Thème: Arithmétique

Niveau: 2nde

TI-83 Premium CE

python™

Approximation de $\sqrt{2}$ par balayage.

L.DIDIER

Compétences visées

- **chercher**, expérimenter – en particulier à l'aide d'outils logiciels ;
- **modéliser**, faire une simulation, valider ou invalider un modèle ;
- **représenter**, choisir un cadre (numérique, algébrique, géométrique...), changer de registre ;
- **calculer**, appliquer des techniques et mettre en œuvre des algorithmes.

Ces compétences sont mises en œuvre dans le cadre de l'extrait du programme de 2nde GT ci-dessous :

" Déterminer par balayage un encadrement de $\sqrt{2}$ d'amplitude inférieure ou égale à10^{-n}."

Situation déclenchante

Le nombre $\sqrt{2}$ n'est pas un nombre rationnel : il ne peut pas s'écrire sous la forme d'une fraction d'entier. Comment déterminer une valeur approchée de ce nombre ? Comme $1 \leq 2 \leq 4$, par croissance de la fonction racine carrée sur son domaine de définition, on peut encadrer $\sqrt{2}$, par les entiers 1 et 2. Comment obtenir en encadrement plus précis ?

Problématique

Ecrire un script qui permet de déterminer un encadrement d'amplitude 10^{-n}de $\sqrt{2}$ .Les bornes de l'intervalle seront arrondies à 10^{-n}.

Thème: Arithmétique

Niveau: 2nd

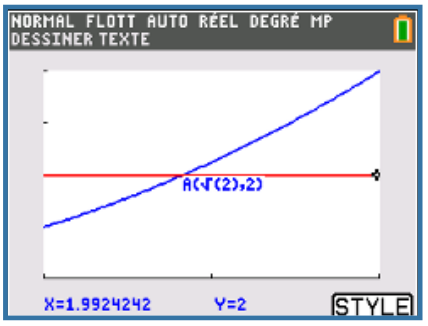
TI-83 Premium CE

python

L.DIDIER

Fiche méthode

Proposition de résolution



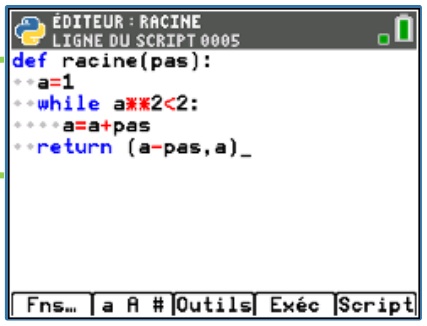
On trace la représentation graphique de la fonction f définie sur [1 ; 2] par : $f(x) = x^2$.

En traçant le tableau de variation de la fonction f sur [1; 2] par le corollaire du théorème des valeurs intermédiaires l'équation $f(x) = 2$ possède une unique solution sur [1 ;2] .

Ainsi, on crée une fonction **racine** qui prend comme argument n un entier naturel et qui renvoie deux réels avec n décimales encadrant $\sqrt{2}$ et ayant un écart de 10^{-n}. Le principe de la fonction est d'incrémenter les abscisses de 10^{-n} tant que les images restent inférieures à 2.

Première idée non satisfaisante

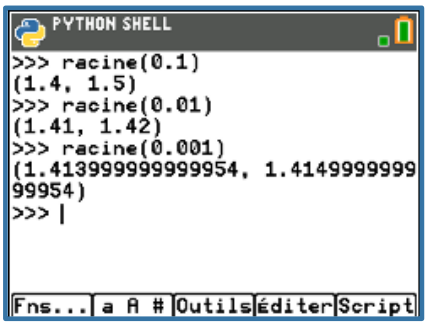
On pourrait naïvement proposer le programme ci contre.



La fonction **racine** permet de retourner 2 nombres dont l'écart est égal au `pas`.

La variable a représente la valeur des abscisses. Tant que l'image de a par la fonction carrée est inférieure à 2 on augmente l'abscisse de la valeur du pas. La boucle `while` s'arrête donc lorsqu'on vient juste de dépasser la valeur recherché.

Observons l'exécution du script pour différentes valeurs du pas.



Lors de l'exécution du script, avec un pas de 0.1 ou de 0.01 , les résultats restent cohérents. Mais avec un pas de 0.001 ou un pas plus précis, les résultats ne correspondent pas aux attentes. D'où provient ce phénomène ?

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème: Arithmétique

Niveau: 2nd

TI-83 Premium CE

python

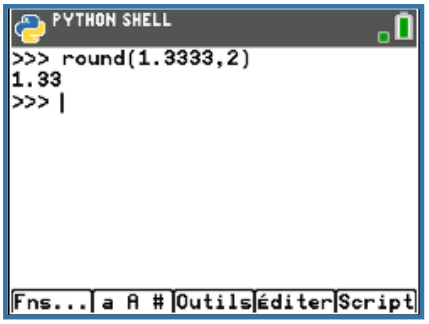
L.DIDIER

Fiche méthode

Problèmes liés aux nombres à virgule

Les nombres à virgule flottante sont représentés, au niveau matériel, en fractions de nombres binaires (base 2). Malheureusement, la plupart des fractions décimales ne peuvent pas avoir de représentation exacte en fractions binaires. Par conséquent, en général, les nombres à virgule flottante qui sont saisies **sont seulement approximés** en fractions binaires pour être stockés dans la machine.

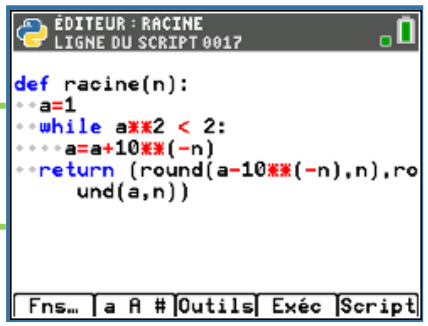
Pour éviter ce type de problème nous aurons recours dans le script à la commande `round(a,b)` qui arrondi le nombre a avec b chiffres après la virgule.



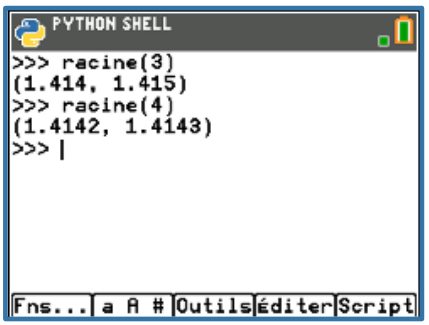
Etapes de résolution

La fonction **racine** prend comme argument n un entier naturel et renvoie deux réels avec n décimales, encadrant $\sqrt{2}$, ayant un écart de 10^{-n}. Round permet de maitriser la précision.

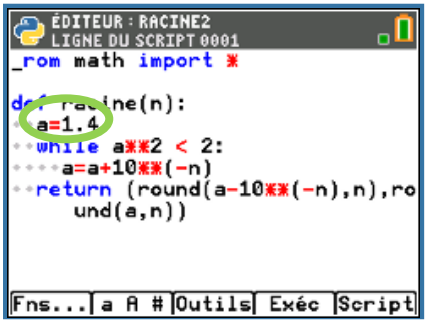
La variable a représente la valeur des abscisses. Tant que l'image de a par la fonction carré est inférieure à 2 on augmente l'abscisse de la valeur du pas (10^{-n}). La boucle while s'arrête donc lorsqu'on vient juste de dépasser le point recherché.



On obtient les résultats ci-contre lors de l'exécution de la fonction.



Pour obtenir un temps de réponse satisfaisant si l'on souhaite une précision plus grande (à partir de `n=6`) il est préférable de changer la valeur de départ dans le script : on peut par exemple saisir `a=1.4` (résultat obtenu précédemment à l'aide du script).



Remarque : Une amélioration possible de la fonction **racine** est de transformer la variable a en paramètre.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus