

### Exemples d'algorithme — Programme officiel (BO)

- Méthodes des rectangles, des milieux, des trapèzes.
- Méthode de Monte-Carlo.
- Algorithme de Brouncker pour le calcul de  $\ln 2$ .

Calcul intégral

### Capacités exigibles — Programme officiel (BO)

- Calculer  $\int_a^b f(x) dx = [F(x)]_a^b$ .
- Interpréter l'intégrale comme une aire algébrique.
- Calculer une valeur moyenne.
- Intégration par parties :  $\int_a^b uv' dx = [uv]_a^b - \int_a^b u'v dx$ .

### Démonstrations exigibles — Programme officiel (BO)

- Si  $f \geq 0$  sur  $[a; b]$ , alors  $\int_a^b f(x) dx \geq 0$ .
- Inégalité de la moyenne :  $\left| \int_a^b f \right| \leq (b - a) \max_{[a;b]} |f|$ .
- Démonstration de la formule d'intégration par parties.

### Démonstrations exigibles — Programme officiel (BO)

- Espérance et variance de la loi binomiale.

### Exemples d'algorithme — Programme officiel (BO)

- Calculer la probabilité de  $(|S_n - pn| > \sqrt{n})$  où  $S_n \sim \mathcal{B}(n, p)$ ; comparer avec l'inégalité de Bienaymé-Tchebychev.
- Simulation d'une marche aléatoire.
- Simuler  $N$  échantillons de taille  $n$  d'une variable aléatoire d'espérance  $\mu$  et d'écart type  $\sigma$ ; calculer l'écart type  $s$  des moyennes observées et le comparer à  $\sigma/\sqrt{n}$ .

Loi des grands nombres

### Capacités exigibles — Programme officiel (BO)

- Calculer espérance, variance et écart-type d'une variable aléatoire.
- Comprendre et utiliser l'inégalité de Bienaymé-Tchebychev.
- Interpréter et appliquer la loi des grands nombres.

### Démonstrations exigibles — Programme officiel (BO)

- $E(X) = np$  et  $V(X) = np(1 - p)$  pour  $X \sim \mathcal{B}(n; p)$ .
- Inégalité de Bienaymé-Tchebychev :  $P(|X - E(X)| \geq \delta) \leq \frac{V(X)}{\delta^2}$ .
- Loi faible des grands nombres (admise au lycée).

# Chapitre 1

## Réurrence, Suites

### Capacités exigibles — Programme officiel (BO)

- Raisonner par récurrence pour établir une propriété d'une suite.
- Étudier des phénomènes d'évolution modélisables par une suite.

### Démonstrations exigibles — Programme officiel (BO)

- Inégalité de Bernoulli :  $(1 + a)^n \geq 1 + na$  pour  $a > 0$ , par récurrence (utilisée pour démontrer la limite de  $(q^n)$ ).

### Exemples d'algorithme — Programme officiel (BO)

- Recherche de seuils.

## A) Récurrence

### Définition 1

Une propriété mathématique est une phrase qui peut être vraie ou fausse.

### Théorème 1

Soient  $n_0$  un entier et  $P(n)$  une propriété dépendant de l'entier naturel  $n$  définie pour  $n \geq n_0$ .

On suppose :

1. **Initialisation** :  $P(n_0)$  est vraie.
2. **Hérédité** : Pour tout entier naturel  $k \geq n_0$  fixé, si  $P(k)$  est vraie alors  $P(k + 1)$  est vraie.

Alors pour tout entier naturel  $n \geq n_0$ ,  $P(n)$  est vraie.

### Exemple 1

Soit  $(u_n)$  une suite définie par  $u_0 = 3$  et pour tout entier naturel  $n$  par  $u_{n+1} = 3u_n - 2$ . Démontrer par récurrence que, pour tout entier naturel  $n$ ,  $u_n = 2 \times 3^n + 1$ .

Notons  $P(n) : \ll u_n = 2 \times 3^n + 1 \gg$ .

1. **Initialisation** : Pour  $n = 0$  on a  $2 \times 3^0 + 1 = 3$ , donc  $P(0)$  est vraie.
2. **Hérédité** : Supposons la propriété vraie pour un  $k \geq 0$  (i.e.  $u_k = 2 \times 3^k + 1$ ). On veut montrer que  $P(k + 1)$  ( $u_{k+1} = 2 \times 3^{k+1} + 1$ ) est vraie.

$$\begin{aligned}u_{k+1} &= 3u_k - 2 \\ &= 3 \times (2 \times 3^k + 1) - 2 \\ &= 2 \times 3 \times 3^k + 3 - 2 \\ &= 2 \times 3^{k+1} + 1\end{aligned}$$

Donc  $P(k + 1)$  est vraie.

3. **Conclusion** : Comme  $P(0)$  est vraie et que pour tout entier  $k$ ,  $P(k)$  vraie implique  $P(k + 1)$  vraie, la propriété  $P(n)$  est vraie pour tout  $n \in \mathbf{N}$ .

## Exemple 2

Démontrons par récurrence que, pour tout entier naturel  $n \geq 1$ , la somme des  $n$  premiers entiers est :

$$1 + 2 + \dots + n = \frac{n(n+1)}{2}.$$

1. **Initialisation** : Pour  $n = 1$  :

$$1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1.$$

La propriété est vraie pour  $n = 1$ .

2. **Hérédité** : Supposons la propriété vraie pour un entier  $k \geq 1$ , c'est-à-dire :

$$1 + 2 + \dots + k = \frac{k(k+1)}{2}.$$

Montrons qu'elle est vraie pour  $k + 1$  :

$$\begin{aligned} 1 + 2 + \dots + k + (k+1) &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2}. \end{aligned}$$

La propriété est vraie pour  $k + 1$ .

3. **Conclusion** : La propriété est vraie pour  $n = 1$  et héréditaire, donc vraie pour tout entier naturel  $n \geq 1$ .

## Exemple 3 (Contre-exemple — importance de l'initialisation)

Montrons que la proposition « pour tout entier naturel  $n \geq 1$ ,  $2^n \geq n^2$  » est **fausse** en exhibant un contre-exemple.

- Pour  $n = 1$  :  $2^1 = 2 \geq 1 = 1^2$ . Vraie.
- Pour  $n = 2$  :  $2^2 = 4 = 2^2$ . Vraie.
- Pour  $n = 3$  :  $2^3 = 8 < 9 = 3^2$ . **Fausse.**

La propriété est donc fausse dès  $n = 3$ .

**Remarque pédagogique** : même si l'hérédité est vérifiable pour  $k \geq 3$ , l'échec de l'initialisation en  $n = 3$  suffit à invalider la proposition. Cet exemple illustre l'importance de vérifier rigoureusement l'initialisation.

## B) Monotonie d'une suite

### Définition 2

Soit  $(u_n)$  une suite de nombres réels.

- La suite  $(u_n)$  est **croissante** si pour tout  $n \in \mathbf{N}$ ,  $u_n \leq u_{n+1}$ .
- La suite  $(u_n)$  est **décroissante** si pour tout  $n \in \mathbf{N}$ ,  $u_n \geq u_{n+1}$ .
- La suite  $(u_n)$  est **monotone** si elle est croissante ou décroissante.

### 1) Comment montrer qu'une suite est monotone ?

#### Méthode 1 — Signe de $u_{n+1} - u_n$

On étudie le signe de  $u_{n+1} - u_n$ .

- $u_{n+1} - u_n \geq 0$  pour tout  $n \Rightarrow$  suite croissante.
- $u_{n+1} - u_n \leq 0$  pour tout  $n \Rightarrow$  suite décroissante.
- $u_{n+1} - u_n > 0$  pour tout  $n \Rightarrow$  suite strictement croissante.
- $u_{n+1} - u_n < 0$  pour tout  $n \Rightarrow$  suite strictement décroissante.

#### Méthode 2 — Quotient $\frac{u_{n+1}}{u_n}$ (termes strictement positifs)

Si  $(u_n)$  est strictement positive :