Suites et séries

■ Comment manipuler une suite ?

SF1 Définir explicitement une suite

Le cas d'une suite définie explicitement est en tout point semblable à celui d'une fonction.

Exemple

Définissons la suite u donnée, pour tout $n \in \mathbb{N}^*$, par $u_n = \frac{1}{n^2+1}$.

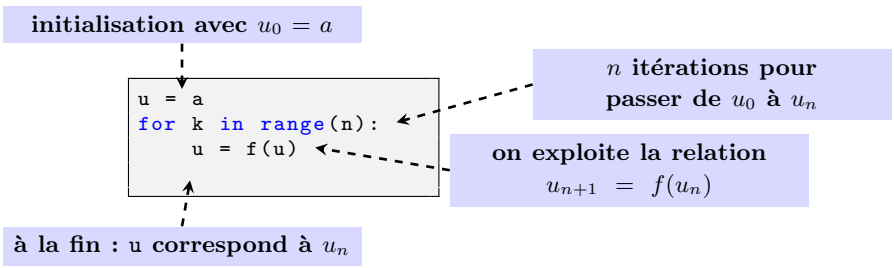
```
def u(n):
    return 1/(n**2+1)
```

En effet :

```
>>> u(2)
0.2
>>> u(5)
0.038461538461538464
```

SF2 Définir une suite par récurrence

Considérons le cas d'une suite définie par une valeur initiale $u_0 = a$ et une relation : $\forall n \in \mathbb{N},\ u_{n+1} = f(u_n)$. On définit une variable initialisée à a puis, au sein d'une boucle `for`, si l'on connaît à l'avance le nombre d'itérations (ou d'une boucle `while` si le nombre d'itérations est conditionné par un test), on remplace successivement les valeurs de la variable par les termes de la suite.



Exemples

1 ► Considérons la suite u définie par $u_0 = 2$ et : $\forall n \in \mathbb{N},\ u_{n+1} = \sqrt{1+u_n}$.

Écrivons une fonction d'en-tête `u(n)` qui renvoie la valeur de u_n.

```
def u(n):
    v = 2
    for k in range(n):
        v = sqrt(1 + v)
    return v
```

Ici, `u` est le nom de la fonction alors que `v` n'est qu'une variable locale à l'intérieur de la fonction.

```
>>> u(2)
1.6528916502810695
>>> v
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'v' is not defined
```

2 ► Considérons la suite de Fibonacci définie par $F_0 = 0$, $F_1 = 1$ et : $\forall n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$.
On vérifie que cette suite diverge vers $+\infty$.
Déterminons le premier terme de cette suite qui soit supérieur à 1000.

```
F = [0, 1]  # il est plus efficace de travailler avec deux termes successifs
while F[1] < 1000:
    s = F[1]
    t = F[0] + F[1]
    F = [s, t]
```

On trouve :

```
>>> F[1]
1597
```

SF3 Donner les premiers termes d'une suite

◇ Dans le cas d'une suite $\left(f(n)\right)_{n\in\mathbb{N}}$ donnée explicitement à l'aide d'une fonction f, on peut directement calculer l'image du vecteur correspondant aux entiers de 0 à n.

Exemple

Définissons une liste contenant les 10 premiers termes de la suite $\left(\frac{1}{n+1}\right)_{n\in\mathbb{N}}$.

```
>>> [1/(n+1) for n in range(10)]
[1.0, 0.5, 0.3333333333333333, 0.25, 0.2, 0.16666666666666666,
    0.14285714285714285, 0.125, 0.1111111111111111, 0.1]
```

On peut également utiliser une boucle en modifiant une liste existante :

```
L = [0]*10
for n in range(10):
    L[n] = 1/(n+1)
```

◇ Dans le cas d'une suite donnée par une relation de récurrence, il suffit de construire pas à pas une liste en exploitant les derniers termes ajoutés ou bien de l'initialiser à la bonne taille puis de modifier ses termes.

Exemple

Définissons une fonction d'en-tête `F(n)` renvoyant la liste des termes d'indice 0 à n de la suite de Fibonacci.

```
def F(n):
    L = [0, 1]
    for k in range(2, n):
        L.append(L[-1] + L[-2])
    return L

def Fbis(n):
    L = [0]*n
    L[1] = 1
    for k in range(2,n):
        L[k] = L[k-1] + L[k-2]
    return L
```

Ce qui donne :

```
>>> F(10)
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
>>> Fbis(10)
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Exercice 1

On considère la suite donnée par :

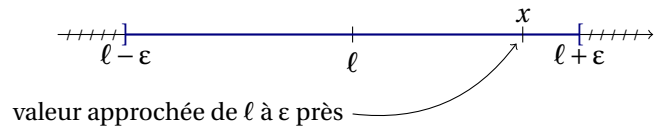
$$\begin{cases} u_0 = 1 \\ u_{n+1} = \dfrac{1}{2}\left(u_n + \dfrac{2}{u_n}\right) \quad \text{pour } n \geqslant 0 \end{cases}.$$

Écrire une fonction d'argument n et renvoyant le vecteur constitué par $u_0, u_1, \ldots, u_n$.

■ Comment obtenir une valeur approchée d'une limite de suite ?

SF4 Comprendre le principe d'une approximation

Le principe de toute approximation réside sur le fait que, pour tout réel $\varepsilon > 0$, tout réel x dans l'intervalle $]\ell - \varepsilon, \ell + \varepsilon[$ vérifie $|\ell - x| < \varepsilon$ donc fournit une approximation de ℓ à ε près (c'est-à-dire avec une erreur d'au plus ε).



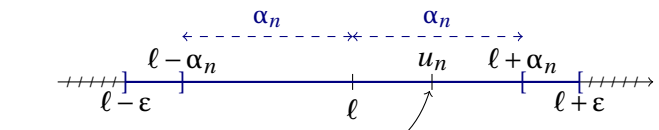
valeur approchée de ℓ à ε près

SF5 Exploiter une majoration de l'erreur

Considérons une suite $(u_n)_{n\in\mathbb{N}}$ et un réel ℓ tels qu'il existe une suite $(\alpha_n)_{n\in\mathbb{N}}$ vérifiant :

$$\forall n \in \mathbb{N},\ |u_n - \ell| \leqslant \alpha_n \quad \text{et} \quad \alpha_n \xrightarrow[n\to+\infty]{} 0.$$

Il résulte tout d'abord du théorème d'existence de limite par encadrement que la suite $(u_n)_{n\in\mathbb{N}}$ est convergente de limite ℓ.
Par ailleurs, pour tout réel $\varepsilon > 0$, **il suffit** que $\alpha_n < \varepsilon$ pour que u_n fournisse une valeur approchée de ℓ à ε près.



valeur approchée de ℓ à ε près

Exemple

Considérons la suite $(x_n)_{n\in\mathbb{N}}$ définie par $x_0 = 7$ et : $\forall n \in \mathbb{N},\ x_{n+1} = \dfrac{x_n - 1}{\ln(x_n) - 1}$.
On montre que la suite $(x_n)_{n\in\mathbb{N}}$ est convergente et, en notant ℓ sa limite et à l'aide de l'inégalité des accroissements finis, on obtient :

$$\forall n \in \mathbb{N},\ |x_n - \ell| \leqslant (0{,}17)^n.$$

Déduisons-en une fonction d'en-tête `approx(eps)` donnant une valeur approchée de ℓ à `eps` près.

```
def f(x):
    return (x-1)/(log(x)-1)

def approx(eps):
    x = 7
    n = 0
    while (0.17)**n > eps:
        x = f(x)
        n = n+1
    return x
```

```
def approxbis(eps): # alternative
    x = 7
    maj = 1
    n = 0
    while maj > eps:
        x = f(x)
        n = n+1
        maj = maj * 0.17
    return x
```

On trouve par exemple :

```
>>> approx(1e-5)
6.305395279271691
```

En fonction de l'expression de α_n, il arrive que l'on puisse explicitement déterminer un entier n tel que $\alpha_n < \varepsilon$ auquel cas on peut ensuite utiliser une boucle `for`.

Exemple

Reprenons l'exemple précédent pour illustrer cette idée.
On a :

$$\begin{aligned}(0,17)^n \leqslant \varepsilon &\iff n\ln(0,17) \leqslant \ln(\varepsilon)\\ &\iff n \geqslant \frac{\ln(\varepsilon)}{\ln(0,17)}.\end{aligned}$$

On en déduit le programme suivant :

```
def approxter(eps):
    n = ceil(log(eps) / log(0.17))
    x = 7
    for k in range(n):
        x = f(x)
    return x
```

SF6 Utiliser des suites adjacentes

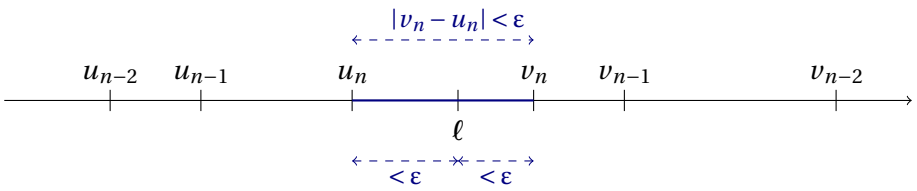
On rappelle que deux suites sont dites *adjacentes* lorsque l'une est croissante, l'autre décroissante et l'écart entre les deux converge vers 0.
Pour fixer les idées, considérons le cas d'une suite $(u_n)_{n\in\mathbb{N}}$ croissante et d'une suite $(v_n)_{n\in\mathbb{N}}$ décroissante telles que :

$$|v_n - u_n| \xrightarrow[n\to+\infty]{} 0.$$

Dans ces conditions, les deux suites convergent et leur limite est la même. De plus, en notant ℓ cette limite, on a :

$$\forall n \in \mathbb{N},\ u_n \leqslant \ell \leqslant v_n.$$

Il s'ensuit que, pour tout réel $\varepsilon > 0$, il suffit que $|v_n - u_n| < \varepsilon$ pour que tout réel entre u_n et v_n fournisse une valeur approchée de ℓ à ε près.



Exemple

On définit les deux suites u et v par $u_0 = 1$, $v_0 = \sqrt{2}$ et :

$$\forall n \in \mathbb{N},\ u_{n+1} = \frac{u_n + v_n}{2} \quad \text{et} \quad v_{n+1} = \sqrt{u_n v_n}.$$

On admet que ces deux suites sont adjacentes.
Déduisons-en une fonction d'en-tête `approx(eps)` renvoyant une valeur approchée de la limite commune de ces deux suites à `eps` près.

```
def approx(eps):
    u = 1
    v = sqrt(2)
    while abs(v-u) > eps:
        u, v = (u+v)/2, sqrt(u*v)
    return (u+v)/2
```

On obtient par exemple :

```
>>> approx(1e-5)
1.1981402347355923
```

Exercice 2

Soit u la suite donnée par $u_0 = -1$ et :

$$\forall n \in \mathbb{N},\ u_{n+1} = \frac{e^{u_n} - 3}{2}.$$

L'étude mathématique de cette suite montre qu'elle converge vers un réel ℓ entre -2 et -1 et que l'on a :

$$\forall n \in \mathbb{N},\ |u_n - \ell| \leqslant \frac{1}{2^n}.$$

Écrire une fonction d'argument p et qui donne une valeur approchée de ℓ à 10^{-p} près.

Exercice 3

On définit les trois suites a, b et p par $a_0 = \frac{1}{3\sqrt{3}}$, $b_0 = 2a_0$ et, pour tout $n \in \mathbb{N}$, par les relations :

$$a_{n+1} = \frac{a_n + b_n}{2}\ ,\ b_{n+1} = \sqrt{a_{n+1} b_n}\ \text{ et }\ p_n = \frac{1}{b_n}.$$

Une étude mathématique, montre que :

$$\forall n \in \mathbb{N},\ 0 \leqslant \pi - p_n \leqslant \frac{3\sqrt{3}}{4^n}.$$

À l'aide de cette majoration, écrire une fonction d'argument p et qui donne une valeur approchée de π à 10^{-p} près.

Exercice 4

On considère les suites u et v donnée par $u_0 = 2$, $v_0 = 1$ et pour tout $n \in \mathbb{N}$:

$$u_{n+1} = \frac{u_n + \lambda v_n}{1+\lambda} \quad \text{et} \quad v_{n+1} = \frac{u_n + \mu v_n}{1+\mu},$$

où $\lambda = 0,05$ et $\mu = 4$.

1. Écrire un programme permettant d'obtenir les valeurs de u_n et v_n pour $n \in [\![0, 10]\!]$.

2. En admettant que ces deux suites sont adjacentes, déterminer le plus petit entier n tel que :

$$|u_n - v_n| \leqslant 10^{-4}.$$

3. Définir le vecteur `U` dont les composantes sont $u_0, u_1, \ldots, u_{50}$.

4. Représenter graphiquement $u_0, u_1, \ldots, u_{50}$.

5. Représenter sur le même dessin $u_0, u_1, \ldots, u_{50}$ et $v_0, v_1, \ldots, v_{50}$.

6. Mêmes questions avec $\lambda = 0,1$ et $\mu = 6$.

Exercice 5

Soit u la suite donnée par $u_0 = \frac{1}{5}$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = f(u_n)$, où f est la fonction définie sur $\mathbb{R}$ par :

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

1. Programmer la fonction f.

2. On peut montrer, à l'aide de l'inégalité des accroissements finis, que la suite u converge vers une limite ℓ et que l'on a pour tout $n \in \mathbb{N}^*$:

$$|u_n - \ell| \leqslant \left(\frac{1}{4}\right)^{n-1}.$$

Déterminer un entier n tel que u_n fournisse une valeur approchée de ℓ à 10^{-10} près. Afficher n et le terme u_n correspondant.

■ Comment manipuler une série ou un produit infini ?

SF7 Calculer une somme partielle d'une série

Le principe du calcul d'une somme partielle $S_n = \sum_{k=0}^{n} u_k$ est celui du calcul d'une suite définie par une relation de récurrence puisque l'on a $S_0 = u_0$ et, pour tout $n \in \mathbb{N}$, $S_{n+1} = S_n + u_{n+1}$.
Il convient donc d'initialiser une variable S à 0 puis de calculer S_n à l'aide d'une boucle.
On peut également initialiser `S` avec u_0 puis n'effectuer que n itérations au lieu de $n+1$. Il convient bien entendu d'adapter également le nombre d'itérations si la somme ne débute pas à l'indice 0.

Exemple

Définissons une fonction d'en-tête `somme(n)` renvoyant $\sum_{k=1}^{n} \frac{1}{k^2}$.

Si l'on cherche à déterminer la première somme partielle inférieure ou supérieure à une valeur donnée alors on ne connaît pas à l'avance le nombre d'itérations donc on doit utiliser une boucle `while`.

Exemple

On a vu que : $\sum_{k=1}^{n} \frac{1}{k} \xrightarrow[n\to+\infty]{} +\infty$.

Déterminons le plus petit entier $n \geqslant 1$ tel que $\sum_{k=1}^{n} \frac{1}{k} > 10$.

SF8 Calculer un «produit partiel»

Le calcul d'un produit partiel $P_n = \prod_{k=0}^{n} u_k$ est semblable à celui d'une somme partielle si ce n'est que les sommes sont remplacées par des produits et que l'on initialise avec la valeur 1 au lieu de 0.

Exemple

Définissons une fonction d'en-tête `produit(n)` renvoyant $\prod_{k=1}^{n} \ln\left(1+\frac{1}{k^2}\right)$.

SF9 Obtenir une valeur approchée d'une somme de série

Considérons une série de terme général $(u_n)_{n\in\mathbb{N}}$ et notons pour tout $n \in \mathbb{N}$:

$$S_n = \sum_{k=0}^{n} u_k.$$

Dans le cas où la série est convergente, de somme $S = \sum_{k=0}^{+\infty} u_k$, on peut définir le reste d'ordre $n \in \mathbb{N}$:

$$R_n = \sum_{k=n+1}^{+\infty} u_k = S - S_n$$

et la suite $(R_n)_{n\in\mathbb{N}}$ converge donc vers 0.
Si l'on dispose d'une majoration du reste d'ordre n, alors on est dans une situation vue dans le cadre des suites.

Exemple

La série de terme général $\frac{1}{n!}$ converge et a pour somme $\sum_{k=0}^{+\infty} \frac{1}{k!} = e$.

Posons, pour tout entier $n \geqslant 1$: $S_n = \sum_{k=0}^{n} \frac{1}{k!}$.

On montre que l'on a : $\forall n \in \mathbb{N}^*,\ |e - S_n| \leqslant \frac{2}{n!}$.

Écrivons une fonction d'en-tête `approx(eps)` renvoyant une valeur approchée de e à `eps` près.

Considérons le cas particulier où u_n est de la forme $u_n = (-1)^n a_n$ avec $(a_n)_{n\in\mathbb{N}}$ une suite décroissante de limite nulle (donc une suite positive ou nulle). D'après le critère spécial de convergence des séries alternées (hors programme mais connu), la série de terme général u_n est dans ce cas convergente.
Pour démontrer ce résultat, on montre en fait que les suite $(S_{2n})_{n\in\mathbb{N}}$ et $(S_{2n+1})_{n\in\mathbb{N}}$ sont deux suites adjacentes.
On se trouve donc dans une situation connue pour les suites.

Exemple

La série de terme général $\dfrac{(-1)^{n+1}}{n}$ converge et a pour somme $\displaystyle\sum_{k=1}^{+\infty} \frac{(-1)^{k+1}}{k} = \ln(2)$.

Posons, pour tout entier $n \in \mathbb{N}$: $S_n = \displaystyle\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k}$.

Dans ces conditions, on vérifie que les suites $\left(\displaystyle\sum_{k=1}^{2n} \frac{(-1)^{k+1}}{k}\right)_{n\in\mathbb{N}}$ et $\left(\displaystyle\sum_{k=1}^{2n+1} \frac{(-1)^{k+1}}{k}\right)_{n\in\mathbb{N}}$ sont adjacentes.

Écrivons une fonction d'en-tête `approx(eps)` renvoyant une valeur approchée de $\ln(2)$ à `eps` près.

Exercice 6

Pour tout entier $n \in \mathbb{N}^*$, on pose :

$$S_n = \sum_{k=1}^{n} \frac{1}{k^2}.$$

1. Écrire une fonction d'argument $n \in \mathbb{N}^*$ et renvoyant la valeur de S_n.

2. En supposant qu'un tel entier existe, déterminer le plus petit entier n tel que :

$$S_n > 1,64.$$

Exercice 7

Pour tout entier $n \in \mathbb{N}^*$, on pose :

$$S_n = \sum_{k=1}^{n} \frac{1}{k^{\frac{5}{4}}}.$$

1. Écrire une fonction d'argument $n \in \mathbb{N}^*$ et renvoyant la valeur de S_n.

2. En supposant qu'un tel entier existe, déterminer le plus petit entier n tel que :

$$S_n > 3,5.$$

Exercice 8

Pour tout entier $n \in \mathbb{N}^*$, on pose :

$$P_n = \prod_{k=1}^{n} \left(1 + \frac{1}{k^2}\right).$$

Écrire une fonction d'argument $n \in \mathbb{N}^*$ et renvoyant la valeur de P_n.

Exercice 9

Pour tout entier $n \in \mathbb{N}^*$, on pose : $S_n = \sum_{k=0}^{n} \frac{4(-1)^k}{2k+1}$.

1. Montrer que les suites (S_{2n}) et (S_{2n+1}) sont adjacentes et en déduire que la série considérée converge.

2. Écrire une fonction d'argument p qui donne une valeur approchée à 10^{-p} près de $\sum_{k=0}^{+\infty} \frac{4(-1)^k}{2k+1}$.

Exercice 10

On a vu en exercice que :

$$P_n = \prod_{k=0}^{2n-1} \left(2 - \frac{k}{2n}\right) \xrightarrow[n\to+\infty]{} +\infty.$$

Déterminer le plus petit indice n tel que $P_n > 100$.