

Principaux domaines abordés : Suites numériques. Algorithmique et programmation.

On s'intéresse au développement d'une bactérie.

Dans cet exercice, on modélise son développement avec les hypothèses suivantes : cette bactérie a une probabilité 0,3 de mourir sans descendance et une probabilité 0,7 de se diviser en deux bactéries filles.

Dans le cadre de cette expérience, on admet que les lois de reproduction des bactéries sont les mêmes pour toutes les générations de bactéries qu'elles soient mère ou fille.

Pour tout entier naturel n , on appelle p_n la probabilité d'obtenir au plus n descendances pour une bactérie.

On admet que, d'après ce modèle, la suite (p_n) est définie de la façon suivante :

$p_0 = 0,3$ et, pour tout entier naturel n ,

$$p_{n+1} = 0,3 + 0,7p_n^2.$$

1. La feuille de calcul ci-dessous donne des valeurs approchées de la suite (p_n)

- (a) Déterminer les valeurs exactes de p_1 et p_2 (masquées dans la feuille de calcul) et interpréter ces valeurs dans le contexte de l'énoncé.

- (b) Quelle est la probabilité, arrondie à 10^{-3} près, d'obtenir au moins 11 générations de bactéries à partir d'une bactérie de ce type ?

- (c) Formuler des conjectures sur les variations et la convergence de la suite (p_n) .

2. (a) Démontrer par récurrence sur n que, pour tout entier naturel n , $0 \leq p_n \leq p_{n+1} \leq 0,5$.

- (b) Justifier que la suite (p_n) est convergente.

3. On appelle L la limite de la suite (p_n) .

- (a) Justifier que L est solution de l'équation

$$0,7x^2 - x + 0,3 = 0$$

- (b) Déterminer alors la limite de la suite (p_n) .

	A	B
1	n	p_n
2	0	0,3
3	1	
4	2	
5	3	0.407,695,62
6	4	0.416,351
7	5	0.421,343,71
8	6	0.424,271,37
9	7	0.426,004,33
10	8	0.427,035,78
11	9	0.427,651,69
12	10	0.428,020,18
13	11	0.428,240,89
14	12	0.428,373,18
15	13	0.428,452,51
16	14	0.428,500,09
17	15	0.428,528,63
18	16	0.428,545,75
19	17	0.428,556,02

1. La fonction suivante, écrite en langage Python, a pour objectif de renvoyer les n premiers termes de la suite (p_n) .

```

1 def suite(n) :
2     p= ...
3     s=[p]
4     for i in range (...) :
5         p=...
6         s.append(p)
7     return (s)

```

Recopier, sur votre copie, cette fonction en complétant les lignes 2, 4 et 5 de façon à ce que la fonction `suite (n)` retourne, sous forme de liste, les n premiers termes de la suite.