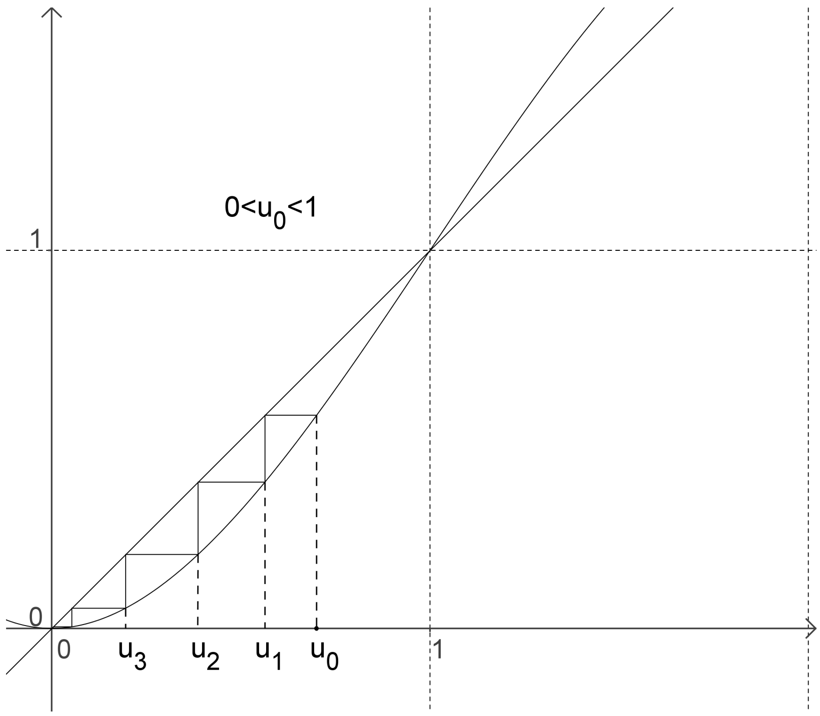
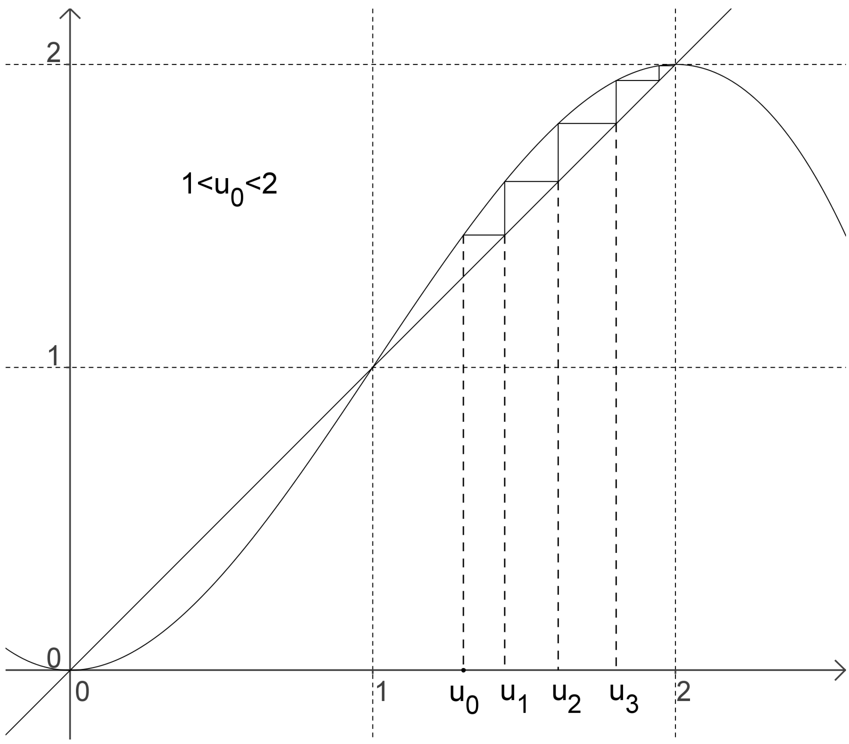
Correction du devoir de Mathématiques n°2

III **1)** • Lorsque $0 < u_0 < 1$, la suite (u_n) semble décroissante et converger vers 0 :



• Lorsque $1 < u_0 < 2$, la suite (u_n) semble croissante et converger vers 2 :



• Si $u_0 = 0$ (resp. $u_0 = 1$; resp $u_0 = 2$), la suite semble constante égale à 0 (resp. 1 ; 2).

2)a) Notons pour $n \in \mathbb{N}$, $\mathscr{P}(n)$ la proposition : $u_n \in [0;1]$.

• **Initialisation** : On a $u_0 = 0,5 \in [0;1]$ donc $\mathscr{P}(0)$ est vraie.

• **Hérédité** : Supposons que $\mathscr{P}(n)$ soit vraie pour un certain $n \in \mathbb{N}$; on a donc $0 \leqslant u_n \leqslant 1$. Or on sait d'après 1) que f est croissante sur $[0;2]$; on peut donc en déduire que : $f(0) \leqslant f(u_n) \leqslant f(1)$. Comme $f(0) = 0$, $f(1) = -\frac{1}{2} + \frac{3}{2} = 1$ et $u_{n+1} = f(u_n)$, cela implique que $0 \leqslant u_{n+1} \leqslant 1$ et $\mathscr{P}(n+1)$ est donc vraie.

• On a donc prouvé par récurrence que $\mathscr{P}(n)$ était vraie pour tout $n \in \mathbb{N}$.

2)b) Notons pour $n \in \mathbb{N}$, $\mathscr{R}(n)$ la proposition : $u_{n+1} \leqslant u_n$.

• **Initialisation** : On a $u_1 = -\frac{1}{2}u_0^3 + \frac{3}{2}u_0^2 = -\frac{1}{2} \times 0,5^3 + \frac{3}{2} \times 0,5^2 = 0,3125$.
Ainsi $u_1 \leqslant u_0$, ce qui établit que $\mathscr{R}(0)$ est vraie.

• **Hérédité** : Supposons que $\mathscr{R}(n)$ soit vraie pour un certain $n \in \mathbb{N}$; on a donc $u_{n+1} \leqslant u_n$. Or d'après 3)a), on sait que u_n et u_{n+1} sont dans $[0;1]$. On peut donc appliquer la croissance de la fonction f sur $[0;1]$ $\big($on sait d'après 1) que f est croissante sur $[0;2]$ $\big)$ et déduire que $f(u_{n+1}) \leqslant f(u_n)$, ce qui prouve que $u_{n+2} \leqslant u_{n+1}$: $\mathscr{R}(n+1)$ est donc vraie.

• On a donc montré par récurrence que $\mathscr{R}(n)$ était vraie pour tout $n \in \mathbb{N}$: la suite (u_n) est donc décroissante.

2)c) La suite (u_n) étant décroissante, on a pour tout $n \in \mathbb{N}$: $u_n \leqslant u_0 = \frac{1}{2}$. Comme $u_n \geqslant 0$, on en déduit que $u_n \times \frac{1}{2}u_n \leqslant \frac{1}{2} \times \frac{1}{2}u_n$, d'où : $0 \leqslant \frac{1}{2}u_n^2 \leqslant \frac{1}{4}u_n$ **(1)** .
D'autre part $0 \leqslant u_n \leqslant 1$ donc $-1 \leqslant -u_n \leqslant 0$ puis $0 < 2 \leqslant 3 - u_n \leqslant 3$ **(2)** .
Comme elles mettent en jeu des termes positifs, on peut faire le produit membre à membre des inégalités **(1)** et **(2)** : $\frac{1}{2}u_n^2 \times (3 - u_n) \leqslant \frac{1}{4}u_n \times 3$, soit : $u_{n+1} = \frac{1}{2}u_n^2(3 - u_n) \leqslant \frac{3}{4}u_n$.

2)d) Soit $n \in \mathbb{N}$. On a déjà vu au 3)a) que $0 \leqslant u_n$. Notons $\mathscr{B}(n)$ la proposition $u_n \leqslant \left(\frac{3}{4}\right)^n$.

• **Initialisation** : On a $u_0 = \frac{1}{2}$ et $\left(\frac{3}{4}\right)^0 = 1$ donc $u_0 \leqslant \left(\frac{3}{4}\right)^0$: $\mathscr{B}(0)$ est vraie.

• **Hérédité** : Supposons que $\mathscr{B}(n)$ soit vraie pour un certain $n \in \mathbb{N}$; on a donc $u_n \leqslant \left(\frac{3}{4}\right)^n$. Comme $\frac{3}{4} > 0$, on en déduit que $\frac{3}{4} \times u_n \leqslant \frac{3}{4} \times \left(\frac{3}{4}\right)^n$, soit $\frac{3}{4}u_n \leqslant \left(\frac{3}{4}\right)^{n+1}$. Or d'après 3)c), on a $u_{n+1} \leqslant \frac{3}{4}u_n$, ce qui prouve que $u_{n+1} \leqslant \left(\frac{3}{4}\right)^{n+1}$ et donc que $\mathscr{B}(n+1)$ est vraie.

• On a ainsi montré par récurrence que $\mathscr{B}(n)$ était vraie pour tout $n \in \mathbb{N}$. On a donc, pour tout $n \in \mathbb{N}$: $0 \leqslant u_n \leqslant \left(\frac{3}{4}\right)^n$.

2)e) Comme $\frac{3}{4} \in]-1;1[$, on a $\displaystyle\lim_{n \to +\infty} \left(\frac{3}{4}\right)^n = 0$. L'inégalité obtenue à la question précédente permet donc ce conclure, en vertu du théorème des gendarmes, que (u_n) converge vers 0. Cela confirme bien la conjecture faite graphiquement à la question 2) dans le cas où $0 < u_0 < 1$.

$\boxed{\text{I}}$ **a)** • L'algorithme commence par demander à l'utilisateur une valeur pour le réel u [qui constituera le terme initial u_0 d'une suite (u_n)] puis une valeur pour l'entier n.

• Commence alors une boucle (POUR k ALLANT DE 1 À n) qui sera parcourue n fois (k joue simplement le rôle d'un compteur : il prend successivement les valeurs $1; 2; \cdots; n$). À chaque fois, la valeur de u précédente est remplacée par $\frac{3u}{1+u}$. Au sortir de la boucle, l'algorithme affiche la dernière valeur de u.

$\hookrightarrow$ L'algorithme calcule et affiche le terme de rang n (choisi par l'utilisateur) de la suite de terme initial u_0 (choisi par l'utilisateur : c'est la première valeur stockée dans u) et définie par la relation de récurrence $u_{n+1} = \dfrac{3u_n}{1+u_n}$.

b) L'algorithme calcule les termes successifs de la suite u définie par $u_0 = 0,3$ et la relation de récurrence $u_{n+1} = \sqrt{u_n}$ en incrémentant un compteur N après chaque calcul, ceci tant que le terme de la suite u est strictement inférieur à 0,999999 (structure TANT QUE). L'entier N joue donc le rôle d'un compteur qui indique combien de fois a été parcourue la boucle TANT QUE. Lorsqu'on sort de la boucle, la valeur stockée dans N est affichée.

$\hookrightarrow$ L'algorithme fournit donc la valeur du plus petit entier naturel N tel que $u_{\text{N}} \geqslant 0,99999$.

c) • L'algorithme commence par stocker la valeur 0 dans la variable S puis demande à l'utilisateur une valeur à stocker dans la variable N.

• Commence alors une boucle dans laquelle l'entier k prend successivement les valeurs $1; 2; \cdots$;N. À chaque passage dans la boucle, on ajoute $\dfrac{1}{k^2}$ à la dernière valeur stockée dans S. Au sortir de la boucle, l'algorithme affiche la valeur finalement stockée dans S.

$\hookrightarrow$ L'algorithme calcule $0 + \dfrac{1}{1^2} + \dfrac{1}{2^2} + \cdots + \dfrac{1}{\text{N}^2}$ (c'est-à-dire $\displaystyle\sum_{k=1}^{\text{N}} \dfrac{1}{k^2}$), où N est la valeur entrée par l'utilisateur au départ.

$\boxed{\text{II}}$

```
VARIABLES
  U EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  k EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE n
  U PREND_LA_VALEUR 2.3
  POUR k ALLANT_DE 0 A n-1
    DEBUT_POUR
    U PREND_LA_VALEUR 0.7*U+0.5*k-1
    FIN_POUR
  AFFICHER "Un="
  AFFICHER U
FIN_ALGORITHME
```