Chapitre 3 : Suites numériques

I Généralités sur les suites

Définition
Une suite numérique (on dit aussi une suite réelle) est une fonction définie sur $\mathbb{N}$ (ou une partie de $\mathbb{N}$) et à valeurs dans $\mathbb{R}$.

Notation : $(u_n) \begin{array}{ccc} \mathbb{N} & \longrightarrow & \mathbb{R} \\ n & \longmapsto & u_n \end{array}$.

Le nombre réel u_n est appelé terme d'indice n de la suite (u_n).

Remarque (importante)
Il y a notamment deux façons de définir une suite réelle :

Par son terme général : il y a une formule qui donne u_n en fonction de n : $u_n = f(n)$.
Exemple :
Pour tout $n \in \mathbb{N}$, $u_n = n^2$.
Alors, $u_0 = 0^2 = 0$, $u_1 = 1^2 = 1$, $u_2 = 2^2 = 4$, $u_3 = 3^2 = 9$.

Par récurrence : il y a une formule qui permet de calculer un terme de la suite à partir du terme précédent (ou des termes précédents).
Par exemple, on a relation du type pour tout $n \in \mathbb{N}$, $u_{n+1} = f(u_n)$.
Il suffit de donner alors la valeur du premier terme pour que la suite soit entièrement définie.
Exemple :
On donne le premier terme $u_0 = 3$ et la relation pour tout $n \in \mathbb{N}$ $u_{n+1} = u_n - 2$.
Alors, $u_1 = u_0 - 2 = 3 - 2 = 1$, $u_2 = u_1 - 2 = 1 - 2 = -1$, $u_3 = u_2 - 2 = -1 - 2 = -3$, $u_4 = -5$ etc.

Dans tous les cas, la suite peut être définie seulement à partir d'un certain rang n_0.
Par exemple, la suite (u_n) de terme général $u_n = \sqrt{n - \pi}$ est définie pour $n \geqslant 4$.

Exercice 1 (premier exemple de suite récurrente)
On considère la suite de nombres suivants :
— le premier terme noté u_0 vaut 2.
— Chaque terme est obtenu à partir du précédent en le multipliant par deux puis en soustrayant 1.

1. Calculer u_1, u_2, u_3 et u_4.
2. Pour tout entier naturel n, proposer une relation entre u_{n+1} et u_n.
3. En s'aidant de la calculatrice, donner u_{20}, u_{30}.
4. Conjecturer une formule permettant de calculer u_n connaissant n.

Exercice 2
Dans chaque cas, calculer les premiers termes jusqu' u_4.

1. Pour tout $n \in \mathbb{N}$, $u_n = \dfrac{n^2 + 1}{n + 3}$
2. $u_0 = 4$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = (u_n)^2 - 3u_n$.
3. $u_0 = 1$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = u_n - n^2 + 1$
4. $u_0 = -2$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = (u_n)^2 - 5n$

Calcul de termes à la calculatrice
Pour une suite définie par son terme général, on peut utiliser le mode fonction et la table ajustée sur des entiers.
Le menu "suite" ou "récurrence" est surtout intéressant pour les suites définies par récurrence.

Calculatrice Casio.
Suivre le menu `recurrence`, choisir `Type`, entrer l'expression de la suite.
Les paramètres sont à adapter dans `set`.
Calculatrice TI.
Suivre `mode`, `suite`.
Pour définir la suite, suivre `f(x)`.
$nMin$ est l'indice du premier terme : 0 si le premier terme est u_0.
$u(n)$: expression qui peut dépendre de n, et/ou du terme précédent $u(n-1)$.
$u(nMin)$ est le premier terme (on ne met rien pour une suite définie par son terme général).

Exercice 3 (suite récurrente à deux pas)
On considère la suite (u_n) définie par $u_0 = 3$, $u_1 = 6$ et pour tout $n \in \mathbb{N}$, $u_{n+2} = u_{n+1} - (u_n)^2$.

1. Calculer u_2 puis u_3.
2. Utiliser la calculatrice pour donner u_{10}.

II Rappel sur les boucles bornées en Python

Boucle bornée en Python
Soient d, n des entiers naturels, $(d < n)$.
L'instruction `for k in range(n)` fait parcourir à la variable k les entiers de 0 à $(n-1)$. Il y a alors n tours de boucle.
L'instruction `for k in range(d,n)` fait parcourir à k les entiers de d à $n-1$. Il y a alors $(n-d)$ tours de boucle.

Remarque
En particulier, l'instruction "Pour k variant de 1 à n" se traduit en Python par "`for k in range(1,n+1):`"

Exercice 4
Compléter le tableau.

Instruction Python	Valeurs prises par k	Nombre de tours
`for k in range(8)`		
`for k in range(1,16)`		
`for k in range(0,6)`		
`for k in range(2,11)`		
	5,6,7,8	
	11, 12, ..., 39	

III Algorithmes de calcul de terme et de somme de termes

III.1 Algorithme de calcul de terme d'une suite récurrente

Exercice 5
Soit (u_n) la suite définie sur $\mathbb{N}$ par $\begin{cases} u_0 = 5 \\ u_{n+1} = 2u_n + 6 \end{cases}$

1. Écrire un algorithme qui renvoie le nombre u_n lorsque l'on entre l'entier n.

Langage courant	Programme CASIO	Programme TEXAS
`Entrer N`	`"N=" ?→ N`	`Prompt N`
`U ← 5`	`5 → U`	`5 → U`
`Pour K allant de 1 à N`	`For 1 → K to N`	`For (K,1,N)`
`    U ← 2U+6`	`2U+6 → U`	`2U+6 → U`
`FinPour`	`Next`	`End`
`Afficher U`	`U ◢`	`Disp U`

2. Compléter la fonction Python d'argument n qui renvoie u_n pour un entier n donné.

```
def Terme(n) :
    u=...
    for k in range(...,...):
        ...
    return(u)
```

3. Programmer la fonction et donner u_{10}.
On obtient $u_{10} = 11\,258$.

Exercice 6 (variante)
Reprendre l'exercice précédent avec la suite (u_n) définie sur $\mathbb{N}^*$ par $\begin{cases} u_1 = -3 \\ u_{n+1} = -\dfrac{3}{2}u_n + 5 \end{cases}$

III.2 Algorithme de calcul de la somme des termes consécutifs d'une suite

Exercice 7
On reprend la suite : $\begin{cases} u_0 = 5 \\ u_{n+1} = 2u_n + 6 \end{cases}$. Pour tout entier n donné en entrée ($n \geqslant 1$), on cherche à obtenir $S_n = u_0 + u_1 + \cdots + u_n = \displaystyle\sum_{k=0}^{n} u_k$.

1. On donne l'algorithme en langage courant. Compléter la fonction Python associée.

Langage courant

```
Entrer N
U ← 5
S ← U
Pour K allant de 1 à N
    U ← 2U+6
    S ← S+U
FinPour
Afficher S
```

Fonction Python

```
def Somme(n) :
    ............
    ............
    ............
    ............
    ............
    ............
```

2. Programmer et donner $S_{10} = u_0 + u_1 + \cdots + u_{10}$.
On obtient $S_{10} = 22\,451$.

Exercice 8 (variantes)
Écrire une fonction Python qui détermine la somme S_n des premiers termes jusqu'à u_n inclus.

1. Pour tout entier naturel n, $u_n = 2 + 5n$.

   Pour tout $n \geqslant 0$, on pose $S_n = u_0 + u_1 + \cdots + u_n = \displaystyle\sum_{k=0}^{n} u_k$.

2. $\begin{cases} u_3 = -1 \\ \text{Pour tout } n \geqslant 3, u_{n+1} = -2u_n + 6 \end{cases}$

   On cherche à obtenir $S_n = u_3 + u_4 + \cdots + u_n = \displaystyle\sum_{k=3}^{n} u_k$ pour un entier $n \geqslant 4$ donné.