Thème: Fonctions

Niveau: 2nde

TI-83 Premium CE

python™

Approximation d'extremums 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 :

Pour une fonction dont le tableau de variations est donné, algorithmes d'approximation numérique d'un extrémum par balayage.

Situation déclenchante

L'étude de fonctions a pour principal objectif de déterminer des extrémums. En effet, dans la vie de tous les jours, cela peut par exemple correspondre à minimiser des coûts de production ou maximiser des bénéfices. Cependant il est assez rare de pouvoir déterminer précisément les coordonnées de ces extrémums. Comment faire alors pour obtenir au moins une valeur approchée ?

Problématique

Ecrire un programme qui permet de déterminer une approximation par balayage du maximum de la fonction $f(x) = 4x^3 - 20x^2 + 25x$ sur l'intervalle $[\,0, 2.5\,]$.

Ecrire un second programme qui permet de déterminer une approximation par balayage du minimum de la fonction $g(x) = -3x^3 + 4x + 1$ sur l'intervalle $[\,-2\,,1\,]$.

Thème: Fonctions

Niveau: 2nde

TI-83 Premium CE

python

L.DIDIER

Fiche méthode

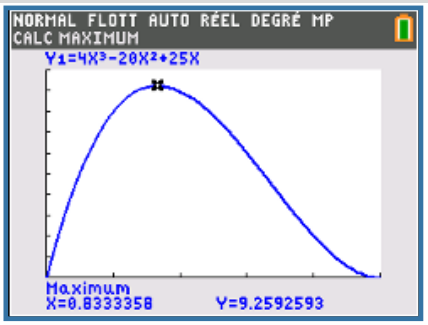
Proposition de résolution

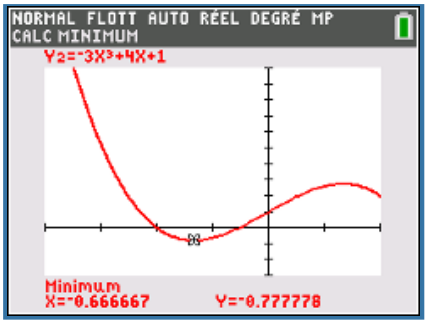
On trace la fonction : $f(x) = 4x^3 - 20x^2 + 25x$ sur l'intervalle $[\,0, 2.5\,]$ et $g(x) = -3x^3 + 4x + 1$ sur l'intervalle $[\,-2\,,1\,]$.

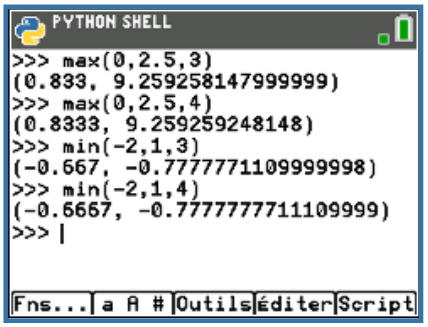
On observe un maximum ou un minimum que l'on peut faire approximer par la calculatrice. Comment fait elle pour les déterminer ? Elle possède un algorithme (surement similaire à celui proposé) qui permettra d'obtenir une approximation de ces coordonnées.

Ainsi, **on crée quatre fonctions dans ce programme (appelé aussi script)** :

- Une fonction **f (x)** qui permet de renvoyer l'image d'un réel x par la fonction f.
- Une fonction **max(a,b,n)** qui permet de renvoyer une approximation de l'abscisse du maximum pour la fonction f sur l'intervalle [a,b] avec une précision de n chiffres après la virgule pour l'approximation de l'abscisse de l'extrémum.
- Une fonction **g (x)** qui permet de renvoyer l'image d'un réel x par la fonction g.
- Une fonction **min(a,b,n)** qui permet de renvoyer une approximation du minimum pour la fonction g sur l'intervalle [a,b] avec une précision de n chiffres après la virgule pour l'approximation de l'abscisse de l'extrémum.



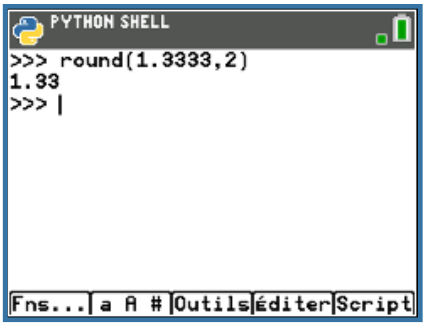




Problèmes liés aux nombres à virgule

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 programme à la commande round(a,b) qui arrondi le nombre a avec b chiffres après la virgule.



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

Thème: Fonctions

Niveau: 2nde

TI-83 Premium CE

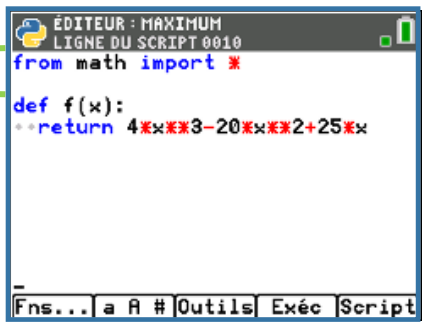
python

L.DIDIER

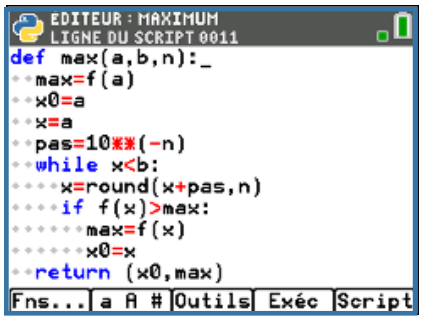
Fiche méthode

Etapes de résolution

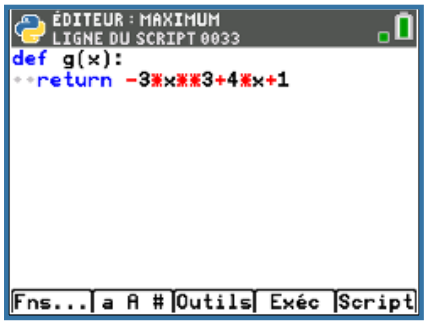
L'instruction from math import * permet d'importer la bibliothèque math et toutes les fonctions associées. Nous en avons ici besoin pour utiliser la fonction puissance pour définir la fonction f en dessous.



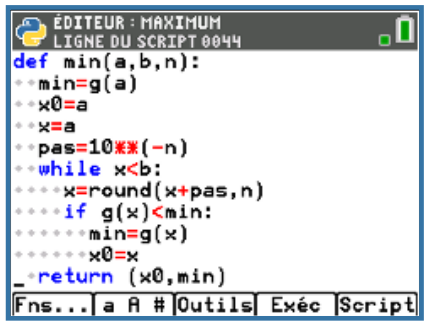
Fonction **max(a, b, n)** renvoyant une approximation des coordonnées du maximum avec une précision de n chiffres après la virgule pour son abscisse.
L'instruction x=round(x+pas,n) permet de conserver le pas souhaité.



C'est un principe à retenir : On peut appeler une fonction (ici : la fonction **f(x)**) à l'intérieur d'une autre fonction (ici: **max(a,b,n)**). L'utilisation successive de fonctions en python rend le programme dans son ensemble plus lisible **.**



Fonction **min(a, b, n)** renvoyant une approximation des coordonnées du minimum avec une précision de n chiffres après la virgule pour son abscisse.
L'instruction x=round(x+pas,n) permet de conserver le pas souhaité.



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