du test 5.10

Un trajet peut être modélisé par un « mot » comme rrrrrrrraaaaaaaa contenant 7 r et 8 a .

Comme pour les Suissesess, on obtient :

$$n = C_{15}^{7,8} = \frac{15!}{8! \times 7!} = 6\,435$$

6 435 jours, c'est environ 17 ans et demi : le capitaine a donc au maximum 35 ans.

Solution du test 5.11

Comme pour les cardinaux :

$$\begin{aligned} \mathbb{P}(A \cup (B \cup C)) &= \mathbb{P}(A) + \mathbb{P}(B \cup C) - \mathbb{P}(A \cap (B \cup C)) \\ &= \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C) - \mathbb{P}(B \cap C) - \mathbb{P}((A \cap B) \cup (A \cap C)) \\ &= \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C) - \mathbb{P}(B \cap C) - (\mathbb{P}(A \cap B) + \mathbb{P}(A \cap C) - \mathbb{P}(A \cap B \cap C)) \\ &= \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(C) - \mathbb{P}(B \cap C) - \mathbb{P}(A \cap B) - \mathbb{P}(A \cap C) + \mathbb{P}(A \cap B \cap C) \end{aligned}$$

Solution du test 5.12

Ω est l'union disjointe des ω_i donc

$$\mathbb{P}(\Omega) = \mathbb{P}\left(\bigcup_{i \geq 1} \{\omega_i\}\right) = \sum_{i \geq 1} \mathbb{P}(\omega_i)$$

d'après le troisième axiome de KOLMOGOROV.

Or $\mathbb{P}(\omega_i) = \frac{\lambda}{37^i}$ donc :

$$\mathbb{P}(\Omega) = \sum_{i \geq 1} \frac{\lambda}{37^i} = \lambda \sum_{i \geq 1} \left(\frac{1}{37}\right)^i$$

Petit rappel : la somme des termes d'une suite géométrique vaut :

$$\text{premier terme} \times \frac{1 - \text{raison}^{\text{nombre de termes}}}{1 - \text{raison}}$$

Ici :

$$\sum_{i=1}^n \left(\frac{1}{37}\right)^i = \frac{1}{37} \times \frac{1 - (1/37)^n}{1 - 1/37} = \frac{1}{36} (1 - (1/37)^n)$$

Or $|1/37| < 1$ donc $\lim_{n \rightarrow +\infty} (1/37)^n = 0$. On en déduit que :

$$\lim_{n \rightarrow +\infty} \sum_{i=1}^n \left(\frac{1}{37}\right)^i = \frac{1}{36}$$

puis que $\mathbb{P}(\Omega) = \frac{\lambda}{36}$. Or, le deuxième axiome de monsieur K. impose que $\mathbb{P}(\Omega) = 1$. Nous en déduisons que nécessairement $\lambda = 36$.

Il reste à vérifier que cette valeur suffit à faire de $(\Omega, \mathfrak{P}(\Omega), \mathbb{P})$ un espace probabilisé. Le deuxième axiome est déjà vérifié.

Toute partie A de Ω est l'union disjointe de ω_i avec $i \in I$ une famille incluse dans \mathbb{N}^* donc :

$$0 \leq \mathbb{P}(A) = \sum_{i \in I} \frac{36}{37^i} \leq \sum_{i \in \mathbb{N}^*} \frac{36}{37^i} = 1$$

ce qui assure que le premier axiome est vérifié.

Soit B une partie de Ω telle que $A \cap B = \emptyset$. B est alors l'union disjointe de ω_i avec $i \in J$ une famille incluse dans \mathbb{N}^* et telle que $I \cap J = \emptyset$ et $I \cup J \subseteq \mathbb{N}$.

$$\mathbb{P}(A \cup B) = \sum_{i \in I \cup J} \frac{36}{37^i} = \sum_{i \in I} \frac{36}{37^i} + \sum_{i \in J} \frac{36}{37^i} = \mathbb{P}(A) + \mathbb{P}(B)$$

ce qui assure que le troisième axiome est vérifié.

Solution du test 5.13

1. Simulons à l'aide de **Python** le lancement de trois dés.

La commande `randint(a,b)` renvoie un nombre entier aléatoirement choisi entre *a* et *b* (*b* non compris à partir de **Python 3** !) selon la loi uniforme.

Si on ajoute l'option `size`, on peut obtenir une liste de tirages.

On utilise ensuite `Counter(liste)` qui crée le dictionnaire des effectifs de la liste donnée en argument.

On effectue par exemple 100 000 **simulations** du lancer de trois dés et on enregistre la somme des numéros des faces dans la liste *s* :

```
from numpy.random import randint
from collections import Counter
s = [sum(randint(1,7,size=3)) for k in range(100000)]
c = Counter(s)
```

On obtient :

```
Counter({11: 12554, 10: 12319, 12: 11737, 9: 11400, 8: 9636, 13: 9564, 7: 7052, 14: 6836, 15: 4708, 6: 4703, 5: 2825, 16: 2750, 17: 1538, 4: 1486, 18: 466, 3: 426})
```

On peut visualiser le résultat sous la forme d'un diagramme en bâtons sur son navigateur libre préféré :

```
from pygal import *
toscane = HorizontalBar()
toscane.title = 'Effectifs des sorties du problème du duc de Toscane'
for v in c:
    toscane.add(str(v), c[v])
toscane.render_to_file('toscane.svg')
```

Ensuite, on ouvre le fichier depuis ipython avec par exemple la commande :

```
In [1] : !firefox ./toscane.svg
```

On observe que « 10 » sort plus que « 9 » ($12\,455 > 11\,416$). Et expérimentalement, la probabilité d'obtenir « 10 » serait de $\frac{12\,455}{100\,000} \approx 0,125$, et la probabilité d'obtenir « 9 » serait de $\frac{11\,416}{100\,000} \approx 0,114$.

Pourtant, il y a six façons d'obtenir « 9 » :

- {6,2,1}
- {5,3,1}
- {5,2,2}
- {4,4,1}
- {4,3,2}
- {3,3,3}

et six façons d'obtenir « 10 » :

- {6,3,1}
- {6,2,2}
- {5,4,1}
- {5,3,2}
- {4,4,2}
- {4,3,3}

Pour modéliser la situation, numérotons les dés. Un lancer correspond donc à un triplet de chiffres entre 1 et 6 et l'univers Ω est : $\Omega = \llbracket 1,6 \rrbracket^2$.

Au total, il y a donc 6^3 lancers possibles.

Il y a 3! triplets donnant les chiffres {6,2,1}, et autant pour les chiffres {5,3,1} et {4,3,2}.

Pour {5,2,2}, le chiffre 2 apparaît deux fois : il y a donc $\frac{3!}{2!}$ triplets donnant les chiffres {5,2,2}, et autant pour les chiffres {4,4,1}.

Enfin il y a un unique triplet donnant les chiffres {3,3,3}.

Au total, on a donc $6 + 6 + 6 + 3 + 3 + 1 = 25$ triplets donnant un total de « 9 ». Comme tous les triplets sont équiprobables, la probabilité d'obtenir un total « 9 » est $\frac{25}{6^3} \approx 0,1157$ (remarquons qu'on retrouve à peu près la valeur expérimentale).

Un calcul similaire permet d'obtenir que la probabilité d'avoir un total de « 10 » est $\frac{6+6+6+3+3+3}{6^3} = \frac{27}{6^3} \approx 0,125$

(remarquons encore une fois qu'on retrouve à peu près la valeur expérimentale).

Comme $27 > 25$, on a donc bien démontré qu'on obtient plus souvent un total de « 10 » qu'un total de « 9 ».

2. On crée 100 000 **simulations** de 4 lancers de dés. Si on trouve au moins un 6 dans cette série on ajoute 1 à notre compteur, sinon rien.

```
cpt = 0
for i in range(100000):
    if 6 in randint(1,7,size = 4) :
        cpt += 1
```

On obtient le résultat suivant :

```
In [21]: cpt
Out [21]: 51866
```

On aurait pu écrire de manière plus compacte :

```
sum([6 in randint(1, 7, size=4) for _ in range(n)])
```

car le booléen True est un synonyme de 1... une *pythonerie* en passant.

On a donc une probabilité expérimentale de 0,51866, qui est plus grande que la valeur d'équilibre 0,5. Le jeu semble donc être favorable. Pour le vérifier, calculons la probabilité de l'évènement contraire de « obtenir au moins un six ». Il s'agit de l'évènement « n'obtenir aucun six ».

Par indépendance des résultats obtenus à chacun des 4 lancers, la probabilité de « n'obtenir aucun six » est $\left(\frac{5}{6}\right)^4$ et donc la probabilité « d'obtenir au moins un six » est :

$$1 - \left(\frac{5}{6}\right)^4 \approx 0,518$$

(ce qui correspond à notre simulation).

Le premier jeu du chevalier de MÉRÉ est donc favorable au joueur.

Adaptons la méthode précédente pour le deuxième jeu :

```
cpt = 0
for i in range(100000):
    if 12 in [randint(1,7) + randint(1,7) for k in range(24)]:
        cpt += 1
```

On obtient cette fois :

```
In [22]: cpt
Out [22]: 49075
```

On a donc une probabilité expérimentale de 0,49075, qui est plus petite que la valeur d'équilibre 0,5. Cette fois, le jeu ne semble pas être favorable. Pour le vérifier, calculons la probabilité de l'évènement contraire de « obtenir au moins un double six ». Il s'agit de l'évènement « n'obtenir aucun double six ».

Par indépendance des résultats obtenus à chacun des 24 lancers, la probabilité de « n'obtenir aucun double six » est $\left(\frac{35}{36}\right)^{24}$ donc la probabilité « d'obtenir au moins un double six » est :

$$1 - \left(\frac{35}{36}\right)^{24} \approx 0,491$$

(ce qui correspond à notre simulation).

Le second jeu du chevalier de MÉRÉ est donc défavorable au joueur.

Cependant, les probabilités sont tellement proches de 0,5 qu'une approche expérimentale « manuelle » comme celle faite par le chevalier ne peut permettre de conclure.

Solution du test 5.14

Appelons M l'évènement : « le Syldave enlevé est un homme », F l'évènement « le Syldave enlevé est une femme » et M « le Syldave enlevé a une photo de Monica B. dans son portefeuille ».

On cherche à calculer $\mathbb{P}_B(F)$. On applique la formule de BAYES :

$$\mathbb{P}_B(F) = \frac{\mathbb{P}_F(B) \times \mathbb{P}(F)}{\mathbb{P}(B)}$$

puis la formule des probabilités totales, sachant que F et M réalisent une partition de l'univers :

$$\mathbb{P}(B) = \mathbb{P}_H(B) \times \mathbb{P}(H) + \mathbb{P}_F(B) \times \mathbb{P}(F) = 0,6 \times 0,7 + 0,25 \times 0,3 = 0,495$$

On en déduit que :

$$\mathbb{P}_B(F) = \frac{0,25 \times 0,3}{0,495} = \frac{5}{33}$$

Solution du test 5.15

Simulons la situation avec **Python** :

```
def monty(n):
    gagne_sans_changer = 0
    gagne_en_changeant = 0
    for i in range(n):
        voiture = randint(0,3)
        choix = randint(0,3)
        if choix == voiture:
            ouverte = (voiture + randint(1,3)) % 3
        else:
            ouverte = (0 + 1 + 2) - choix - voiture
        changement = (0 + 1 + 2) - choix - ouverte
        if choix == voiture:
            gagne_sans_changer += 1
        if changement == voiture:
            gagne_en_changeant += 1
    return f"Gagne en changeant : {100. * gagne_en_changeant/n}%\nGagne sans changer : {100. * gagne_sans_changer/n}%"
```

On obtient alors, pour 10 000 000 de simulations :

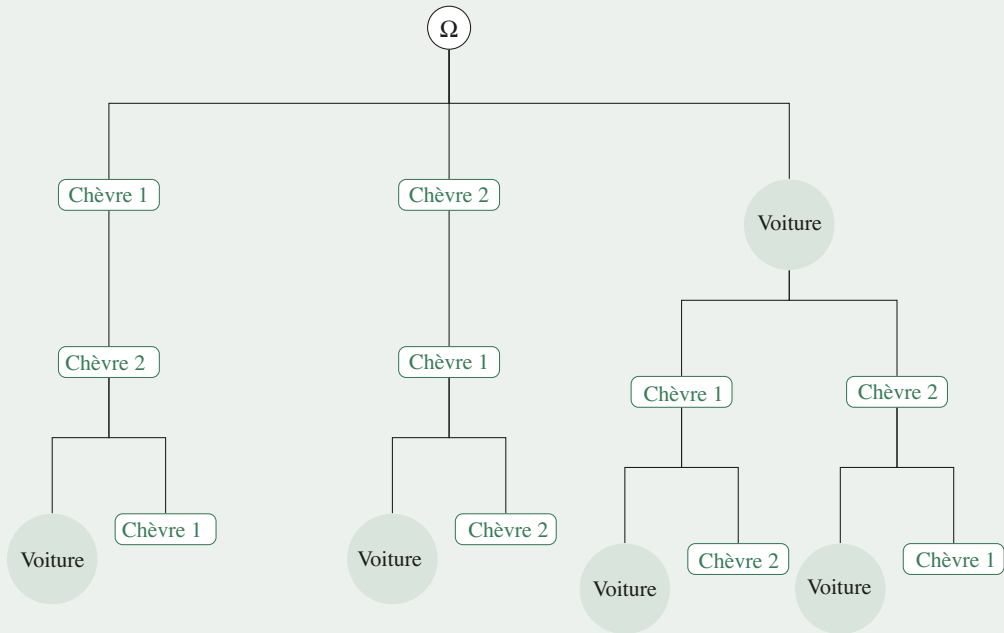
```
In [15]: monty(10000000)
Out[15]: 'Gagne en changeant : 66.67858% Gagne sans changer : 33.32142%'
```

La deuxième option semble la bonne... il reste à le démontrer !

Dressons un arbre en trois étapes (après avoir numérotées nos deux chèvres) :

- le candidat choisit d'abord une porte de manière équiprobable ;
- le présentateur, sachant où se trouve la voiture, choisit une porte cachant une chèvre ;
- le candidat prend ensuite la décision de garder ou de changer son choix initial.

Le premier nœud correspond au choix initial du joueur, le second à la chèvre découverte par le présentateur, et le troisième au choix final du joueur.



Notons A l'évènement « le joueur gagne en conservant son choix initial ». Il n'y a que deux déroulements de l'expérience qui réalisent A :

- le joueur choisit la porte cachant la voiture (avec probabilité $\frac{1}{3}$), le présentateur choisit la porte cachant la chèvre 1 (avec probabilité $\frac{1}{2}$), le joueur choisit de garder son choix initial (avec probabilité $\frac{1}{2}$) ;
- le joueur choisit la porte cachant la voiture (avec probabilité $\frac{1}{3}$), le présentateur choisit la porte cachant la chèvre 2 (avec probabilité $\frac{1}{2}$), le joueur choisit de garder son choix initial (avec probabilité $\frac{1}{2}$).

On a donc :

$$\mathbb{P}(A) = \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times \frac{1}{2} = \frac{1}{3}$$

De même, notons B l'évènement « le joueur gagne en changeant son choix initial ». Il n'y a que deux déroulements de l'expérience qui réalisent B :

- le joueur choisit la porte cachant la chèvre 1 (avec probabilité $\frac{1}{3}$), le présentateur choisit la porte cachant la chèvre 2 (avec probabilité 1), le joueur choisit de changer son choix initial (avec probabilité $\frac{1}{2}$) ;
- le joueur choisit la porte cachant la chèvre 2 (avec probabilité $\frac{1}{3}$), le présentateur choisit la porte cachant la chèvre 1 (avec probabilité $\frac{1}{2}$), le joueur choisit de changer son choix initial (avec probabilité $\frac{1}{2}$).

On a donc :

$$\mathbb{P}(B) = \frac{1}{3} \times 1 + \frac{1}{3} \times 1 = \frac{2}{3}$$

Il vaut donc mieux pour le candidat changer son choix initial.

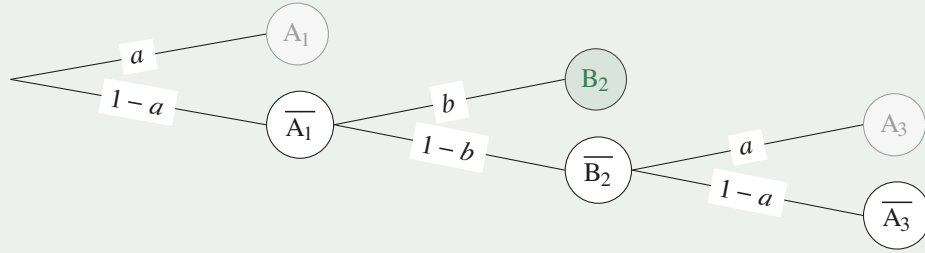
Pour avoir une idée de la controverse passée et présente autour de ce problème voir :

http://fr.wikipedia.org/wiki/Problème_de_Monty_Hall

Nous avons ici supposé que le joueur garde/change son choix initial de manière équiprobable, ce qui peut être discuté...

Solution du test 5.16

1. Dressons un arbre :



Remarquons que les tirs ne sont pas indépendants, car ils n'ont lieu que si personne n'a gagné au tour précédent...

D'après la formule des probabilités composées :

- $\mathbb{P}(A_1) = a$
- $\mathbb{P}(B_2) = \mathbb{P}(B_2 \cap \overline{A_1}) = \mathbb{P}(\overline{A_1})\mathbb{P}_{\overline{A_1}}(B_2) = (1-a)b$
- $\mathbb{P}(A_3) = \mathbb{P}(A_3 \cap \overline{B_2} \cap \overline{A_1}) = \mathbb{P}(\overline{A_1})\mathbb{P}_{\overline{A_1}}(\overline{B_2})\mathbb{P}_{\overline{B_2} \cap \overline{A_1}}(A_3) = (1-a)(1-b)a$

De manière générale, pour tout entier naturel n non nul :

- $\mathbb{P}(A_{2n-1}) = (1-a)^{n-1}(1-b)^{n-1}a$
- $\mathbb{P}(B_{2n}) = (1-a)^n(1-b)^{n-1}b$

2. C_n est une réunion d'évènements bien sûr deux à deux incompatibles

$$C_n = A_1 \cup A_3 \cup \dots \cup A_{2n-1}$$

On en déduit que, pour tout entier naturel n non nul :

$$\begin{aligned} \mathbb{P}(C_n) &= \sum_{k=1}^n (1-a)^{k-1}(1-b)^{k-1}a \\ &= a \sum_{k=0}^{n-1} (1-a)^k(1-b)^k \\ &= a \frac{1 - (1-a)^n(1-b)^n}{1 - (1-a)(1-b)} \end{aligned}$$

car $(1-a)(1-b) \neq 1$.

On obtient de même que, pour tout entier naturel n non nul :

$$\begin{aligned} \mathbb{P}(D_n) &= \sum_{k=1}^n (1-a)^k(1-b)^{k-1}b \\ &= \frac{b}{1-b} \sum_{k=1}^n (1-a)^k(1-b)^k \\ &= b(1-a) \times \frac{1 - (1-a)^n(1-b)^n}{1 - (1-a)(1-b)} \end{aligned}$$

3. Comme $|1-a| < 1$ et $|1-b| < 1$ on obtient :

$$\begin{aligned} \alpha &= \lim_{n \rightarrow +\infty} \mathbb{P}(C_n) = \frac{a}{1 - (1-a)(1-b)} \\ \beta &= \lim_{n \rightarrow +\infty} \mathbb{P}(D_n) = \frac{b(1-a)}{1 - (1-a)(1-b)} \end{aligned}$$

On vérifie que $\alpha + \beta = \frac{a+b(1-a)}{1-(1-a)(1-b)} = \frac{a+b-ab}{a+b-ab} = 1$.

On peut interpréter ce résultat de la façon suivante : au bout d'un très grand nombre de tours, il y a toujours un des deux joueurs qui gagne...

Solution du test 5.17

1. Le tirage est sans remise donc le paramètre des expériences de BERNOULLI change : X ne suit pas une loi binomiale.
2. Cette fois $X \rightsquigarrow \mathcal{B}(3, 7/20)$ et $X(\Omega) = \{0, 1, 2, 3\}$.
3. Lancer 10 dés revient à lancer 10 fois le même dé (enfin, pas tout à fait mais bon...). $X \rightsquigarrow \mathcal{B}(10, 1/6)$ et $X(\Omega) = \llbracket 0, 10 \rrbracket$.
4. Le montage est en série : si une lampe ne fonctionne pas, les suivantes ne fonctionneront pas non plus. Les expériences ne sont donc pas indépendantes.
5. Cette fois $X \rightsquigarrow \mathcal{B}(32, 3/100)$ et $X(\Omega) = \llbracket 0, 32 \rrbracket$.

Solution du test 5.18

1. Toutes les cartes peuvent être choisies de manière équiprobable donc chacune des huit hauteurs également. On en déduit que :

- $X \rightsquigarrow \mathcal{U}(\llbracket 1, 8 \rrbracket)$;

- $\mathbb{E}(X) = \frac{8+1}{2} = \frac{9}{2}$.

2. On répète dix fois dans les mêmes conditions la même expérience qui n'a que deux issues contraires, et ceci de manière indépendante. On en déduit que :

- $X \rightsquigarrow \mathcal{B}\left(10, \frac{1}{3}\right)$;

- $\mathbb{E}(X) = 10 \times \frac{1}{3} = \frac{10}{3}$.

3. On effectue 10 tirages successifs sans remise parmi 45 soldats. La proportion de Troyens est $\frac{30}{45} = \frac{2}{3}$. On en déduit que :

- $X \rightsquigarrow \mathcal{H}\left(45, 10, \frac{2}{3}\right)$;

- $\mathbb{E}(X) = 10 \times \frac{2}{3} = \frac{20}{3}$.

4. Toutes les cartes peuvent être choisies de manière équiprobable. On en déduit que :

- $X \rightsquigarrow \mathcal{U}(\llbracket 1, 32 \rrbracket)$;

- $\mathbb{E}(X) = \frac{32+1}{2} = \frac{33}{2}$.

5. Les 6 sorties possibles sont équiprobables. On en déduit que :

- $X \rightsquigarrow \mathcal{U}(\llbracket 1, 6 \rrbracket)$;

- $\mathbb{E}(X) = \frac{6+1}{2} = \frac{7}{2}$.

6. On effectue 5 tirages successifs sans remise parmi 19 objets. La proportion de pièces est $\frac{12}{19}$. On en déduit que :

- $X \rightsquigarrow \mathcal{H}\left(19, 5, \frac{12}{19}\right)$;

- $\mathbb{E}(X) = 5 \times \frac{12}{19} = \frac{60}{19}$.

7. On répète 3 000 fois dans les mêmes conditions et de manière indépendante la même expérience qui n'a que deux issues contraires. On en déduit que :

- $X \rightsquigarrow \mathcal{B}\left(3\,000, \frac{2}{3}\right)$;

- $\mathbb{E}(X) = 3\,000 \times \frac{2}{3} = 2\,000.$

8. On effectue 7 tirages successifs sans remise d'une clé parmi. Il n'y a qu'une clé sur les 7 qui ouvre la porte. On en déduit que :

- $X \rightsquigarrow \mathcal{U}(\llbracket 1, 7 \rrbracket)$;
- $\mathbb{E}(X) = \frac{7+1}{2} = 4.$

Solution du test 5.19

Tirer simultanément n boules et compter les blanches, c'est un peu comme jouer au poker avec n cartes n'ayant que deux couleurs : Np cartes blanches et $N(1-p)$ cartes non blanches.

Il y a $\binom{N}{n}$ tirages possibles de n cartes parmi N . Il faut choisir k blanches parmi les Np blanches et $(n-k)$ cartes parmi les $N(1-p)$ non blanches, l'ordre n'étant pas important, on utilise des combinaisons :

$$\mathbb{P}(\{X = k\}) = \frac{\binom{Np}{k} \binom{N(1-p)}{n-k}}{\binom{N}{n}}$$

Solution du test 5.20

A contient les numéros obtenus à chaque tirage. On ne rajoute un élément dans A que s'il n'est pas déjà présent.

Ligne 6 : tant que A ne contient pas n éléments ;

Ligne 7 : on tire un entier de 1 à n selon la loi uniforme ;

Ligne 8 : on cherche dans A où se trouve le numéro tiré ;

Ligne 10 : si i est de longueur nulle c'est que le numéro n'est pas présent dans A ;

Ligne 11 : on le rajoute donc dans le vecteur A ;

Ligne 12 : à chaque tirage on rajoute 1 au compteur.

On renvoie le compteur à la fin qui représente donc le numéro du tirage à partir duquel chaque numéro a été tiré au moins une fois.

Solution du test 5.21

1. On initialise un entier S qui est incrémenté par $1 + \text{rd.binomial}(1, p, 1)[0]$ tant que S ne dépasse pas k :

```
import numpy.random as rd
def f1(p, k):
    S = 0
    while S < k :
        S += 1 + rd.binomial(1, p, 1)[0]
        print(S) # pour vérifier le cheminement
    return 1 if S == k else 0
```

2. $\mathbb{E}(Y-1) = p$ donc par linéarité de l'espérance on a $\mathbb{E}(Y_1) = 1 + p$. On calcule donc $f1(p, k)$ n fois, on fait la somme et on compare à $1(1+p)$.

```
def f2(n, p, k):
    return [ sum([f1(p, k) for essai in range(n)]) / n, 1/(1+p) ]
```

Par exemple:

```
In [21]: f2(1000, 0.7, 100)
Out [21]: [0.601, 0.5882352941176471]
```



```

In [22]: f2(1000, 0.3, 100)
Out [22]: [0.768, 0.7692307692307692]
In [23]: f2(1000, 0.2, 100)
Out [23]: [0.82, 0.8333333333333334]
In [24]: f2(1000, 0.8, 100)
Out [24]: [0.576, 0.5555555555555556]

```

Il y a de la loi géométrique là-dedans...

Solution du test 5.22

1. La variable aléatoire C prend deux valeurs : a et $-a$. On obtient :

c_i	$-a$	a
$\mathbb{P}(C = c_i)$	$\frac{19}{37}$	$\frac{18}{37}$

Alors $\mathbb{E}(C) = -a \frac{19}{37} + a \frac{18}{37} = -\frac{a}{37}$: le casino est gagnant...

$$\sigma(C) = \sqrt{\left(-a + \frac{a}{37}\right)^2 \times \frac{19}{37} + \left(a + \frac{a}{37}\right)^2 \times \frac{18}{37}} = \frac{6a\sqrt{38}}{37}$$

2. La variable aléatoire N prend deux valeurs : $35a$ et $-a$. On obtient :

n_i	$-a$	$35a$
$\mathbb{P}(N = n_i)$	$\frac{36}{37}$	$\frac{1}{37}$

Alors $\mathbb{E}(N) = -a \frac{36}{37} + 35a \frac{1}{37} = -\frac{a}{37}$: le casino est toujours gagnant...

$$\sigma(N) = \sqrt{\left(-a + \frac{a}{37}\right)^2 \times \frac{36}{37} + \left(35a + \frac{a}{37}\right)^2 \times \frac{1}{37}} = \frac{216a}{37}$$

3. Les espérances sont égales mais les écarts-types très différents : $\sigma(N) \approx 6 \times \sigma(C)$. Il est donc beaucoup plus risqué de miser sur un numéro.

Solution du test 5.23

1. Pour chaque patient, l'extraction d'une dent est une épreuve de BERNOULLI de paramètre $\frac{1}{32}$ (et une épreuve pour le patient). La variable aléatoire X suit donc une loi binomiale de paramètres n et $\frac{1}{32}$. La probabilité pour qu'aucune dent malade n'ait été extraite vaut donc

$$\mathbb{P}(X = 0) = \left(\frac{31}{32}\right)^n$$

2. On veut déterminer n tel que $\mathbb{P}(X \geq 1) > 0,9$ ce qui équivaut à $1 - \mathbb{P}(X = 0) > 0,9$ c'est-à-dire $\mathbb{P}(X = 0) < 0,1$.

Cela revient à résoudre l'inéquation $\left(\frac{31}{32}\right)^n < 0,1$ dans \mathbb{N} . On obtient $n > \frac{\ln(0,1)}{\ln\left(\frac{31}{32}\right)} \approx 72,5$.

Il faudra donc traiter au minimum 73 patients pour extraire au moins une dent malade avec une probabilité de 0,9...

3. Calculons la probabilité d'arracher d dents saines puis une dent malade. Au départ il y a 31 dents saines sur 32, puis 30 dents saines sur 31, etc. À la fin, il faut arracher 1 dent malade parmi les $32 - d$ qui restent.

$$\mathbb{P}(Y = d) = \frac{31}{32} \times \frac{30}{31} \times \cdots \times \frac{32-d}{32-(d-1)} \times \frac{1}{32-d} = \frac{1}{32}$$

La variable aléatoire Y suit donc la loi uniforme sur $\llbracket 0, 31 \rrbracket$.

La probabilité de repartir édenté est $\mathbb{P}(d = 31) = \frac{1}{32}$.

$$\mathbb{E}(Y) = \sum_{d=0}^{31} d \mathbb{P}(Y = d) = \frac{1}{32} \times \frac{31 \times 32}{2} = \frac{31}{2}$$

$$\mathbb{V}(Y) = \sum_{d=0}^{31} \left(d - \frac{31}{2}\right)^2 \times \frac{1}{32} = \frac{341}{4} \text{ donc } \sigma(Y) = \frac{\sqrt{341}}{2}$$

Solution du test 5.24

1. Notons p la probabilité d'obtenir « 1 » avec le dé pipé alors la probabilité d'obtenir « 2 » est $2p$ et, de manière générale, la probabilité d'obtenir « k » est kp .

Or il n'y a que 6 faces donc :

$$\sum_{k=1}^6 kp = 1 = p \sum_{k=1}^6 k = p \frac{6(6+1)}{2} = 21p$$

On a donc : $p = \frac{1}{21}$. Finalement, si on note X la variable aléatoire égale au chiffre donné par le dé pipé :

$$\mathbb{P}(X = k) = \frac{k}{21}$$

Représentons cette loi dans un tableau :

k	1	2	3	4	5	6
$\mathbb{P}(X = k)$	$\frac{1}{21}$	$\frac{2}{21}$	$\frac{3}{21}$	$\frac{4}{21}$	$\frac{5}{21}$	$\frac{6}{21}$

On en déduit que :

$$\mathbb{E}(X) = \sum_{k=1}^6 \frac{k^2}{21} = \frac{13}{3}$$

$$\text{Puis : } \mathbb{E}(X^2) = \sum_{k=1}^6 \frac{k^3}{21} = \frac{441}{21} = 21 \text{ donc :}$$

$$\mathbb{V}(X) = 21 - \left(\frac{13}{3}\right)^2 = \frac{20}{9}$$

2. On répète n fois dans les mêmes conditions et de manière indépendante la même expérience qui n'a que deux issues contraires : « tirer ou ne pas tirer le dé truqué » dont l'issue favorable a pour probabilité $\frac{1}{6}$.

On en déduit que $X \rightsquigarrow \mathcal{B}\left(n, \frac{1}{6}\right)$.

On cherche ensuite à déterminer n tel que $\mathbb{P}(X \geq 1) \geq \frac{1}{2}$.

Or $\mathbb{P}(X \geq 1) = 1 - \mathbb{P}(X = 0)$ donc

$$\mathbb{P}(X \geq 1) \geq \frac{1}{2} \Leftrightarrow \mathbb{P}(X = 0) \leq \frac{1}{2}$$

Or $\mathbb{P}(X=0) = \left(1 - \frac{1}{6}\right)^n$.

On doit donc résoudre l'inéquation :

$$\begin{aligned} \left(\frac{5}{6}\right)^n \leq \frac{1}{2} &\Leftrightarrow n \ln\left(\frac{5}{6}\right) \leq \ln\left(\frac{1}{2}\right) \\ &\Leftrightarrow n \geq \frac{\ln\left(\frac{1}{2}\right)}{\ln\left(\frac{5}{6}\right)} \end{aligned}$$

On doit donc effectuer au moins 4 tirages.

3. Cette fois les tirages s'effectuent sans remise.

On en déduit que $X \rightsquigarrow \mathcal{H}\left(6, n, \frac{1}{6}\right)$.

Cette fois encore on cherche les valeurs de n telles que $\mathbb{P}(X \geq 1) \geq \frac{1}{2}$ c'est-à-dire $\mathbb{P}(X=0) \leq \frac{1}{2}$.

On a :

$$\begin{aligned} \mathbb{P}(X=0) &= \frac{\binom{1}{0} \times \binom{n}{5}}{\binom{n}{6}} \\ &= 1 \times \frac{5!}{n!(5-n)!} \times \frac{n!(6-n)!}{6!} \\ &= \frac{6-n}{6} \end{aligned}$$

Or $\frac{6-n}{6} \leq \frac{1}{2} \Leftrightarrow 3 \leq n$.

On doit donc effectuer au moins 3 tirages.

Solution du test 5.25

Soit X la variable aléatoire qui à chaque disposition de la file d'attente fait correspondre la distance entre vous et Monica. Il s'agit de calculer $\mathbb{P}(X=p)$.

Si vous avez le numéro d'ordre le plus petit (notons-le n_0), celui-ci varie entre 1 et $n-p$ sachant que Monica a le numéro $n_0 + p$. Les $n-2$ autres personnes ont alors $(n-2)!$ façons de se placer dans la file. Il y a donc $(n-p) \times (n-2)!$ files telles que vous soyez à la distance p de Monica en ayant un numéro plus petit qu'elle.

Il y a autant de files telles que Monica porte le numéro le plus petit ou il y a $n!$ files d'attentes possibles. Finalement,

$$\mathbb{P}(X=p) = \frac{2(n-p)(n-2)!}{n!} = \frac{2(n-p)}{n(n-1)}$$

Vérifions que la somme des probabilités quand p parcourt $X(\Omega)$ est égale à 1. Il peut y avoir entre 0 et $n-2$ personnes entre vous et Monica donc $X(\Omega) = \llbracket 1, n-1 \rrbracket$.

$$\begin{aligned} \sum_{p \in X(\Omega)} \mathbb{P}(X=p) &= \sum_{p=1}^{n-1} \frac{2(n-p)}{n(n-1)} \\ &= \frac{2}{n-1} \sum_{p=1}^{n-1} 1 - \frac{2}{n(n-1)} \sum_{p=1}^{n-1} p \\ &= \frac{2}{n-1} (n-1) - \frac{2}{n(n-1)} \frac{n(n-1)}{2} \\ &= 1 \end{aligned}$$

Calculons l'espérance de X :

$$\begin{aligned}
 \mathbb{E}(X) &= \sum_{p=1}^{n-1} p \cdot \mathbb{P}(X=p) \\
 &= \sum_{p=1}^{n-1} p \frac{2(n-p)}{n(n-1)} \\
 &= \frac{2}{n-1} \sum_{p=1}^{n-1} p - \frac{2}{n(n-1)} \sum_{p=1}^{n-1} p^2 \\
 &= \frac{2}{n-1} \cdot \frac{n(n-1)}{2} - \frac{2}{n(n-1)} \cdot \frac{n(n-1)(2n-1)}{6} \\
 &= \frac{n+1}{3}
 \end{aligned}$$

Solution du test 5.26

D'après la définition de la variance, en posant $e = \mathbb{E}(X)$:

$$\begin{aligned}
 \mathbb{V}(X) &= \mathbb{E}\left((X - \mathbb{E}(X))^2\right) \\
 &= \mathbb{E}\left((X - e)^2\right) \\
 &= \mathbb{E}(X^2 - 2eX + e^2) \\
 &= \mathbb{E}(X^2) - 2e\mathbb{E}(X) + \mathbb{E}(e^2) \\
 &= \mathbb{E}(X^2) - 2e^2 + e^2 \\
 &= \mathbb{E}(X^2) - e^2 \\
 &= \mathbb{E}(X^2) - (\mathbb{E}(X))^2
 \end{aligned}$$

Solution du test 5.27

Notons $T_n = \sum_{k=1}^n kq^{k-1}$ et $S_n = \sum_{k=0}^n q^k$.

$$\begin{aligned}
 (1-q)T_n &= (1-q) \sum_{k=1}^n kq^{k-1} \\
 &= \sum_{k=1}^n kq^{k-1} - \sum_{k=1}^n kq^k \\
 &= q^0 + 2q^1 + 3q^2 + \dots + nq^{n-1} - q^1 - 2q^2 - \dots - (n-1)q^{n-1} - nq^n \\
 &= q^0 + q^1 + \dots + q^{n-1} - nq^n \\
 &= S_n - nq^n
 \end{aligned}$$

Puisque $nq^n = \frac{1}{\ln q}(n \ln q)e^{n \ln q}$ et que $\lim_{n \rightarrow +\infty} n \ln q = -\infty$ car $0 \leq q < 1$, on obtient $\lim_{n \rightarrow +\infty} nq^n = 0$ donc $(1-q)T = S$ et

$$T = \frac{1}{1-q} S = \frac{1}{(1-q)^2}$$

$$\mathbb{E}(X) = \sum_{n \geq 1} n \times \mathbb{P}(\{X=n\}) = \sum_{n \geq 1} np(1-p)^{n-1} = p \sum_{n \geq 1} nq^{n-1} \text{ en notant } q = 1-p.$$

D'après ce qui précède, on obtient $\mathbb{E}(X) = p \times T = p \times \frac{1}{(1-(1-p))^2} = \frac{1}{p}$.

« 1991 - Dutch programmer Guido van Rossum travels to Argentina for a mysterious operation. He returns with a large cranial scar, invents Python, is declared Dictator for Life by legions of followers, and announces to the world that "There Is Only One Way to Do It!" Poland becomes nervous. »

James IRY – *« A brief, incomplete, and mostly wrong history of programming languages »*

Un enseignement d'informatique pour tous est officiellement entré en vigueur à la rentrée 2013. À l'heure où nous imprimons, le langage retenu pour illustrer cet enseignement est **Python** que ce soit en MP, MPI, PC, PSI, PT, TSI, BCPST ou ECG. On présentera également dans ces sections une introduction aux bases de données et au langage SQL. Les MPI étudieront en plus les langages **OCaml** et **C**.

6.1 Avant de commencer

— Python

Python est un vrai langage de programmation. Il est naturellement disponible sur toute distribution GNU-Linux et vous pouvez même l'utiliser sur les autres systèmes d'exploitation. C'est un langage de très haut niveau dont la syntaxe encourage à écrire du code clair et de qualité. Dans le domaine de la gestion de la mémoire, les détails de bas niveau de langages comme le C disparaissent. De plus, l'apprentissage de **Python** est facilité par l'existence d'une interface interactive.

Ceci dit, son intérêt ne se réduit pas à l'apprentissage de la programmation ou de l'algorithmique, en témoigne sa popularité croissante. Il a été choisi par des acteurs majeurs : Google, YouTube, la NASA, etc. Il favorise la programmation impérative structurée et la programmation orientée objet ; dans une moindre mesure, il permet de programmer dans un style fonctionnel. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. C'est un langage multi-plateforme, polyvalent (jusque dans les domaines comme le web, les graphiques, le réseau), open source et gratuit.

En **Python**, tout est objet mais la programmation orientée objet n'est vue que de manière optionnelle en prépa.

Python est accompagné d'un très grand nombre de bibliothèques. Les plus utiles pour le programme de prépa seront **numpy**, **scipy**, **matplotlib**, qui permettent de faire du calcul scientifique.

— Environnement de travail

Python est un langage qui dispose d'un interpréteur très pratique pour tester de petits bouts de code.

On distinguera donc un fichier texte d'extension `.py`, dans lequel on écrira nos diverses lignes de code que nous enregistrerons et pourrons utiliser à diverses occasions, et nos « dialogues » avec **Python**. Cette approche intuitive suffit pour l'instant : vous parlerez plus tard de langage compilé, interprété, semi-interprété, **bytecode**, **opcode**, etc.

Il existe de très nombreux environnements de travail adaptés à **Python**. Nous ne parlerons que de **iPython**, qui est utilisé dans les distributions **Pyzo** (<https://www.pyzo.org/>) et **Spyder** (<https://pythonhosted.org/spyder/>) le plus souvent disponibles aux concours et dans tous les lycées.

Pour installer **Spyder** par exemple, il suffit d'aller dans un terminal : « l'écran noir ».

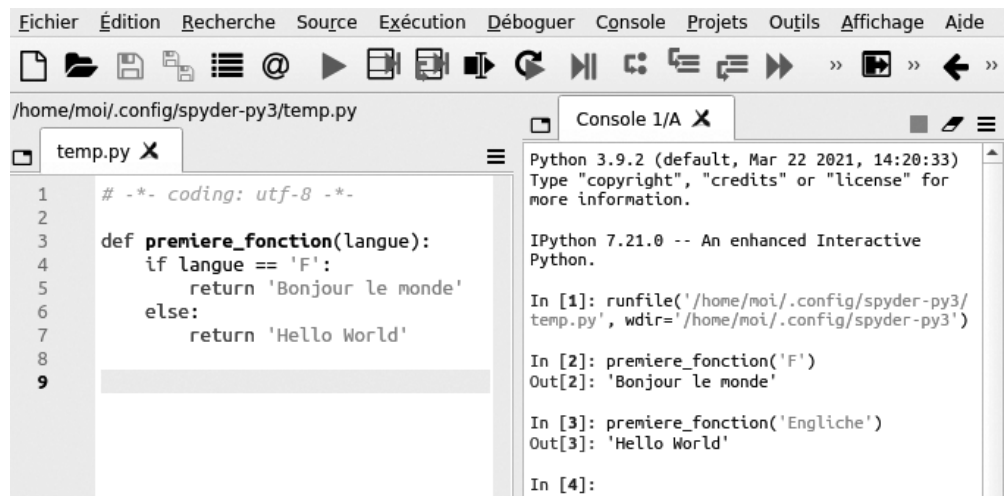
```
python3.X -m pip install spyder
```

Vous remplacerez le X par la version de **Python** disponible sur votre machine et éventuellement suivi de `.exe` si vous êtes sur Windows. Dans la suite de cet ouvrage, nous utiliserons **Python 3.9**.

```
Python 3.9.2 (default, Mar 22 2021, 14:20:33)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.21.0 -- An enhanced Interactive Python. Type '?'
for help.

In [1]:
```

Et voici l'aspect de **Spyder** :



6.2 Premiers pas

Python comme super-calculatrice

Si l'on veut faire des calculs simples, on peut aller dans la console **iPython** de son environnement **Spyder**.

On importe au besoin certaines bibliothèques de calcul comme par exemple **sympy** pour le calcul symbolique, **scipy** et **numpy** pour le calcul numérique, à partir de la console **iPython** avec la commande magique `%pip` :

```
In[1]: %pip install scipy

Defaulting to user installation because normal site-
packages is not writeable
Collecting scipy
  Downloading scipy-1.6.2-cp39-cp39-manylinux1_x86_64.whl
(27.3 MB)
|████████████████████████████████████████| 27.3 MB 2.2 MB/s
eta 0:00:01
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in
./local/lib/python3.9/site-packages (from scipy) (1.20.1)
Installing collected packages: scipy
Successfully installed scipy-1.6.2
Note: you may need to restart the kernel to use updated
packages.
```

On utilisera aussi **matplotlib.pyplot** pour les tracés de courbes.

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sy

np.sqrt(2)
Out[7]: 1.4142135623730951

sy.sqrt(2)
Out[8]:
sqrt(2)

np.sqrt(2)**2
Out[9]: 2.0000000000000004

sy.sqrt(2)**2
Out[10]:
2
```

6.3 Un peu de vocabulaire

Quand on écrit :

```
var1 = "Point n'est besoin d'espérer"
var2 = " pour entreprendre"
var3 = var1 + var2
```

On crée trois **objets** de **type** `str`, « Point n'est besoin d'espérer » et « pour entreprendre » sont deux **constantes** (*littéraux* en français) et `var1 + var2` une **expression**. Cela permet de **lier** les **variables** `var1`, `var2` et `var3` à ces constantes ou expressions. On fait ainsi des **affectations**.

Remarque

En **Python**, une expression est évaluée avant d'être affectée.

Par exemple :

```
>>> a = 0
>>> b = 2
>>> c = (1, b/a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
```

— Constante

C'est la représentation en code **Python** d'une valeur d'un certain type. `1` est une constante de type `int`, `True` est une constante de type `bool`, etc.

— Opérateur

C'est la représentation en code **Python** d'une opération sur des objets d'un certain type : par exemple `+` est l'opérateur représentant l'opération *addition des int* mais c'est aussi l'opérateur représentant l'opération *concaténation de deux str*.

— Identifiant

C'est la représentation en code **Python** d'un nom qui est associé à une adresse mémoire. Les identifiants sont des suites de lettres, de chiffres et de tirets bas, le premier caractère ne pouvant être un chiffre.



Attention à ne pas choisir comme identifiants des mots-clés ou des types comme `while`, `int`, etc.

— Variable

Une variable est un nom associé à une valeur pour faciliter son identification par le programmeur et dont la valeur peut changer au cours du programme.

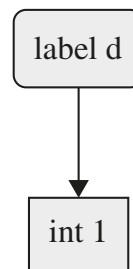
— Expression

Une expression est une combinaison de constantes, de variables et d'opérateurs que **Python** va *évaluer* : c'est-à-dire que **Python** va produire une valeur associée à cette expression.

— Affectation

Quand on écrit `d = 1` en **Python**, on n'exprime pas une identité mathématique mais une **action** qui consiste à :

- définir un *identifiant* `d` comme une nouvelle *variable* ;
- associer à cette variable un entier dont la *valeur* est 1.



On fait bien cette distinction dans d'autres langages comme **OCaml** :

```

# let d = ref 1;;
val d : int ref = {contents = 1}

# d;;
- : int ref = {contents = 1}

# !d;;
- : int = 1
  
```

Quand on écrit :

```

a = 12
t = a
  
```

on a deux variables `a` et `t` qui pointent vers le même objet 12 (qui est ici un `int`).

— Objet

Toutes les données d'un programme **Python** sont des objets caractérisés par :

- leur **identifiant** : en gros, cela indique un numéro qui identifie une case mémoire ;
- leur **type** : cela permet de connaître le comportement de l'objet, ce qu'il peut représenter et ce qu'on peut lui faire faire ;
- leur **valeur** : la donnée que l'objet représente.

— Référence d'un objet

En fait, les noms qu'on donne aux objets sont juste des liens vers les objets. Écrire `a = b` signifie qu'on copie l'adresse `a` dans `b`.

La fonction `id` nous permet d'avoir un *identifiant* unique pour un objet qui sera lui-même unique tant que cet objet restera « vivant » (la plupart des objets disparaissent dans le *ramasse-miettes*). C'est comme un numéro de sécurité sociale.

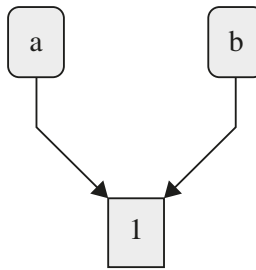
Observez, commentez, expliquez

```
>>> a = 1

>>> b = a

>>> id(a)
93835776299776

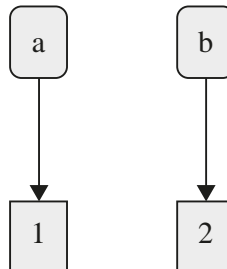
>>> id(b)
93835776299776
```



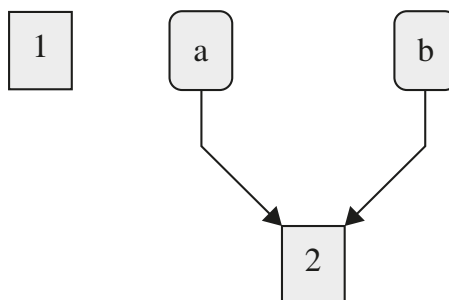
```
>>> b = 2

>>> id(b)
93835776299808

>>> id(a)
93835776299776
```



```
>>> a = b
>>> id(a)
93835776299808
```



— Et dans d'autres langages ?

D'autres langages n'ont carrément pas d'affectation, comme **Haskell** qui a inspiré de nombreux langages parmi les plus utilisés actuellement.

On ne peut pas écrire $a = a + 1$.

Nous en rediscuterons plus tard mais on essaiera de recourir le plus souvent possible aux **affectations uniques**.

On évitera les $a = a + a$ qui rendent la lecture du code difficile.

On préférera utiliser une nouvelle variable.

Quant à l'utilisation du symbole $=$ pour l'affectation, voici ce qu'en dit Niklaus WIRTH :

« A notorious example for a bad idea was the choice of the equal sign to denote assignment. It goes back to Fortran in 1957 and has blindly been copied by armies of language designers. Why is it a bad idea? Because it overthrows a century old tradition to let “=” denote a comparison for equality, a predicate which is either true or false. But Fortran made it to mean assignment, the enforcing of equality. In this case, the operands are on unequal footing: The left operand (a variable) is to be made equal to the right operand (an expression). $x = y$ does not mean the same thing as $y = x$. »

— On retiendra

En **Python**, on peut lire $a = 1$ comme « a est lié à un objet de type `int` dont la valeur est 1 ».

On peut même préciser `a : int = 1`.

En **Python**, on copie des adresses, pas des valeurs.

6.4 Les types de base

— Les chaînes de caractères

Les chaînes de caractères sont de type `str`. Un simple caractère est considéré comme une chaîne de caractères de longueur 1, ce qui n'est pas vrai dans d'autres langages.

Les chaînes de caractères sont représentées par des caractères entre simples ou doubles guillemets :

```
>>> "Bonjour le Monde"
'Bonjour le Monde'

>>> 'Bonjour le Monde'
'Bonjour le Monde'

>>> "C'est la vie"
"C'est la vie"

>>> 'C'est la vie'
File "<ipython-input-28-11202c0f01b2>", line 1
    'C'est la vie'
      ^
SyntaxError: invalid syntax

>>> 'Il a dit "oui !"'
'Il a dit "oui !"'

>>> "Il a dit "oui !""
File "<ipython-input-30-abb082a6ed83>", line 1
    "Il a dit "oui !""
              ^
SyntaxError: invalid syntax

>>> print('C\'est "Oui!"')
C'est "Oui!"
```

On peut **concaténer** deux chaînes avec l'opérateur `+` :

```
>>> 'Bonjour' + ' le Monde'
'Bonjour le Monde'

>>> '12' + '34'
'1234'

>>> '12' + ' ' + ' ' + '34'
'12  34'
```

```
>>> '12' + 34
-----
TypeErrorTraceback (most recent call last)
<ipython-input-36-afb1c3d28cba> in <module>
----> 1 '12' + 34

TypeError: can only concatenate str (not "int") to str
```

Convertir en chaîne de caractères

On effectue souvent des échanges avec le monde extérieur *via* des chaînes de caractères (import/export de fichiers) car c'est aussi ce que se transmettent les commandes du shell depuis les années 1970.

On a donc besoin de convertir des objets en `str` (vous verrez cela plus en détail en prépa). Il suffit d'utiliser en **Python** la commande `str` :

```
>>> str(12 + 5)
'17'

>>> str(12) + str(5)
'125'

>>> a = input('Donne-moi un entier : ')
Donne-moi un entier : 12

>>> a
'12'

>>> type(a)
str
```

— Les entiers

Les entiers de **Python** n'ont pas de limitation en taille si ce n'est l'espace mémoire qu'ils occupent.

Les opérateurs habituels sont `+`, `-`, `*`, `**`, `//`, `%`.

La fonction `int` a différents usages. Voyons la documentation :

```
>>> ?int
Init signature: int(self, /, *args, **kwargs)
Docstring:
int([x]) -> integer
int(x, base=10) -> integer

Convert a number or string to an integer, or return 0 if no
arguments are given. If x is a number, return x.__int__().
For floating point numbers, this truncates towards zero.
```

If x **is not** a number **or if** base **is** given, then x must be a string, bytes, **or** bytearray instance representing an integer literal **in** the given base. The literal can be preceded by '+' **or** '-' **and** be surrounded by whitespace. The base defaults to 10. Valid bases are 0 **and** 2-36. Base 0 means to interpret the base from the string as an integer literal.

```
>>> int('0b100', base=0)
4
```

Par exemple :

```
>>> int('12') + int('5')
17

>>> a = int(input('Donne-moi un entier : '))
Donne-moi un entier : 12

>>> a + 5
17

>>> 5 // 2
2

>>> 5 % 2
1

>>> int('bac', 16)
2988

>>> int('1101', 2)
13
```

— Les nombres à virgule flottante

Nous parlerons en détail de la représentation des nombres à virgule flottante en machine plus tard. Ils sont représentés en **Python** par le type float. Les opérateurs de base sont +, -, *, **, /.

À noter la syntaxe pour utiliser les puissances de 10. Le nombre $1,602 \times 10^{-19}$ s'écrit 1.602e-19.

Si l'on a besoin de la racine carrée ou de toute autre fonction élémentaire ou même de π , on peut les charger depuis la bibliothèque **math** :

```
>>> import math
>>> math.sqrt(2)
1.4142135623730951
>>> math.pi
3.141592653589793
>>> math.sqrt(math.pi**2)
3.141592653589793
```

```
>>> math.sqrt(-1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: math domain error
```

La fonction `float` va agir un peu comme `int` pour convertir un `str` en `float` :

```
>>> x = float(input('Donne-moi un nombre à virgule flottante : '))
Donne-moi un nombre à virgule flottante : 3.1415926
>>> x
3.1415926
>>> x.is_integer()
False
>>> x / 2
1.5707963
```

Il s'agit de conversions *explicites* car on précise la nature de la conversion (pour *explicite*, le dictionnaire précise : « *Qui est énoncé complètement et ne peut prêter à aucune contestation* »).

Certaines conversions peuvent être *implicites* (définition du dictionnaire pour *implicite* : « *Qui, sans être énoncé formellement, découle naturellement de quelque chose* ») :

```
>>> a = 3
>>> type(a)
<class 'int'>
>>> b = a * 2.0
>>> type(b)
<class 'float'>
>>> math.sqrt(4)
2.0
>>> type(math.sqrt(4))
<class 'float'>
```

— Les booléens

Nous avons déjà pas mal exploré les booléens au chapitre 1. En **Python** leur type est `bool`. Notons que c'est un sous-type des entiers et que `True` est un synonyme de 1 et `False` un synonyme de 0 :

```
>>> a = True
>>> a == 1
True
>>> 5*a
5
```

Et, avec un peu d'avance sur notre programme, cela nous permettra de compter les bonnes « réponses » à une « question » :

```
>>> def est_voyelle(c): return c in "aeiouy"

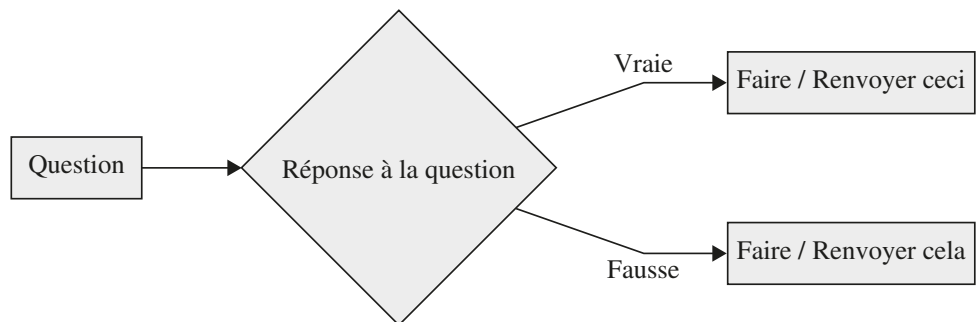
>>> texte = "Use the force young Padawan"

>>> sum(est_voyelle(c.lower()) for c in texte)
11
```

Il y a 11 voyelles dans la phrase « *Use the force young Padawan* ».

6.5 Instructions et expressions conditionnelles

Une fonction renvoyant un booléen permet de répondre par Vrai ou Faux à une question.



- Si on *fait* quelque chose, on parle d'**instruction** conditionnelle.
- Si on *renvoie* une valeur, on parle d'**expression** conditionnelle.

Voyons un calcul de maximum :

- **Instruction conditionnelle :**

```
if x > y:
    maxi = x
else:
    maxin = y
```

- **Expression conditionnelle :**

```
maxi = x if x > y else y
```


6.6 Elif

Parfois, les choix ne sont pas binaires. On doit alors distinguer plus de deux cas.

```
if age < 18:
    message = "Va te coucher !"
elif age < 60:
    message = "Tu peux regarder le film"
else :
    message = "Reveille-toi !"
```

ou

```
message = "Va te coucher" if age < 18 else "Tu peux
regarder le film" if age < 60 else "Reveille-toi !"
```

6.7 Survol des structures de données

Vous aurez besoin de parcourir des « groupes » de données qui peuvent avoir différentes propriétés selon vos besoins. Nous allons rapidement parler d'un premier lot sans trop détailler sachant que vous irez plus loin si besoin une fois en prépa.

Nous allons surtout les découvrir telles que **Python** nous les fournit et c'est assez spécifique.

Listes...Python (list)

Les listes **Python**, de type `list`, ne sont pas des listes au sens algorithmique du terme mais pour l'instant nous ne nous en occuperons pas.

Une liste Python est une structure ordonnée et mutable et dont les éléments sont de type homogène (autant que faire se peut)...

Les éléments d'une liste sont numérotés de « gauche » à « droite » à partir de 0 et de « droite » à « gauche » à partir de -1 :

```
-5 -4 -3 -2 -1

[ a, b, c, d, e ]

0  1  2  3  4
```

Par exemple, créons une liste de chaînes de caractères :

```
>>> xs: list[str] = ['Papa', 'Maman', 'La bonne', 'Moi']
>>> xs
['Papa', 'Maman', 'La bonne', 'Moi']
>>> xs[0]
'Papa'
>>> xs[1]
'Maman'
```

```
'Maman'
>>> xs[2]
'La bonne'
>>> xs[3]
'Moi'
>>> xs[-1]
'Moi'
>>> xs[-2]
'La bonne'
>>> xs[-3]
'Maman'
>>> xs[-4]
'Papa'
>>> xs[-5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>> xs[4]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

`len` comme *LEN*gth (« longueur », en anglais) nous permet d'avoir accès à la longueur de la liste i.e. le nombre de ses éléments :

```
>>> len(xs)
4
```

On peut extraire des **sous-listes** d'une liste (ou *tranches* ou *slices*) à l'aide de la syntaxe :

```
xs[indice début : indice fin]
```



Attention ! L'indice de fin n'est pas compris.

```
>>> xs[1:3]
['Maman', 'La bonne']
```

Si l'on veut aller jusqu'à la fin, on n'écrit rien à droite des :

```
>>> xs[1:]
['Maman', 'La bonne', 'Moi']
```

Si l'on veut partir du début, on ne met rien à gauche des :

```
>>> xs[:3]
['Papa', 'Maman', 'La bonne']
```

Expliquez :

```
>>> xs[1:-1]
['Maman', 'La bonne']
```

Concaténation – Mutabilité

On peut « coller » des listes ou plutôt les **concaténer** avec l'opérateur + :

```
>>> xs + ['Ma soeur']
['Papa', 'Maman', 'La bonne', 'Moi', 'Ma soeur']

>>> ['Odin'] + xs
['Odin', 'Papa', 'Maman', 'La bonne', 'Moi']
```

Une liste **Python** étant *mutable*, on peut la modifier :

```
>>> xs = ['Odin'] + xs
>>> xs
['Odin', 'Papa', 'Maman', 'La bonne', 'Moi']
>>> xs = xs[1:]
>>> xs
['Papa', 'Maman', 'La bonne', 'Moi']
```

On peut utiliser une fonction spécifique aux listes, qui n'existe donc que pour celles-ci, et dont la syntaxe est différente : si on veut ajouter un élément à la fin, on utilise `append`.

```
>>> xs.append('Ma soeur')
>>> xs
['Papa', 'Maman', 'La bonne', 'Moi', 'Ma soeur']
```

Il y a deux choses à noter :

- `append` ne renvoie rien, ou plutôt renvoie `None` mais, « en douce », modifie la liste et lui ajoute un élément à la fin. Nous en reparlerons dans notre section sur les fonctions.
- Pour appliquer `append` à la liste, on écrit `liste.append(élément)`. Nous comprendrons plus tard pourquoi : cela vient de l'orientation « objet » de la programmation en **Python**.

Test d'appartenance

Ce sera souvent utile :

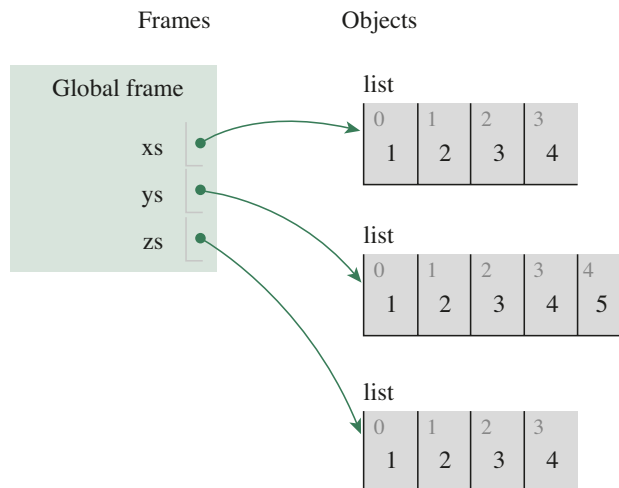
```
>>> xs = [1, 2, 3]
>>> 2 in xs
True
>>> 4 in xs
False
```

Les listes et la mémoire

Observez, c'est étrange...

```
>>> xs = [1, 2, 3]
>>> id(xs)
140077907522656
>>> id(xs[0])
93990234841856
>>> ys = xs
>>> id(ys)
140077907522656
>>> xs.append(4)
>>> xs
[1, 2, 3, 4]
>>> ys
[1, 2, 3, 4]
>>> ys = ys + [5]
>>> xs
[1, 2, 3, 4]
>>> id(xs)
140077907522656
>>> id(ys)
140077906785616
>>> zs = xs[:]
>>> id(zs)
140077906700240
>>> zs
[1, 2, 3, 4]
>>> xs
[1, 2, 3, 4]
>>> ys
[1, 2, 3, 4, 5]
```

Allez observer comment la mémoire est utilisée en tapant le code précédent sur le site <http://pythontutor.com/> et vous obtiendrez l'image finale suivante :



— Les chaînes de caractères vues (presque) comme des listes

```
>>> s = "Petit Papa Noël!"

>>> len(s)
16

>>> s[:10]
'Petit Papa'

>>> s[1] = 'o'
-----
TypeError                                Traceback (most recent call last)
----> 1 s[1] = 'o'

TypeError: 'str' object does not support item assignment

>>> s.split(' ')
['Petit', 'Papa', 'Noël!']

>>> ss = "abracadabrantesque"

>>> ss.split('a')
['', 'br', 'c', 'd', 'br', 'ntesque']

>>> ss.split('a')
['', 'br', 'c', 'd', 'br', 'ntesque']
>>> ' '.join(["Vive", "la", "Syldavie" ])
'Vive la Syldavie'

>>> '$$'.join(["Vive", "la", "Syldavie" ])
'Vive$$la$$Syldavie'
```

— Les n-uplets (tuple)

Les n-uplets (tuples) de **Python**, c'est presque pareil, sauf que...

Un tuple Python est une structure ordonnée et *NON mutable* et dont les éléments sont de type *homogène OU NON*...

Donc on favorisera l'utilisation de n-uplets lorsqu'on voudra manipuler des objets de types divers et qui n'ont pas vocation à évoluer.

On peut cependant faire évoluer des n-uplets par concaténation :

```
>>> (1,2) + (3,4)
(1, 2, 3, 4)
```

Mais comment ajouter un seul élément ? Pour **Python**, (1) n'est pas un tuple, c'est juste un 1 entre parenthèses...

```
>>> type((1))
<class 'int'>
```

Ruse ! On écrit (1,).

```
>>> type((1,))
<class 'tuple'>
```

Ainsi :

```
>>> (1,2) + (3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate tuple (not "int") to tuple
>>> (1,2) + (3,)
(1, 2, 3)
```

Bref, tout s'explique quand on cherche à comprendre !

Test d'appartenance

```
>>> t = (1, 'a', 12.3)
>>> 1 in t
True
```

— Affectation multiple

C'est souvent très pratique.

```
>>> (x, y) = (1, 3)
>>> x
1
>>> y
3
```

On peut même **dans ce cas** se passer de parenthèses, qui seront mises automatiquement :

```
>>> x, y = 1, 3
>>> x
1
>>> y
3
```

Observez et expliquez :

```
>>> x, y = 1, 2
>>> x
1
>>> y
2
>>> x, y = y, x
>>> x
2
>>> y
1
>>> x, y = y, x + y
>>> x
1
>>> y
3
```

— Les ensembles (set)

Un ensemble Python (**set**) est une structure *NON ordonnée* et *NON mutable* et dont les éléments sont de type *homogène OU NON* et **SANS RÉPÉTITIONS...**

```
>>> s1 = {1, 2, 3}
>>> s2 = {2, 3, 1}
>>> s3 = {1, 3, 1, 2, 2, 3}
>>> s1 == s2
True
>>> s1 == s3
True
>>> s1
{1, 2, 3}
>>> s2
{1, 2, 3}
>>> s3
{1, 2, 3}
```

Opérations

On a l'union | :

```
>>> s1 = {1, 2, 3}
>>> s2 = {2, 3, 4, 5}
>>> s1 | s2
{1, 2, 3, 4, 5}
```

On a l'intersection & :

```
>>> s1 & s2
{2, 3}
```

On a l'appartenance in :

```
>>> 2 in {1, 2, 3}
True
```

On a l'inclusion < ou <= :

```
>>> {1, 2} < {1, 2, 3}
True
>>> {1, 2} <= {1, 2}
True
>>> {1, 2} < {1, 2}
False
```

Et on la différence (on enlève les éléments du second dans le premier) :

```
>>> {1, 2, 3} - {2, 3, 4}
{1}
>>> {2, 3, 4} - {1, 2, 3}
{4}
```

➤ Range

C'est un objet un peu à part... Voyons l'aide :

```
Help on class range in module builtins:

class range(object)
|   range(stop) -> range object
|   range(start, stop[, step]) -> range object
|
|   Return an object that produces a sequence of integers
|   from start (inclusive)
|   to stop (exclusive) by step.  range(i, j) produces i,
|   i+1, i+2, ..., j-1.
|   start defaults to 0, and stop is omitted!  range(4)
|   produces 0, 1, 2, 3.
|   These are exactly the valid indices for a list of 4
|   elements.
|   When step is given, it specifies the increment (or
|   decrement).
```



```
>>> r = range(1, 5)
>>> 1 in r
True
>>> 0 in r
False
>>> 4 in r
True
>>> 5 in r
False
>>> 2.5 in r
False
```

6.8 Les boucles

— Boucles conditionnelles

L'humain pense, la machine répète.

Supposons que l'on veuille afficher une série de « Bonjour » numérotés :

```
"C'est mon bonjour numéro 1"
"C'est mon bonjour numéro 2"
"C'est mon bonjour numéro 3"
"C'est mon bonjour numéro 4"
```

On écrit :

```
no = 1

while no < 5:
    print( f"C'est mon bonjour numéro {no}" )
    no = no + 1
```

La syntaxe est :

```
while <expression booléenne>:
    faire ceci
    faire cela
faire autre chose quand l'expression booléenne est devenue
fausse
```

En fait, cela revient à écrire :

```
no = 1

if no < 5:
    print( f"C'est mon bonjour numéro {no}" )
    no = no + 1

if no < 5:
```

```

    print( f"C'est mon bonjour numéro {no}" )
    no = no + 1

if no < 5:
    print( f"C'est mon bonjour numéro {no}" )
    no = no + 1

if no < 5:
    print( f"C'est mon bonjour numéro {no}" )
    no = no + 1

...

```

Mais qui s'arrêterait lorsque l'expression booléenne devient fausse.

Incrément

L'expression `no = no + 1` est bien sûr une affectation et non pas une égalité mathématique qui serait absurde.

Comme dans la plupart des autres langages qui permettent l'affectation, **Python** dispose d'un *sucré syntaxique* : `no += 1` qui se décline avec tout autre opérateur (par exemple `n -= 1`, `n *= 2`, `n /= 2`,...).

Par exemple, **quelle est la première puissance de 2 qui dépasse 10 000 ?**

```

In [39]: n = 1

In [40]: while n < 10000:
...:     n *= 2

In [41]: n
Out[41]: 16384

In [42]: n = 1

In [43]: while n < 10000:
...:     n <= 1

In [44]: n
Out[44]: 16384

```

— Boucles *foreach* : pour chaque élément d'une collection

On peut aussi prendre chacun des éléments d'une collection et faire quelque chose avec ceux-ci.

Par exemple, on dispose d'une liste d'élèves et on veut leur dire bonjour :

```

In [49]: classe = {'Joe', 'Max', 'Bill', 'Zoé', 'Isa',
...:               'Zaza'}

In [50]: for élève in classe:
...:     print(f"Bonjour {élève}")

```

```

...:
Bonjour Isa
Bonjour Bill
Bonjour Joe
Bonjour Max
Bonjour Zoé
Bonjour Zaza

```

Si l'on veut un peu d'ordre :

```

In [53]: classe = ['Joe', 'Max', 'Bill', 'Zoé', 'Isa',
'Zaza']

In [54]: for élève in classe:
...:     print(f"Bonjour {élève}")
...:
Bonjour Joe
Bonjour Max
Bonjour Bill
Bonjour Zoé
Bonjour Isa
Bonjour Zaza

```

La syntaxe est donc :

```

for élément in collection: # pour chaque élément dans la
                           # collection

    faire ceci

faire cela # quand on a fini de parcourir TOUS les
           # éléments de la collection

```

— Cas particulier du parcours d'un "range"

Si l'on veut parcourir une collection de nombres entiers espacés régulièrement, on peut utiliser `for indice in range(début, fin + 1, pas[qui vaut par défaut 1])`. Par exemple, si on veut sélectionner un élève sur deux dans la classe en partant du premier :

```

In [1]: classe = ['Joe', 'Max', 'Bill', 'Zoé', 'Isa',
'Zaza']

In [2]: for indice in range(0, len(classe), 2):
...:     print(classe[indice])
...:
Joe
Bill
Isa

```

— Défi

On dispose d'une liste `fi = [1, 1, 0, 0, 0, ..., 0]` de longueur 20.

On veut remplacer tous les termes à partir du troisième par la somme des deux précédents.

On veut donc obtenir `[1, 1, 2, 3, 5, 8, 13, ...]`.

Comment faire ?

— Le beurre et l'argent du beurre : obtenir la valeur d'un élément et son rang dans une liste

Comment mettre en majuscules les lettres de rang pair d'un mot écrit en minuscules.

On utilisera la méthode `upper()`, qui transforme une lettre en la majuscule correspondante :

```
In [3]: 'a'.upper()
Out[3]: 'A'

In [4]: 'A'.upper()
Out[4]: 'A'

In [5]: '1'.upper()
Out[5]: '1'

In [6]: 'Papa'.upper()
Out[6]: 'PAPA'
```

Et on utilisera surtout `enumerate(collection)`, qui renvoie l'ensemble des couples (rang, valeur) pour chaque valeur de la collection :

```
In [14]: mot = input('Rentrer un mot : ')
Rentrer un mot : anticonstitutionnellement

In [15]: new_mot = ''
In [16]: for rang, lettre in enumerate(mot):
...:     new_mot += lettre.upper() if rang % 2 == 0
else lettre

In [17]: new_mot
Out[17]: 'AnTiCoNsTiTuTiOnNeLlEmEnT'
```

— Définitions par compréhension

On dispose d'une liste de lettres en minuscules et on veut créer une autre liste des mêmes lettres en majuscules :

```
In [21]: ls = ['a', 'b', 'c', 'd']

In [22]: [lettre.upper() for lettre in ls]
Out[22]: ['A', 'B', 'C', 'D']
```

On utilise la construction par compréhension :

```
{ f(x) for x in collection }
# l'ensemble des f(x) pour chaque élément x de la collection
```

Filtre

On peut utiliser un filtre en ajoutant une expression booléenne introduite par `if`. Par exemple, une liste de nombre étant donnée, on veut fabriquer la liste des multiples communs de 8 et 6 de cette liste :

```
>>> [nb for nb in range(200) if nb % 6 == 0 and nb % 8 == 0]
[0, 24, 48, 72, 96, 120, 144, 168, 192]
```

6.9 Les fonctions

Vous allez être amenés à construire des programmes de plus en plus élaborés. Vous allez avoir besoin de les organiser en morceaux plus petits. Cela sera plus facile à contrôler, mais cela vous permettra aussi d'utiliser ces blocs élémentaires pour d'autres programmes : c'est la **modularité**. Une fonction est **définie** sur **Python** en l'introduisant avec le mot-clé `def`.

On **appelle** ensuite la fonction... par son nom.

Fonction : premier contact

Définissons par exemple une fonction qui affiche des données personnelles sur une personne :

```
def afficheMesDonnees():
    print("James")
    print("Bond")
    print("jbond@secret.uk")
```

À présent **appelons** cette fonction :

```
>>> afficheMesDonnees()
James
Bond
jbond@secret.uk
```

Une fonction doit être appelée !

Quel est l'intérêt ? On aurait pu tout aussi bien écrire :

```
print("James")
print("Bond")
print("jbond@secret.uk")
```

Oui mais, si nous avons besoin d'écrire souvent et dans différents endroits du code ces adresses, cela nous permet de ne pas tout réécrire et en plus, pour la lecture du code, cela traduit une action.

— Deuxième contact : paramètre et argument

Imaginons que toute la famille Bond ait le même type d'adresse e-mail, seule l'initiale du prénom changeant. On peut alors proposer le code suivant :

```
def afficheLesDonneesDe(prenom):
    print(prenom)
    print("Bond")
    print(f"{prenom[0].lower()}bond@secret.uk")
```

Et cette fois :

```
>>> afficheLesDonneesDe("Max")
Max
Bond
mbond@secret.uk

>>> afficheLesDonneesDe("Barbara")
Barbara
Bond
bbond@secret.uk
```

`afficheLesDonneesDe("Max")` est le code **appelant** (*caller*). Il **pass**e l'**argument** (ou **paramètre réel**) "Max" à la fonction **appelée** (...par l'appelant).

Quand elle reçoit l'argument "Max", la fonction substitue le **paramètre** (ou **paramètre formel**) `prenom` par l'argument "Max".

- Un paramètre (formel) est une variable **non liée** (*unbound*).
- Un argument (ou paramètre réel) peut être une valeur, une variable liée ou une expression complexe.

Ainsi, les paramètres ne changent pas d'un appel à l'autre de la fonction alors que les arguments peuvent changer (en fait, ça peut être plus compliqué et subtil mais on se tiendra à cette explication à notre niveau...).

Dans la pratique, on peut parfois utiliser un terme pour un autre et parler d'argument au lieu de paramètre, ou d'argument formel au lieu de paramètre. Bon, le principal est de bien voir la différence entre les deux notions et on essaiera donc d'utiliser correctement paramètre et argument.

— Troisième contact : plusieurs arguments

Et que se passe-t-il si nous voulons aussi changer le nom de famille et le nom de domaine ? Il faut passer plusieurs arguments et donc avoir une fonction ayant plusieurs paramètres...

```
def afficheLesDonneesDe(prenom, nom, domaine):
    print(prenom)
    print(nom)
    print(f"{prenom[0].lower()} {nom.lower()}@{domaine}")

afficheLesDonneesDe('James', 'Haskell', 'wasps.uk')
```

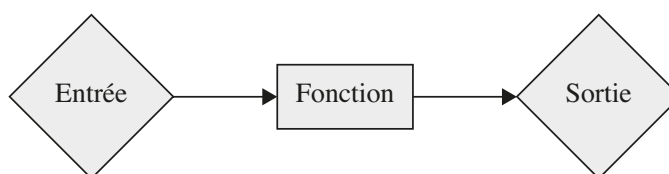
Et on obtient :

```
$ python fonction_cours.py
James
Haskell
jhaskell@wasps.uk
```

— Quatrième contact : le retour

Tout ceci ne nous convient pas pour plusieurs raisons : nous avons comme mot d'ordre de n'avoir qu'un seul `print` par programme pour matérialiser notre sortie finale. Il faut donc abolir les `print` et autres sorties de nos fonctions.

Il faut de plus avoir en tête qu'une fonction est une machine à transformer des paramètres d'entrée en paramètres de sortie :



Nous préférons donc la forme suivante qui utilise le mot-clé `return` :

```
def lesDonneesDe(prenom, nom, domaine):
    donnees = f"{prenom}\n{nom}\n"
    donnees += f"{prenom[0].lower()} {nom.lower()}@{domaine}"
    return donnees

print(lesDonneesDe('James', 'Haskell', 'wasps.uk'))
```

Nous pouvons même typer nos fonctions pour faciliter la lecture des programmes :

```
def lesDonneesDe(prenom: str, nom: str, domaine: str) -> str:
    donnees = f"{prenom}\n{nom}\n"
    donnees += f"{prenom[0].lower()} {nom.lower()}@{domaine}\n"
    return donnees

print(afficheLesDonneesDe('James', 'Haskell', 'wasps.uk'))
```

De manière générale, il faudra présenter les fonctions sous cette forme :

```
def nomDeLaFonction(param1: type1, param2: type2, ...) ->
type_de_la_sortie:
    Traitement des paramètres
    return valeur_de_sortie
```

Pour être complet et si la fonction n'est pas évidente, il faut prendre l'habitude de la documenter à l'aide d'un *docstring* dans lequel on fera figurer une rapide explication et quelques tests. Les *docstrings* sont entre trois paires de guillemets.

La fonction `afficheLesDonneesDe` a un nom suffisamment clair, les paramètres sont typés et portent des noms explicites. Un *docstring* serait redondant.

— Cinquième contact : réutilisation

Supposons que nous disposions d'une liste de triplets (prenom, nom, domaine) et que nous voulions afficher les données comme d'habitude.

On dispose donc d'une liste du type :

```
data = [("James", "Haskell", "wasps.uk"), ("James", "Bond",
"secret.uk"), ("Mata", "Hari", "spion.de"), ("Sofia",
"Loren", "cinecitta.it")]
```

Plusieurs possibilités s'offrent à nous. Nous pouvons utiliser la fonction déjà définie mais il faut faire attention au fait que l'on récupère des triplets quand on parcourt la liste alors que la fonction attend trois paramètres.

On peut faire :

```
sortie = ""
for triplet in data:
    prenom, nom, domaine = triplet
    sortie += lesDonneesDe(prenom, nom, domaine)

print(sortie)
```

On peut créer une fonction qui traite les listes de données. Pour suivre nos instructions, il faut pouvoir indiquer que le paramètre de cette fonction est une liste de triplets de chaînes de caractères. Nous avons besoin pour cela d'importer la bibliothèque `typing` qui nous permet d'utiliser des types plus complexes que les simples `str`, `int`... Ici, on va devoir créer des listes contenant des triplets de chaînes. Ce seront des `list[tuple[str, str, str]]`.

```
def tasDeDonnees(donnees: list[tuple[str, str, str]]) -> str:
    """
    donnees est une liste de triplets de la forme (prenom,
    nom, domaine)
    revoie une chaîne sous la forme :
    prenom
    nom
    pnom@domaine

    >>> data = [("James", "Haskell", "wasps.uk"),
    ("James", "Bond", "secret.uk")]
    >>> tasDeDonnees(data)
    'James\nHaskell\njhaskell@wasps.uk\nJames\nBond\njbond@
    secret.uk\n'
    """
    sortie = ""
    for triplet in donnees:
        prenom, nom, domaine = triplet
        sortie += lesDonneesDe(prenom, nom, domaine)
    return sortie

print(tasDeDonnees(data))
```


— Sixième contact : fonctions pures et impures

Une fonction doit toujours renvoyer quelque chose, même si cette chose est `None`.

Voyons `print` par exemple :

```
In [6]: type(print(2))
2
Out [6]: NoneType

In [7]: print(2)
2

In [8]: [print(k) for k in range(4)]
0
1
2
3
Out [8]: [None, None, None, None]
```

En fait, `print` est une fonction qui renvoie `None`. MAIS elle ne fait pas que ça. Si nous lisons sa documentation :

```
In [5]: ?print
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout,
flush=False)

Prints the values to a stream, or to sys.stdout by
default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the
current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a
newline.
flush: whether to forcibly flush the stream.
Type: builtin_function_or_method
```

Elle a donc un **effet secondaire** (*side effect*) en dehors de la valeur renvoyée. Ce n'est donc pas une fonction pure car elle fait quelque chose en dehors de sa signature.

Il faut, si possible, éviter ce genre de fonction car les effets secondaires sont sources de nombreux bugs. Imaginons par exemple une variable qui représente le score à un jeu durant une partie :

```
score: int = 1

def new_score(points: int) -> int:
    global score
    score = score + points
    return score
```

```
for parties in range(1,5):
    print(f"\nJ'ai gagné 10 points ! new_score(10) =
{new_score(10)}")
```

Ceci peut prêter confusion :

```
J'ai gagné 10 points ! new_score(10) = 11
J'ai gagné 10 points ! new_score(10) = 21
J'ai gagné 10 points ! new_score(10) = 31
J'ai gagné 10 points ! new_score(10) = 41
```

On remarque que les mêmes appels à `new_score(10)` ne renvoient pas la même valeur. La variable `score` est *globale* : elle est en effet définie en dehors de la fonction. Mais elle est modifiée par cette fonction : c'est un effet secondaire de l'appel de cette fonction.

Une fonction pure est une fonction qui, quand on lui passe les mêmes arguments, renverra toujours la même valeur.

Une fonction impure va changer l'état du système.

Il est donc très difficile de contrôler ce genre de fonction. On peut être tenté de les utiliser lorsqu'on veut faire évoluer un plateau de jeu par exemple. Mais il faut être conscient des risques.

— Septième contact : paramètre par défaut

Certains paramètres d'une fonction peuvent avoir le plus souvent la même valeur. On leur donne cette valeur par défaut dans la signature de la fonction.

Par exemple, reprenons la fonction `lesDonneesDe` mais supposons que nous travaillions pour une entreprise dont tous les employés ont pour nom de domaine `mafirme.fr`. Nous pouvons nous éviter de toujours rentrer le nom de domaine pour gagner du temps :

```
def lesDonneesDe(prenom: str, nom: str, domaine: str =
"mafirme.fr") -> str:
    donnees = f"{prenom}\n{nom}\n"
    donnees += f"{prenom[0].lower()}{nom.lower()}@
{domaine}"
    return donnees
```

Alors on obtient :

```
>>> print(lesDonneesDe("Ada", "Lovelace"))
Ada
Lovelace
alovelace@mafirme.fr

>>> print(lesDonneesDe("Charles", "Babbage", "machine.en"))
Charles
Babbage
cbabbage@machine.en
```

— Bilan du vocabulaire

```
def test(x: int, y: int) -> int:
    somme = 0
    for i in range(y):
        somme = somme + x
    return somme

z = test(5, 4)
```

Dans le script ci-dessus :

1. Quel est le nom de la fonction ?
2. Quels sont les paramètres (formels) de la fonction ?
3. Quels sont les arguments (ou paramètres réels) ?
4. Quelles sont les variables locales à la fonction ?
5. Quelle sera la valeur de z après l'exécution de ce script ?
6. Quelle est la signature de la fonction ?
7. Est-elle pure ?

— Correction

1. test
2. Deux paramètres formels : x et y.
3. Les paramètres réels sont 5 et 4.
4. Variables locales à la fonction : somme et i.
5. Après l'exécution de ce script, z vaudra 20.
6. (x: int, y: int) -> int
7. Oui : pas d'effets secondaires.

— Structure d'un programme

Un programme va maintenant se présenter en trois parties :

1. Une série d'importations de bibliothèques utilisées en 2.
2. Une série de fonctions qui seront utiles en 3.
3. Un programme principal qui appellera judicieusement des fonctions créées en 2.

Par exemple :

```
#1 Les importations

#2 Les fonctions
def lesDonneesDe(prenom: str, nom: str, domaine: str) -> str:
    donnees = f"{prenom}\n{nom}\n"
    donnees += f"{prenom[0].lower()} {nom.lower()}@{domaine}\n"
    return donnees
```

```

def tasDeDonnees(donnees: list[tuple[str, str, str]]) ->
str:
    """
    donnees est une liste de triplets de la forme (prenom,
nom, domaine)
    revoie une chaîne sous la forme :
    prenom
    nom
    pnom@domaine

    >>> data = [("James", "Haskell", "wasps.uk"),
("James", "Bond", "secret.uk")]
    >>> tasDeDonnees(data)
    'James\nHaskell\njhasckell@wasps.uk\nJames\nBond\njbond@
secret.uk\n'
    """

    sortie = ""
    for triplet in donnees:
        prenom, nom, domaine = triplet
        sortie += lesDonneesDe(prenom, nom, domaine)
    return sortie

#3 Le programme principal
## L'entrée
data = [("James", "Haskell", "wasps.uk"), ("James", "Bond",
"secret.uk"), ("Mata", "Hari", "spion.de"), ("Sofia",
"Loren", "cinecitta.it")]

## Le traitement
res = tasDeDonnees(data)

## La sortie
print(res)

```

6.10 Exercices

— Programmer avec des listes

Test 6.1 (Longueur d'une liste - ♥)

Déterminez une fonction qui calcule la longueur d'une liste... sans utiliser `len`... Vous donnerez une version récursive, une autre utilisant une boucle, une autre utilisant `reduce`.

Test 6.2 (Test de croissance - ♥)

Déterminez une fonction qui teste si une liste est croissante.

Test 6.3 (Informatique - Centrale 2015 - ♥)

1. Donnez la valeur des expressions suivantes :

(a) $[1, 2, 3] + [4, 5, 6]$

(b) $2 * [1, 2, 3]$

2. Écrivez une fonction **Python** `smul` à deux paramètres, un nombre et une liste de nombres, qui multiplie chaque élément de la liste par le nombre et renvoie une nouvelle liste : `smul(2, [1, 2, 3]) → [2, 4, 6]`.

3. Arithmétique des listes

(a) Écrivez une fonction **Python** `vsom` qui prend en paramètre deux listes de nombres de même longueur et qui renvoie une nouvelle liste constituée de la somme terme à terme de ces deux listes :

`vsom([1, 2, 3], [4, 5, 6]) → [5, 7, 9]`.

(b) Écrivez une fonction **Python** `vdif` qui prend en paramètre deux listes de nombres de même longueur et qui renvoie une nouvelle liste constituée de la différence terme à terme de ces deux listes :

`vdif([1, 2, 3], [4, 5, 6]) → [-3, -3, -3]`.

4. Créez une fonction d'ordre supérieure qui prendra l'opération comme paramètre. Ainsi, `vsom` sera `vop(addition)`.

Test 6.4 (Informatique - Oral Agro-Véto 2017 - ♥)

On dispose d'une matrice rectangulaire contenant des entiers de 0 à 3.

Ajouter une ligne en bas de la matrice où figureront le nombre de 2 présents dans chaque colonne et une colonne où figureront le nombre de 2 présents dans chaque ligne.

La dernière cellule en bas à droite contiendra donc le nombre total 2 de la matrice initiale.

Par exemple:

```
In [49]: m
Out[49]:
[[1, 1, 1, 1, 1, 0, 2, 3, 0, 3],
 [3, 1, 0, 3, 0, 2, 2, 2, 2, 1],
 [2, 2, 3, 1, 3, 2, 2, 3, 3, 2],
 [0, 2, 2, 1, 1, 3, 0, 3, 3, 3],
 [1, 3, 1, 1, 1, 1, 2, 0, 3, 0],
 [3, 3, 1, 3, 0, 0, 3, 3, 2, 1]]
In [50]: bords(m)
Out[50]:
[[1, 1, 1, 1, 1, 0, 2, 3, 0, 3, 1],
 [3, 1, 0, 3, 0, 2, 2, 2, 2, 1, 4],
 [2, 2, 3, 1, 3, 2, 2, 3, 3, 2, 5],
 [0, 2, 2, 1, 1, 3, 0, 3, 3, 3, 2],
 [1, 3, 1, 1, 1, 1, 2, 0, 3, 0, 1],
 [3, 3, 1, 3, 0, 0, 3, 3, 2, 1, 1],
 [1, 2, 1, 0, 0, 2, 4, 1, 2, 1, 14]]
```

Test 6.5 (Mathématiques II - CCP 2015 - ♥♥)

1. Donnez la décomposition en binaire (base 2) de l'entier 21.

On considère la fonction `mystere` suivante :

```
def mystere(n, b) :
    """Données : n > 0 un entier et b > 1 un entier
        Résultat : ..... """
    t = [ ] # tableau vide
    while n > 0 :
        c = n % b
        t.append(c)
        n = n // b
    return t
```

On rappelle que la méthode `append` ajoute un élément en fin de liste. Si l'on choisit par exemple `t = [4, 5, 6]`, alors, après avoir exécuté `t.append(12)`, la liste a pour valeur `[4, 5, 6, 12]`.

Pour $k \in \mathbb{N}^*$, on note c_k , t_k et n_k les valeurs prises par les variables `c`, `t` et `n` à la sortie de la k -ième itération de la boucle « `while` ».

2. Quelle est la valeur renvoyée lorsqu'on exécute `mystere(256, 10)` ?

On recopiera et complètera le tableau suivant en ajoutant les colonnes nécessaires pour tracer entièrement l'exécution :

k	1	2	...
c_k			...
t_k			...
n_k			...

3. Soit $n > 0$ un entier. On exécute `mystere(n, 10)`. On pose $n_0 = n$.

(a) Justifier la terminaison de la boucle `while`.

(b) On note p le nombre d'itérations lors de l'exécution de `mystere(n, 10)`. Justifiez que pour tout $k \in [0, p]$, on a $n_k \leq \frac{n}{10^k}$. Déduisez-en une majoration de p en fonction de n .

4. En vous aidant du script de la fonction `mystere`, écrivez une fonction `somme_chiffres` qui prend en argument un entier naturel et renvoie la somme de ses chiffres. Par exemple, `somme_chiffres(256)` devra renvoyer 13.

5. Écrivez une version récursive de la fonction `somme_chiffres` que vous nommerez `somme_rec`.

Test 6.6 (Banque PT 2015 - ♥♥)

Soit N un entier naturel non nul. On cherche à trier une liste L d'entiers naturels strictement inférieurs à N .

1. Écrire une fonction `comptage`, d'arguments L et N , renvoyant une liste P dont le k -ième élément désigne le nombre d'occurrences de l'entier k dans la liste L .

2. Utiliser la liste P pour en déduire une fonction `tri`, d'arguments L et N , renvoyant la liste L triée dans l'ordre croissant.

3. Tester la fonction `tri` sur une liste de 1000 entiers naturels inférieurs à 500, tirés aléatoirement puis doubler plusieurs fois la longueur de la liste et mesurer les temps d'exécution.

Test 6.7 (Banque PT 2015 - ♥♥)

Soit un entier naturel non nul n et une liste t de longueur n dont les termes valent 0 ou 1. Le but de cette question est de trouver le nombre maximal de 0 contigus dans t . Par exemple, le nombre de zéros contigus de la liste t_1 suivante vaut 4 :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$t_1[i]$	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0

1. Écrire une fonction `nombreZeros(t, i)` prenant en paramètres une liste t de longueur n et un indice i compris entre 0 et $n - 1$ et renvoyant :

$$\begin{cases} 0, & \text{si } t[i] = 1 \\ \text{le nombre de zéros consécutifs dans } t \text{ à partir de } t[i] \text{ inclus,} & \text{si } t[i] = 0 \end{cases}$$

2. Comment obtenir le nombre maximal de zéros contigus d'une liste t connaissant la liste des `nombreZeros(t, i)` pour $0 \leq i \leq n - 1$?

En déduire une fonction `nombreZerosMax(t)` renvoyant le nombre maximal de 0 contigus d'une liste t non vide. On utilisera la fonction `nombreZeros`.

3. Trouver un moyen simple, toujours en employant `nombreZeros`, d'obtenir un algorithme plus performant.

Test 6.8 (Informatique - Centrale 2016 - ♥♥)

Ce problème s'intéresse à différents aspects relatifs à la sécurité aérienne et plus précisément au risque de collision entre deux appareils.

Lors du dépôt d'un plan de vol, la compagnie aérienne doit préciser à quel niveau de vol elle souhaite faire évoluer son avion lors de la phase de croisière. Ce niveau de vol souhaité, le RFL pour *requested flight level*, correspond le plus souvent à l'altitude à laquelle la consommation de carburant sera minimale. Cette altitude dépend du type d'avion, de sa charge, de la distance à parcourir, des conditions météorologiques, etc. Cependant, du fait des similitudes entre les différents avions qui équipent les compagnies aériennes, certains niveaux de vol sont très demandés, ce qui engendre des conflits potentiels, deux avions risquant de se croiser à des altitudes proches. Les contrôleurs aériens de la région concernée par un conflit doivent alors gérer le croisement de ces deux avions. Pour alléger le travail des contrôleurs et diminuer les risques, le système de régulation s'autorise à faire voler un avion à un niveau différent de son RFL. Cependant, cela engendre généralement une augmentation de la consommation de carburant. C'est pourquoi on limite le choix aux niveaux immédiatement supérieur et inférieur au RFL.

Ce problème de régulation est modélisé par une matrice. Les valeurs de la cellule $[i][j]$ correspondent au coût d'un conflit éventuel entre le vol numéro i et le vol numéro j . Nous conviendrons que pour $0 \leq k \leq n$:

- l'indice $3k$ représente le vol v_k à son RFL ;
- l'indice $3k+1$ représente le vol v_k au-dessus de son RFL ;
- l'indice $3k+2$ représente le vol v_k au-dessous de son RFL.

S'il n'y a pas de conflit entre deux sommets, `conflit[i][j]` vaut 0. On convient que `conflit[i][j]` vaut 0 si les indices i et j correspondent au même vol.

On notera que, pour tout couple d'indices (i, j) , `conflit[i][j]` et `conflit[j][i]` représentent un seul et même conflit et donc `conflit[i][j] == conflit[j][i]`.

```
conflit = [[ 0, 0, 0, 100, 100, 0, 0, 150, 0],
            [ 0, 0, 0, 0, 0, 50, 0, 0, 0],
            [ 0, 0, 0, 0, 200, 0, 0, 300, 50],
            [100, 0, 0, 0, 0, 0, 0, 400, 0],
            [100, 0, 200, 0, 0, 0, 200, 0, 100],
            [ 0, 50, 0, 0, 0, 0, 0, 0, 0],
            [ 0, 0, 0, 400, 200, 0, 0, 0, 0],
            [150, 0, 300, 0, 0, 0, 0, 0, 0],
            [ 0, 0, 50, 0, 100, 0, 0, 0, 0]]
```

1. Écrire une fonction `nb_conflits()` sans paramètre qui renvoie le nombre de conflits potentiels.
2. Pour un vol v_k , on appelle niveau relatif l'entier r_k valant 0, 1 ou 2 tel que :
 - $r_k = 0$ représente le vol v_k à son RFL ;
 - $r_k = 1$ représente le vol v_k au-dessus de son RFL ;
 - $r_k = 2$ représente le vol v_k au-dessous de son RFL.

On appelle *régulation* la liste $(r_0, r_1, \dots, r_{n-1})$. Par exemple la régulation $(0, 0, \dots, 0)$ représente la situation dans laquelle chaque avion se voit attribuer son RFL. Une régulation sera implantée en **Python** par une liste d'entiers. Il pourra être utile d'observer que les indices de la matrice des conflits choisis par la régulation r portent les numéros $3k + r_k$ pour $0 \leq k < n$. On remarque également qu'à l'indice s correspond le niveau relatif $r_k = s[3]$ et le vol v_k tel que $k = \lfloor s/3 \rfloor$ (partie entière de $s/3$).

- (a) Écrire en **Python** une fonction `nb_vol_par_niveau_relatif(regulation)` qui prend en paramètre une régulation (liste de n entiers) et qui renvoie une liste de 3 entiers $[a, b, c]$ dans laquelle a est le nombre de vols à leurs niveaux RFL, b le nombre de vols au-dessus du RFL et c le nombre de vols au-dessous.
- (b) On appelle *coût d'une régulation* la somme des coûts des conflits potentiels que cette régulation engendre.
Écrire en **Python** une fonction `cout_regulation(regulation)` qui prend en paramètre une liste représentant une régulation et qui renvoie le coût de celle-ci.
- (c) En déduire une fonction `cout_RFL()` qui renvoie le coût de la régulation pour laquelle chaque avion vole à son RFL.

— Un peu d'arithmétique

Test 6.9 (Multiples de 3 et de 5 - ♥)

Le projet Euler¹ propose plus de 500 problèmes à résoudre dans le langage de votre choix. L'orientation est plutôt mathématique c'est pourquoi passer l'été à les résoudre est tout à fait adapté à votre situation ;-).

Nous allons explorer le premier problème pour vous allécher...

1. <https://projecteuler.net/>

Si on fait la liste de tous les entiers naturels inférieurs à 10 qui sont des multiples de 3 OU de 5, on obtient 3, 5, 6 et 9. La somme de ces multiples vaut 23.

Trouver la somme de tous les multiples de 3 ou 5 inférieurs (strictement) à 1 000.

Test 6.10 (Nombre de diviseurs - ♥)

Déterminez une fonction qui renvoie l'ensemble des diviseurs d'un entier non nul.

Test 6.11 (Crible d'ÉRATOSTHÈNE - ♥♥♥)

On considère les nombres de 2 à n . On barre tous les multiples de 2 sauf 2 puis tous les multiples du prochain nombre non barré sauf lui-même... Les nombres non barrés sont les nombres premiers inférieurs à n .

Test 6.12 (Tests de primalité - ♥♥♥♥)

Un entier naturel est dit premier s'il est supérieur à 2 et n'est divisible que par 1 et lui-même. Mais comment vérifier qu'un nombre est premier ?

1. Nous allons commencer par la force brute... Créez une fonction `les_diviseurs_brute(n)` qui renvoie l'ensemble des diviseurs de n en testant les entiers de 1 à n :

```
In [71]: les_diviseurs_brute(100)
Out[71]: {1, 2, 4, 5, 20, 25, 50, 100}
```

2. Peut-on faire moins de tests de divisibilité et avoir un algorithme plus efficace ?

```
In [337]: %timeit les_diviseurs_brute(100000000)
1 loops, best of 3: 9.84 s per loop
In [338]: %timeit les_diviseurs(100000000)
1000 loops, best of 3: 985 µs per loop
```

3. Déduisez-en un test `est_premier :: Int -> Bool`

```
In [77]: est_premier(2 ** 10 - 1)
Out[77]: False
```

4. Déterminez une fonction `prochain_premier(n)` qui renvoie le plus petit entier premier supérieur à n .
5. Nous pouvons gagner du temps en déterminant une fonction qui renvoie le plus petit diviseur (autre que 1 !) d'un nombre :

```
In [100]: ppd(2 ** 59 - 1)
Out[100]: 179951
```

Modifiez alors votre fonction `prochain_premier(n)` qui va nous faire gagner beaucoup de temps.

Test 6.13 (Complément à 2 puissance n - ♥♥♥♥)

« Le monde se sépare en 10 catégories : ceux qui comprennent cette phrase et les autres. »

Zorglub in *Les aventures de Zorglub et Martine au Far West*

Nous allons, pour favoriser la compréhension et simplifier les écritures, travailler sur 8 bits. Cela signifie que nos nombres entiers peuvent être codés avec 8 chiffres égaux à 0 ou 1 en base 2.

1. Quel est le plus grand nombre que l'on peut écrire avec 8 bits ?
2. En fait, on aimerait avoir des entiers signés, c'est-à-dire pouvoir représenter des entiers tant positifs que négatifs avec cette contrainte.
On va donc réserver un bit (le plus à gauche) pour indiquer le signe du nombre : on mettra un 0 s'il est positif et un 1 s'il est négatif.
Quel est l'ensemble des nombres que l'on peut ainsi représenter ?
3. Bon, pas de problème... enfin si quand même un peu. Que pensez-vous de ces deux nombres ? :

00000000 10000000

4. On voudrait additionner facilement ces nombres. Une idée serait d'utiliser l'addition posée comme à l'école primaire. Par exemple pour calculer $5 + 9$:

```

00000101
+ 00001001
-----
00001110

```

On trouve bien 14.

et maintenant si on veut calculer $5 + (-9)$, qu'est-ce que cela donne avec ce système ?

5. Considérez maintenant le nombre $m = 11111111$ et ajoutez-lui 1 (vous n'avez que 8 bits à disposition). Que vaut $m + 1$? Que cela vous donne-t-il comme idée ?

Considérez $n=11111110$ et calculez $n + 2$.

Considérez $n=11111100$ et calculez $n + 4$.

Considérez $n=11111000$ et calculez $n + 8$.

Considérez $n=11111010$ et calculez $n + 6$.

Des idées ?

6. Notons $\sim x$ le complément à 1 de x , i.e. le nombre obtenu en inversant les 1 et les 0 dans son écriture sur 8 bits.

Énoncez votre norme de représentation des entiers sur 8 bits.

vérifiez en calculant $5 + (-9)$.

7. Un entier en base 2 peut être introduit en **Python** en le faisant précéder de 0b :

```

In [26]: 0b1101
Out[26]: 13

```

L'opérateur $\&$ (*ET bit à bit*) entre deux entiers a et b crée un entier dont les bits sont positionnées à 1 si, et seulement si, ils sont positionnées à 1 dans a et b .

```

In [31]: 0b11 & 0b111000110101
Out[31]: 1

In [32]: 0b11 & 0b111000110111
Out[32]: 3

In [33]: 0b11 & 0b11100011011111011
Out[33]: 3

In [34]: 0b11 & 0b1110001101111101100
Out[34]: 0

```

L'opérateur \wedge (*OU EXCLUSIF bit à bit*) entre deux entiers a et b crée un entier dont les bits sont positionnées à 1 si, et seulement si, ils sont positionnées différemment dans a et b .

```

In [463]: 0b11 ^ 0b11
Out[463]: 0

In [464]: 0b11 ^ 0b10
Out[464]: 1

In [465]: 0b11 ^ 0b1
Out[465]: 2

```

L'opérateur $\gg n$ (*décalage à droite*) décale la représentation d'un nombre de n bits vers la droite (à quelle opération arithmétique cela correspond-il ?)

```

In [468]: 0b1111 >> 1
Out[468]: 7

In [469]: 0b1111 >> 2
Out[469]: 3

In [470]: 0b1111 >> 3
Out[470]: 1

In [471]: 0b1111 >> 4
Out[471]: 0

```

Imaginez alors un moyen de calculer sur des entiers 8 bits signés en **Python**.

```

In [178]: huit(256,8)
Out[178]: 0
In [179]: huit(257,8)
Out[179]: 1

In [180]: huit(127,8)
Out[180]: 127

sIn [181]: huit(128,8)
Out[181]: -128

```

```
In [182]: huit(129,8)
Out [182]: -127
```

— Simulations probabilistes

Test 6.14 (Simulations - ♥♥♥)

1. On lance trois fois de suite une pièce de monnaie. On compte combien de fois on obtient pile : simulez à l'aide de **Python**.
2. Faites de même avec l'expérience aléatoire suivante : on dispose de trois urnes, la première contenant 7 boules blanches et 4 noires, la seconde 5 blanches et 2 noires et la troisième 6 blanches et 3 noires. On tire une boule dans chaque urne et on note le nombre de boules blanches obtenues.
3. On tire successivement et avec remise quatre boules dans une urne qui contient 7 boules blanches et 3 roses. On compte le nombre de tirages contenant exactement deux boules blanches.

Test 6.15 (Oral Maths-Info Concours Agro-Véto 2015 - ♥♥♥)

Une urne contient initialement une boule blanche et une boule noire. On effectue une série de tirages aléatoires d'une boule jusqu'à obtenir une boule noire. À chaque tirage amenant une boule blanche, on replace la boule blanche puis on multiplie par 2 le nombre de boules blanches présentes dans l'urne après la remise de la boule, puis on procède au tirage suivant. Soit X la variable aléatoire égale au rang du tirage amenant une boule noire (si on obtient la boule noire), et qui vaut 0 si on n'obtient jamais de boule noire.

L'objectif de l'exercice est d'évaluer la probabilité de ne jamais obtenir de boule noire, et de déterminer en particulier si cette probabilité est nulle.

1. Compléter la fonction ci-dessous afin qu'elle réalise m fois l'expérience décrite ci-dessus (en arrêtant les tirages après l'obtention de t_{\max} boules blanches), et renvoie la proportion des expériences où une boule noire a été obtenue :

```
def EstimeProbaEchec (m, tmax) :
    CompteSucces = 0
    for i ... :
        b = ...
        succes = 0
        tirages = 0
        while succes == ... and tirages ... :
            tirages ...
            if random() ... :
                succes = 1
            else :
                ...
        CompteSucces += ...
    return ...
```

2. Utiliser cette fonction avec plusieurs valeurs de t_{\max} pour établir une conjecture en réponse au problème posé.

— Représentations graphiques

Test 6.16 (Banque PT 2015 - ♥♥♥)

On pose $M = 20$ et $m = 10$. À un nombre c quelconque, on associe la suite $(u_n)_{n \geq 0}$ définie par :

$$u_0 = 0 \quad u_{n+1} = u_n^2 + c$$

S'il existe, on note k le plus petit entier tel que l'on ait $0 \leq k \leq m$ et $|u_k| > M$.

On définit alors la fonction f par :

$$f : c \mapsto \begin{cases} k & \text{s'il existe} \\ m + \sin c & \text{sinon} \end{cases}$$

1. Donner un code **Python** définissant la fonction f .
2. Tracer l'allure de la courbe représentative de la fonction f sur $[-2, 2]$ en créant une liste `LX` de 401 valeurs équiréparties entre -2 et 2 inclus et en utilisant les fonctions `plot` et `show` de la sous-bibliothèque `matplotlib.pyplot`.
3. Construire le tableau des valeurs $f(x + iy)$ où x prend 101 valeurs comprises entre -2 et $0,5$ et y prend 101 valeurs entre $-1,1$ et $1,1$. On rappelle que le nombre complexe i est représenté par `1j`. Par exemple le complexe $1 + 2i$ est représenté par `1+2j`.
4. Tracer l'image que code ce tableau. On pourra utiliser les fonctions `imshow` et `show` de la sous-bibliothèque `matplotlib.pyplot`. Quel(s) paramètre(s) peut-on modifier pour obtenir une meilleure résolution ?

Coups de pouce

Coup de pouce pour le test 6.1

La longueur d'une liste vide est zéro. Sinon, c'est la longueur de la liste sans sa tête +1.

Coup de pouce pour le test 6.2

Une liste est croissante *a priori* mais on s'arrête dès qu'on rencontre un élément plus petit que le précédent.

Coup de pouce pour le test 6.3

1. + est l'opérateur de concaténation dans ce contexte et * est l'itération de la concaténation comme la multiplication est l'itération de l'addition des entiers.
2. Pensez à une liste par compréhension.
3. Pensez à une liste par compréhension.
4. On remarque que `vsom` et `vdifff` sont en fait une seule et même fonction au nom de l'opérateur près.

Coup de pouce pour le test 6.4

Il faut par exemple commencer par parcourir chaque ligne et ajouter le nombre de 2 pour chacune puis ajouter une colonne en bas remplie de 0 et la mettre à jour en parcourant chaque colonne.

Coup de pouce pour le test 6.5

1. Écrire un nombre en base 2 c'est déterminer $n \in \mathbb{N}$ et les $b_i \in \{0,1\}$ tels que ce nombre s'écrive $b_n \times 2^n + \dots + b_1 \times 2^1 + b_0 \times 2^0$.
2. Déterminez qui est qui : `t` est le tableau retourné qui contient les valeurs successives de `c` qui contient le reste de la division par `b` (constant) de `n` qui contient lui-même le quotient de la division par `b` de la valeur précédente de `n`.
3. (a) Quel est le sens de variation de la suite (n_k) des valeurs prises par `n` ?
(b) Une petite récurrence puis pensez au nombre de fois qu'on peut diviser un entier par 10 et appliquez le résultat précédent à $k = p - 1$ par exemple.
4. Il y a juste un tout petit détail à changer : au lieu de concaténer deux listes, on va additionner deux entiers...
5. La somme des chiffres d'un entier est la somme des chiffres de son quotient par la base plus le reste de sa division par la base.

Coup de pouce pour le test 6.6

1. Pour être efficace, il faut essayer de ne parcourir la liste `L` qu'une seule fois.
2. On parcourt la liste `P` dans l'ordre : pourquoi cela va nous aider à construire la liste triée ?

3. On utilise par exemple `%timeit`.

Coup de pouce pour le test 6.7

1. Si `t[i]` est nul, on incrémente un compteur tant que le voisin est nul.
2. On cherche le maximum de la liste des `nombreZeros(t, i)` : pensez à une liste par compréhension.
3. Pas la peine de s'occuper des indices entourés de zéros...

Coup de pouce pour le test 6.8

1. Il suffit de parcourir la moitié du tableau : le triangle strictement inférieur par exemple.
2. (a) Une des possibilités assez élégante : introduire une liste `res = [0, 0, 0]` et parcourir la régulation en accumulant dans la case adéquate.
(b) Pour chaque vol dont on connaît la régulation, il faut aller chercher son coût dans la matrice `conflit` en tenant compte de la valeur de sa régulation pour savoir quelle valeur prendre.
(c) Les avions volent à leur RFL donc la régulation est une liste remplie de 0... mais combien sont-ils ?

Coup de pouce pour le test 6.9

Il y a une solution assez évidente et naïve qui utilise l'opérateur arithmétique `%` et l'opérateur booléen `or`. Ensuite, vous pourrez chercher une solution beaucoup plus rapide qui ne fait intervenir aucune boucle... On peut en effet être plus efficace en remarquant que l'on peut faire la somme des entiers multiples de 3 plus la somme des entiers multiples de 5. On a alors des multiples en double : les nombres qui sont multiples à la fois de 3 et de 5 donc les multiples de 15 : pouvez-vous démontrer ce résultat ? Pourquoi cela va-t-il nous donner un algorithme plus efficace ?

Coup de pouce pour le test 6.10

Une ligne suffit si vous utilisez un ensemble défini par compréhension avec un filtre. Vous pouvez améliorer le temps d'exécution en remarquant que les diviseurs vont souvent deux par deux...

Coup de pouce pour le test 6.11

On peut par exemple traduire directement ce que l'on fait à la main en créant une liste remplie de booléens indiquant si le nombre correspondant à l'indice de la cellule a été rayé ou non. On sait aussi que 2 est le seul entier pair premier. On peut donc commencer ainsi :

```
def crible(n) :
    rayes = [False] * n
    valeur = 3
    prems = {2}
    while valeur < n:
        ???
    return prems
```

Coup de pouce pour le test 6.12

1. Le test de divisibilité est $n \% k == 0$
2. Si un nombre n'est pas premier, il peut s'écrire comme produit de deux nombres plus grands que 1. Quel est la valeur maximale du plus petit des deux ?
L'opérateur de la réunion de deux ensembles est $|$ (lire *pipe* i.e. « paillepe ») qui est aussi le symbole du *ou inclusif*.
3. Si un nombre est premier, combien a-t-il de diviseurs ?
4. On teste chaque suivant jusqu'à obtenir un nombre premier en utilisant le test de la question précédente.
5. Le plus petit diviseur peut être n lui-même ou alors il est plus petit qu'une certaine limite déjà évoquée.

Coup de pouce pour le test 6.13

1. Un nombre sur 8 bits s'écrit $b_7 \times 2^7 + b_6 \times 2^6 + \dots + b_0 \times 2^0$. Quel est le plus grand possible ? Et si on lui ajoute 1 ?
2. Un bit en moins c'est ? ? ? fois moins de nombres.
3. Il y a un zéro de trop dans cette ville.
4. Posez l'addition de la même manière.
5. Quand une retenue « tombe à l'eau » en allant au-delà des huit bits, on la perd...
6. Que vaut $x + \sim x$?
7. En faisant le ET bit à bit d'un nombre positif avec 11111111 on obtient sa représentation sur 8 bits.
Si le nombre est négatif, il faut calculer sa valeur absolue, appliquer le *masque* précédent pour l'avoir sur 8 bits et enfin mettre un signe négatif devant.

Coup de pouce pour le test 6.14

1. Une petite aide pour la première fonction qui servira de modèle aux suivantes :

```
def piece(n) :
    M = [sum(randint(2, size = 3))
         for k in range(n)]
    c = Counter(M)
    return {v : c[v] / n for v in c}
```

2. Tirer une boule blanche dans la première urne, c'est obtenir un nombre inférieur à 7 lorsqu'il est aléatoirement choisi selon la loi uniforme discrète sur $[[1, 11]]$. C'est aussi obtenir un nombre inférieur à 7/11 lorsqu'il est aléatoirement choisi selon la loi uniforme continue sur $[0, 1]$. On dispose de `binomial(n,k,taille)` qui renvoie un échantillon de taille `taille` d'une variable aléatoirement choisie selon la loi binomiale de paramètres n et k .
3. Loi binomiale.

Coup de pouce pour le test 6.15

`random()` renvoie un nombre aléatoirement choisi entre 0 et 1 selon une loi uniforme et la probabilité d'obtenir une noire et l'inverse du nombre total de boules.

Coup de pouce pour le test 6.16

1. Définissez une variable k et une variable u qui contiendra les valeurs successives de la suite.
2. Utilisez `linspace` pour LX et un tableau **NumPy** par compréhension pour LY .
3. Allez voir l'aide de `imshow`, `numpy.meshgrid` et `numpy.vectorize`.

Solutions et commentaires

Solution du test 6.1

Version récursive :

```
def long_r(liste):  
    if liste == [] :  
        return 0  
    else :  
        return 1 + long_r(liste[1:])
```

Version bouclée :

```
def long_b(liste):  
    l = liste  
    cpt = 0  
    while l != []:  
        l = l[1:]  
        cpt += 1
```

Version réduite : on ajoute 1 à chaque fois qu'on plie un élément

```
def long_p(liste):  
    return reduce(lambda x,y : x + 1 , liste)
```

On compare :

```
In [43]: %timeit long_p(list(range(10000)))  
1000 loops, best of 3: 1.26 ms per loop  
In [44]: %timeit long_r(list(range(10000)))  
1 loops, best of 3: 332 ms per loop  
In [46]: %timeit long_b(list(range(10000)))  
10 loops, best of 3: 177 ms per loop
```

La version réduite est bien plus rapide...

Solution du test 6.2

Une version récursive :

```
def croit_r(liste):  
    if len(liste) <= 1:  
        return True  
    elif liste[0] > liste[1]:  
        return False  
    else :  
        return croit_r(liste[1:])
```


Une version bouclée :

```
def croit_b(liste):
    i = 0
    while i < len(liste):
        if liste[i] > liste[i+1]:
            return False
        i += 1
    return True
```

Solution du test 6.3

1. (a) Concaténation : `[1,2,3,4,5,6]`.
(b) Dans ce cas, on concatène la liste deux fois de suite avec elle-même et on obtient `[1,2,3,1,2,3]`.
2. Une définition par compréhension s'impose :

```
def smul(nb: float, xs: list[float]) -> list[float]:
    return [nb * x for x in xs]
```

3. (a) On vérifie au départ que les listes sont de même taille et on crée une liste par compréhension :

```
def vsom(xs: list[float], ys: list[float]) -> list[float]:
    assert len(xs) == len(ys), 'Listes de tailles différentes'
    return [xs[i] + ys[i] for i in range(len(xs))]
```

Attention à bien faire la différence avec cette fonction :

```
def vsom2(xs: list[float], ys: list[float]) -> list[float]:
    assert len(xs) == len(ys), 'Listes de tailles différentes'
    return [x + y for x in xs for y in ys]
```

qui aura pour longueur le carré de la longueur des listes initiales :

```
In [35]: vsom([1,2,3],[4,5,6])
Out[35]: [5, 7, 9]

In [36]: vsom2([1,2,3],[4,5,6])
Out[36]: [5, 6, 7, 6, 7, 8, 7, 8, 9]
```

- (b) Même combat :

```
def vdif(xs: list[float], ys: list[float]) -> list[float]:
    assert len(xs) == len(ys), 'Listes de tailles différentes'
    return [xs[i] - ys[i] for i in range(len(xs))]
```

4. On peut bien sûr abstraire l'opérateur entre les deux listes :

```
def vop(op, xs, ys):
    assert len(xs) == len(ys), 'Listes de tailles différentes'
    return [op(xs[i], ys[i]) for i in range(len(xs))]
```

On peut alors choisir son opérateur :

```
In [41]: vop(lambda x, y : x + y, [1,2,3], [4,5,6])
Out[41]: [5, 7, 9]
In [42]: vop(lambda x, y : x - y, [1,2,3], [4,5,6])
Out[42]: [-3, -3, -3]
In [43]: vop(lambda x, y : x * y, [1,2,3], [4,5,6])
Out[43]: [4, 10, 18]
```

ou recréer vsom à partir de vop grâce à l'opérateur lambda :

```
In [44]: vsom = lambda xs, ys : vop(lambda x, y : x + y, xs, ys)
In [45]: vsom([1,2,3], [4,5,6])
Out[45]: [5, 7, 9]
```

Solution du test 6.4

```
def bords(m: list[list[int]]) -> list[list[int]]:
    nb_lignes = len(m)
    nb_cols = len(m[0])
    for ligne in m:
        sum_l = 0
        for cel in ligne:
            sum_l += cel == 2
        ligne += [sum_l]
    last_line = [0]*(nb_cols + 1)
    new_m = m + [last_line]
    for j in range(nb_cols):
        for i in range(nb_lignes):
            last_line[j] += m[i][j]==2
    last_line[nb_cols] = sum(last_line[:-1])
    return new_m
```

Solution du test 6.5

1. Quand on effectue les divisions euclidiennes successives du nombre par 2, on obtient dans l'ordre b_0, b_1, \dots, b_n . Il suffit ensuite de les réordonner sous la forme $\overline{b_n b_{n-1} \dots b_1 b_0}^2$.

Ici $21 = \overline{10101}^2$.

2. L'exécution de `mystere(256,10)` renvoie la liste `[6, 5, 2]` :

k	1	2	3
c_k	6	5	2
t_k	[6]	[6,5]	[6,5,2]
n_k	25	2	0

3. (a) Chaque nouvelle valeur de n_k est le quotient entier de la valeur précédente par 10. Ainsi la suite est définie par :

$$n_0 = n > 0 \quad n_k = 10n_{k+1}/r_k \quad 0 \leq r_k < 10$$

Tant que n_k est non nul, on a :

$$\frac{n_k}{n_{k+1}} = 10 + \frac{r_k}{r_{k+1}} > 1$$

Ainsi, tant que n_k est non nul, la suite est strictement décroissante or c'est une suite d'entiers naturels. Il existe donc une valeur de k telle que n_k est nulle la boucle termine donc.

- (b) On montre par récurrence sur k que $n_k \leq \frac{n}{10^k}$ pour tout $k \in [0, p]$.
C'est vrai pour $k = 0$.
On a $n_p = 0$ et $n_k > 0$ pour tout entier naturel k strictement inférieur à p .

Nous avons démontré que $n_{k+1} \leq \frac{n_k}{10}$ sur $[0, p-1]$.

Donc sur cet intervalle :

$$n_k \leq \frac{n}{10^k} \Rightarrow n_{k+1} \leq \frac{\frac{n}{10^k}}{10} \leq \frac{n}{10^{k+1}}$$

On en déduit le résultat.

En particulier pour $k = p-1$, on a $n_{p-1} \geq 1$ donc :

$$0 < n_{p-1} \leq \frac{n}{10^{p-1}} \Rightarrow 10^{p-1} \leq \frac{n}{n_{p-1}}$$

On compose alors par la fonction logarithme de base 10 qui est strictement croissante sur $]0, +\infty[$:

$$p-1 \leq \log_{10} n - \log_{10} n_{p-1}$$

Or $n_{p-1} \geq 1$ donc $\log_{10} n_{p-1} \geq 0$ et finalement :

$$p \leq 1 + \log_{10} n$$

ce qui n'est pas étonnant : c'est le nombre de fois qu'on peut diviser un nombre entier par 10 avant d'obtenir zéro.

4. On remplace la liste vide (l'élément neutre de la concaténation) par 0 (l'élément neutre de l'addition des entiers) et la concaténation par l'addition des entiers. On peut même travailler dans une base quelconque :

```
def som_chiffres(n: int, b: int) -> int:
    """Données : n > 0 un entier et b > 1 un entier
    Résultat : la somme des chiffres de l'écriture de n en base b"""
    s = 0 # somme nulle
    while n > 0:
        c = n % b
        s = s + c
        n = n // b
    return s
```

5. Si $n = 0$, la somme des chiffres de l'écriture en base 10 de n est nulle...

```
def som_rec(n: int, b: int) -> int:
    if n == 0 :
        return 0
    return somme_rec(n // b, b) + n % b
```

Solution du test 6.6

1. On part d'une liste P de longueur N remplie au départ de zéros. À chaque nombre k rencontré, on incrémente la valeur de P[k] :

```
def comptage(L: list[int], N: int) -> list[int]:
    P = [0] * N
    for k in L:
        P[k] += 1
    return P
```

Par exemple :

```
In [50]: from numpy.random import randint
In [52]: L = randint(0,10,20)
In [53]: L
Out[53]: array([9, 7, 7, 8, 9, 4, 4, 2, 6, 1, 8, 5, 0, 0, 6, 7, 6, 6, 5, 9])
In [54]: comptage(L,10)
Out[54]: [2, 1, 1, 0, 2, 2, 4, 3, 2, 3]
```

2. On répète P[indice] fois l'indice de la liste triée :

```
def tri(L: list[int], N: int) -> list[int]:
    P = comptage(L, N)
    L_triee = []
    for indice in range(N) :
        L_triee += [indice] * P[indice]
    return L_triee
```

Par exemple :

```
In [61]: L
Out[61]: array([9, 7, 7, 8, 9, 4, 4, 2, 6, 1, 8, 5, 0, 0, 6, 7, 6, 6, 5, 9])
In [62]: tri(L,10)
Out[62]: [0, 0, 1, 2, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 7, 8, 8, 9, 9, 9]
```

3. Comparons l'effet d'un doublement de la taille de la liste.

```
In [86]: L = randint(0,500,1000)
In [87]: %timeit tri(L,500)
1000 loops, best of 3: 238 µs per loop
In [88]: L = randint(0,500,2000)
In [89]: %timeit tri(L,500)
1000 loops, best of 3: 380 µs per loop
```

Normalement, doubler la taille devrait doubler le temps car on ne fait que des parcours simple de longueur N. **Python** doit cependant mettre en cache certains résultats lors de la répétition des appels à `tri` qui permettent d'améliorer le rendement.

Solution du test 6.7

1.

```
def nombreZeros(t: list[int], i:int) -> int:
    if t[i] == 1 :
        return 0
    cpt = 1
    while i + cpt < len(t) and t[i + cpt] == 0 :
        cpt += 1
    return cpt
```

2. On n'utilise pas la fonction `max` de **Python** : ce n'est pas l'esprit de cette épreuve...

```
def nombreZerosMax(t: list[int]) -> int:
    N = len(t)
    lesNbs = [nombreZeros(t,i) for i in range(N)]
    maxi = 0
    for nb in lesNbs :
        if nb > maxi :
            maxi = nb
    return maxi
```

3. On peut gagner du temps en « sautant » au prochain indice correspondant à une case non nulle.

```
def nombreZerosMax2(t: list[int]) -> int:
    N = len(t)
    maxi = 0
    ind = 0
    while ind < N - maxi :
        saut = nombreZeros(t,ind)
        if saut > maxi :
            maxi = saut
        ind += 1 + saut
    return maxi
```

On gagne un peu de temps en moyenne :

```
In [157]: L = randint(0,2,10000)
In [158]: %timeit nombreZerosMax(L)
100 loops, best of 3: 7.68 ms per loop
In [159]: %timeit nombreZerosMax2(L)
100 loops, best of 3: 4.21 ms per loop
```

et beaucoup de temps dans les cas pathologiques :

```
In [164]: L = [0] * 10000
In [165]: %timeit nombreZerosMax2(L)
100 loops, best of 3: 2.5 ms per loop
In [166]: %timeit nombreZerosMax(L)
1 loop, best of 3: 12.7 s per loop
```

Solution du test 6.8

1. Par exemple :

```
def nb_conflits() -> int:
    cpt = 0
    n = len(conflit)
    for i in range(n) :
        for j in range(i) :
            if conflit[i][j] > 0 : cpt += 1
            # ou bien : cpt += conflit[i][j] > 0
    return cpt
```

2. (a) La liste regulation contient des 0, 1 et 2 qui correspondent en fait aux indices de la liste res qu'il faut transformer au fur et à mesure du parcours.

```
def nb_vol_par_niveau_relatif(regulation: list[int]) -> list[int]:
    res = [0, 0, 0]
    for k in regulation:
        res[k] += 1
    return res
```

(b) Par symétrie, il suffit de lire le triangle inférieur de la matrice conflit :

```
def cout_regulation(regulation: list[int]) -> int:
    cout = 0
    for no_vol1 in range(len(regulation)):
        for no_vol2 in range(no_vol1):
            cout += conflit[3*no_vol1 + regulation[no_vol1]][3*no_vol2
                + regulation[no_vol2]]
    return cout
```

(c) Il y a `len(conflit) // 3` vols :

```
def cout_RFL() -> int:
    return cout_regulation((len(conflit) // 3) * [0])
```

Solution du test 6.9

Une première idée peut être la suivante en mettant la borne supérieure en argument pour gagner en généralité :

```
def pb1_1(borne: int) -> int:
    som = 0
    for nb in range(1, borne) :
        if (nb % 3 == 0) or (nb % 5 == 0) :
            som += nb
    return som
```

Ensuite, la syntaxe pythonnesque permet d'obtenir la même chose différemment :

```
def pb1_2(borne: int) -> int:
    som = 0
    for nb in range(1, borne) :
        som += 1 if (nb % 3 == 0) or (nb % 5 == 0) else 0
    return som
def pb1_3(borne: int) -> int:
    return sum([nb for nb in range(1, borne) if (nb % 3 == 0) or (nb % 5 == 0) ])
```

Il n'y a pas grande différence entre ces trois solutions :

```
In [26]: %timeit pb1_1(99999)
10 loops, best of 3: 16 ms per loop
In [27]: %timeit pb1_2(99999)
10 loops, best of 3: 17.8 ms per loop
In [28]: %timeit pb1_3(99999)
10 loops, best of 3: 15.8 ms per loop
```

Cependant, avec une très grande borne, ça risque d'être long.

Si l'on double la taille du nombre, on double le temps d'exécution :

```
In [23]: %timeit pb1_1(99999)
10 loops, best of 3: 16 ms per loop
In [24]: %timeit pb1_1(199999)
10 loops, best of 3: 32.7 ms per loop
In [25]: %timeit pb1_1(399999)
10 loops, best of 3: 66 ms per loop
```

Vous étudierez ce problème en prépa lors de l'étude sur la *complexité des algorithmes*.

Tirons parti de la remarque donnée dans le coup de pouce.

Mais d'abord que penser du problème : « les nombres à la fois multiples de 3 et de 5 sont les multiples de 15 » ? Cela a l'air d'être vrai pour tout le monde... quoique : 24 est un multiple de 6 et de 8 mais n'est pas un multiple de 6×8 .

Bref, il est assez clair qu'un multiple de $n \times p$ est un multiple de n et un multiple de p mais c'est moins évident dans l'autre sens.

Soit z un multiple commun de 3 et 5. Alors $z = 3k = 5\ell$. Ceux qui ont eu la chance d'avoir choisi la spécialité maths en Terminale se souviennent du lemme de GAUSS. Les autres réalisent que 3 divise 5ℓ : il y a donc « du 3 » dans 5ℓ mais comme il n'y a pas de 3 dans 5 (il vaut mieux dire que 3 et 5 sont premiers entre eux), alors il y a du 3 dans ℓ : ainsi $\ell = 3m$ et $z = 5 \times 3m$ et z est donc un multiple de 15.

Revenons à notre programme. Faisons la somme des multiples de 3 :

$$S_3 = 3 + 6 + 9 + 12 + \dots + 999 = 3(1 + 2 + 3 + \dots + 333)$$

De même :

$$S_5 = 5(1 + 2 + 3 + \dots + 199) \quad S_{15} = 15(1 + 2 + 3 + \dots + 66)$$

On reconnaît la somme des entiers successifs de 1 jusqu'au quotient entier de 999 par 3, 5 ou 15.

On en déduit le programme suivant :

```
def somme_divisibles(nb: int, borne: int) -> int:
    last = borne // nb
    return nb * (last * (last + 1)) // 2

def pb1_4(borne: int) -> int:
    return somme_divisibles(3, borne) + somme_divisibles(5, borne) -
           somme_divisibles(15, borne)
```

Non seulement le problème est traité plus rapidement mais en plus il ne semble pas être influencé par la taille du nombre en entrée :

```
In [35]: %timeit pb1_4(99999)
1000000 loops, best of 3: 1 µs per loop

In [36]: %timeit pb1_4(199999)
1000000 loops, best of 3: 1.01 µs per loop

In [37]: %timeit pb1_4(399999)
1000000 loops, best of 3: 1.01 µs per loop
```

Enfin, jusqu'à un certain point car ensuite intervient la façon dont sont représentés les entiers sur machine mais ceci est une autre histoire que vous étudierez en prépa...

Solution du test 6.10

On teste tous les entiers jusque n :

```
def ens_div1(n: int) -> set[int]:
    return {k for k in range(1, n+1) if n % k == 0}
```

```
In [8]: ens_div1(20)
Out[8]: set([1, 2, 4, 5, 10, 20])
```

On gagne du temps en s'arrêtant à $\lfloor \sqrt{n} \rfloor$ si on ajoute à notre ensemble la paire $\{k, n/k\}$. L'utilisation d'ensembles règle de plus les problèmes de doublons en cas de carrés parfaits :


```
def ens_div(n: int) -> set[int]:
    E = set([])
    for k in range(1, int(sqrt(n)) + 1):
        if n % k == 0:
            E |= {k, n//k}
    return E
```

Il n'y a pas photo : c'est dans le cas ci-dessous 40 000 fois plus rapide !...

```
In [61]: %timeit ens_div(1234567890)
100 loops, best of 3: 3.72 ms per loop

In [62]: %timeit ens_div1(1234567890)
1 loops, best of 3: 145 s per loop
```

Solution du test 6.11

```
def crible(n: int) -> set[int]:
    rayes = [False] * n
    valeur = 3
    prems = {2}
    while valeur < n:
        if not rayes[valeur]:
            prems.add(valeur)
            i = valeur
            while i < n:
                rayes[i] = True
                i += valeur
            valeur += 2
    return prems
```

On a les nombres premiers inférieurs à un million en un tiers de seconde :

```
In [65]: %timeit crible(1000000)
1 loops, best of 3: 324 ms per loop
```

Solution du test 6.12

1. On teste tous les nombres un par un :

```
def les_diviseurs_brute(n: int) -> set[int]:
    ds = set([])
    for k in range(1, n + 1) :
        if n % k == 0 :
            ds.add(k)
    return ds
```

2. On peut faire mieux.

Observons les diviseurs de 1 321 par exemple : si aucune des 1 319 divisions ne « tombe juste », alors on pourra dire que 1 321 est premier. Et puis d'abord, il est impair, il ne semble pas être divisible par 3, alors pourquoi pas ?...

diviseur	2	3	4	5	6	7	8	9	10	11	12	13
quotient	660,5	440,3	330,3	264,2	220,2	188,7	165,1	146,8	132,1	120,1	110,1	101,6
diviseur	14	15	16	17	18	19	20	21	22	23	24	25
quotient	94,36	88,07	82,56	77,71	73,39	69,53	66,05	62,90	60,05	57,43	55,04	52,84
diviseur	26	27	28	29	30	31	32	33	34	35	36	37
quotient	50,81	48,93	47,18	45,55	44,03	42,61	41,28	40,03	38,85	37,74	36,69	35,70

Nous observons de plus que, si la suite des diviseurs est croissante, celle des quotients est décroissante. Vous avez sûrement remarqué qu'à partir de 37 les quotients sont inférieurs aux diviseurs donc nous pouvons nous arrêter là : si pour un diviseur supérieur à 37, on trouvait un quotient entier q , alors en divisant 1 321 par q , qui est inférieur à 37, on devrait se trouver au début de notre liste de diviseurs et ce diviseur serait entier. Le problème, c'est qu'aucune division par un entier inférieur à 37 n'a donné de quotient entier donc on peut s'épargner les 1 283 divisions restantes.

Mine de rien, nous venons de franchir un grand pas dans la théorie des nombres : pour tester si l'entier 1 321 est premier, il nous a suffi d'effectuer 36 divisions au lieu de 1 319.

Généralisons.

Il est assez simple de démontrer que tout naturel n admet un plus petit diviseur p qui est premier.

Écartons le cas où n est premier. L'entier n étant composé, il s'écrit donc $n = pq$, avec q un entier supérieur à p (car p est le plus petit diviseur). Ainsi :

$$p \leq q \text{ et donc } p^2 \leq pq = n \text{ c'est-à-dire } p \leq \sqrt{n}$$

On peut donc améliorer notre fonction :

```
from math import sqrt, ceil
def les_diviseurs(n: int) -> set[int]:
    ds = set([])
    for k in range(1, ceil(sqrt(n)) + 1) : # + 1 car borne sup non comprise
        if n % k == 0 :
            ds |= {k, n//k} # | est l'opérateur de réunion des ensembles
    return ds
```

3. Un nombre premier n'aura donc que 2 diviseurs :

```
def est_premier1(n: int) -> bool:
    return len(les_diviseurs(n)) == 2
```

4. On sait qu'il y a une infinité de nombres premiers (célèbre démonstration à laquelle vous pouvez vous frotter...) donc on peut sans danger utiliser une boucle conditionnelle :

```
def prochain_premier(n: int) -> int:
    p = n
    while not est_premier(p) :
        p += 1
    return p
```

... mais ça peut être lent... :

```
In [412]: %timeit prochain_premier10(2 ** 50)
1 loops, best of 3: 3min 23s per loop
```

5. On peut tester les entiers impairs jusqu'à la limite fatidique de \sqrt{n} :

```
def ppd(n: int) -> int:
    if n % 2 == 0 :
        return 2
    for k in range(3, ceil(sqrt(n)) + 1, 2) :
        if n % k == 0 :
            return k
    return n
```

Un nombre premier est alors un nombre dont le plus petit diviseur (autre que 1) est lui-même :

```
def est_premier(n: int) -> bool:
    return ppd(n) == n
def prochain_premier(n: int) -> int:
    p = n
    while not est_premier(p) :
        p += 1
    return p
```

On gagne en efficacité : 2 secondes au lieu de 3 minutes...

```
In [423]: %timeit prochain_premier(2 ** 50)
1 loops, best of 3: 2.36 s per loop
```

Cependant, nous sommes loin de pouvoir tester ou générer de très grands nombres de taille « cryptographique », i.e. de l'ordre de 2 000 bits.

On pourrait alors utiliser un outil incontournable en cryptographie, le petit théorème de FERMAT qui nous permettrait de dire si un nombre est *probablement* premier, mais il faudrait savoir deux ou trois choses sur le calcul modulaire... patience !

Solution du test 6.13

1. Le plus grand nombre est 11111111. Si on lui ajoute 1 il devient 1 00000000 c'est-à-dire 2^8 .

Le plus grand nombre entier non signé sur 8 bits est donc $2^8 - 1$.

2. Il y a un bit de signe. Il reste sept bits pour la valeur absolue. On va pouvoir représenter des nombres entre 1 111 1111 et 0 111 1111, c'est-à-dire entre $-(2^7 - 1)$ et $2^7 - 1$.

3. Ces deux nombres représentent zéro : deux représentations différentes pour zéro, c'est du gâchis.

4. Posons l'addition :

```
  0000101
+ 10001001
-----
 10001110
```

Aïe, $5 + (-9) = -14$. On pourrait se débrouiller en changeant d'algorithme d'addition, mais cela compliquerait le traitement de cette opération pourtant simple.

5. Sur 8 bits $1111\ 1111 + 1$ vaut $0000\ 0000$ c'est-à-dire zéro.

Il en va de même pour tous les autres exemples proposés.

En fait, on remarque que les opposés proposés correspondent au nombre initial mais en remplaçant les zéros par des uns et vice versa puis en ajoutant 1.

6. On a $x + \sim x = 1111\ 1111$ donc $x + (\sim x + 1) = 0$. Bingo ! L'opposé d'un nombre peut donc être obtenu en ajoutant un à son complément à un.

Ainsi -9 va s'écrire :

```
x = 0000 1001
~x = 1111 0110
~x + 1 = 1111 0111
```

Calculons $5 + (-9)$:

```
  00000101
+ 11110111
-----
 11111100
```

Que vaut la **valeur absolue** du nombre obtenu ? C'est $\sim x + 1$. Il faut donc lui enlever 1 et prendre le complément à 1 du nombre obtenu :

```
y      = 11111100
y - 1  = 11111011
~(y - 1) = 00000100
```

La valeur absolue vaut 4 donc $5 + (-9)$ vaut bien -4 !

7. Par exemple :

```
def huit(n: int) -> int:
    cache = (1 << 8) - 1 # pour avoir 11111111 soit 2^8 - 1
    rep = n & cache # la troncature du nombre sur 8 bits
    signe = rep >> (8 - 1) # on extrait le signe qui est le 8e bit
    if signe == 0 :
        return rep # si n est positif, pas de transformation
    return - ((rep - 1) ^ cache) # sinon, il faut calculer sa valeur absolue
```

Des erreurs dans la représentation des nombres ont causé de graves accidents, les plus célèbres étant le missile Patriot lors de la première guerre du Golfe et le premier lancement d'Ariane V.

Solution du test 6.14

Dans tout l'exercice, on travaillera dans **iPython** avec le module **numpy.random** et la fonction **Counter** du module **collections**.

1. Redonnons le code donné en coup de pouce :

```
def piece(n: int) -> dict[int, float]:
    M = [sum(randint(2, size = 3)) for k in range(n)]
    c = Counter(M)
    return {v : c[v] / n for v in c}
```

On redéfinit le dictionnaire du « compteur » pour obtenir les fréquences de sorties. On obtient pour un million de simulations :

```
In [8]: piece(1000000)
Out [8]: {0: 0.124888, 1: 0.375004, 2: 0.375593, 3: 0.124515}
```

En effet, si on note X le nombre de « pile », alors $X \sim \mathcal{B}(3, 1/2)$.

Les probabilités de chaque sorties sont donc :

$$\mathbb{P}(X = K) = \binom{3}{k} \frac{1}{2^k} \frac{1}{2^{3-k}} = \frac{\binom{3}{k}}{8}$$

c'est-à-dire 1/8, 3/8, 3/8 et 1/8 respectivement.

2. Il s'agit de faire la somme de trois variables aléatoires indépendantes suivant des lois binomiales de paramètres 1 et respectivement 7/11, 5/7 et 6/9 :

```
def boules1(n: int) -> dict[int, float]:
    M = binomial(1, 7/11, n) + binomial(1, 5/7, n) + binomial(1, 6/9, n)
    c = Counter(M)
    return {v : c[v]/n for v in c}
```

Pour un million de simulations :

```
In [32]: boules1(1000000)
Out [32]: {0: 0.0345, 1: 0.216106, 2: 0.44601, 3: 0.303384}
```

Par exemple :

$$\mathbb{P}(X_1 = X_2 = X_3 = 1) = \frac{7}{11} \times \frac{5}{7} \times \frac{6}{9} \approx 0,303$$

3. On calcule $\mathbb{P}(X = 2)$ pour $X \sim \mathcal{B}(4, 7/10)$:

```
def boules2(n: int) -> dict[int, float]:
    M = Counter(binomial(4, 0.7, n))
    c = Counter(M)
    return {v : c[v]/n for v in c}
```

Pour dix millions de simulations :

```
In [37]: boules2(10000000)
Out [37]: {0: 0.0081178, 1: 0.0756848, 2: 0.2647574, 3: 0.411485, 4: 0.239955}
```

En effet, $C_4^2(0,7)^2 \times (0,3)^2 \approx 0,2646$.

Solution du test 6.15

1.

```
def EstimeProbaEchec (m: int, tmax: int) -> int:
    CompteSucces = 0
    for i in range(m) :
        b = 1 # b est le nb de boules blanches
        succes = 0
        tirages = 0
        while succes == 0 and tirages < tmax :
            tirages += 1
            if random() < 1 / (b + 1) :
                succes = 1
            else :
                b *= 2
        CompteSucces += succes
    return 1 - (CompteSucces / m)
```

2. On effectue plusieurs simulations et la probabilité cherchée semble tourner autour de 0,2 donc ne pas être nulle. La suite de l'exercice tend à démontrer ce résultat sans chercher à calculer explicitement la probabilité¹.

Solution du test 6.16

1.

```
M, m = 20, 10
def mandelbrot(c: float) -> int:
    k = 0
    u = 0 + 0 * 1j
    while k <= m and abs(u) <= M :
        u = u ** 2 + c
        k += 1
    if k <= m :
        return k
    return m + 1
```

2.

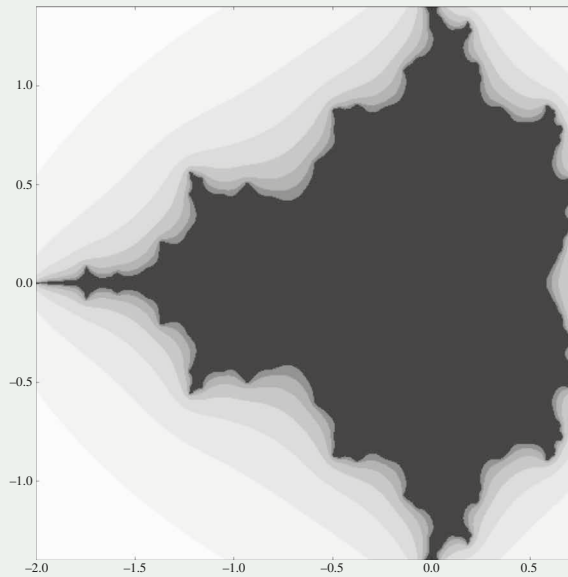
```
LX = np.linspace(-2,2,401)
LY = np.array([mandelbrot(c) for c in LX])
plt.plot(LX, LY)
plt.show()
```

1. cf. https://www.concours-agro-veto.net/IMG/pdf_15_EXOS_math_info.pdf

3. Il faut vectorialiser la fonction mandelbrot :

```
Lx = np.linspace(-2, 0.5, 501)
Ly = np.linspace(-1.1, 1.1, 501)
X,Y = np.meshgrid(Lx, Ly)
Z = X + Y * 1j
A = np.vectorize(mandelbrot)(Z)
plt.imshow(A, interpolation='none', extent=(-2, 0.75, -1.4, 1.4))
plt.show()
```

Cela donne :



Mais on peut améliorer le tracer en affinant les distinctions entre zones de convergence. Il suffit donc d'augmenter m . Voici ce qu'on obtient avec $m = 100$:

