Thème: Fonctions

Niveau: 2nde

TI-83 Premium CE

python™

L.DIDIER

Approximation d'extremums par balayage

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 extrema. 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 extrema. Comment faire alors pour obtenir au moins une valeur approchée ?

Problématique

Ecrire un script 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 script 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

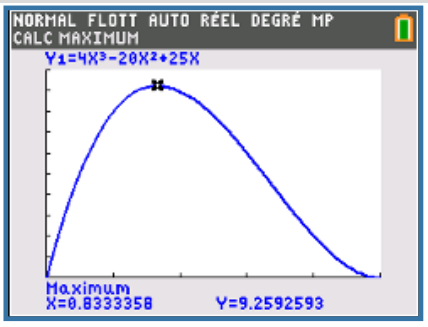
Proposition de résolution

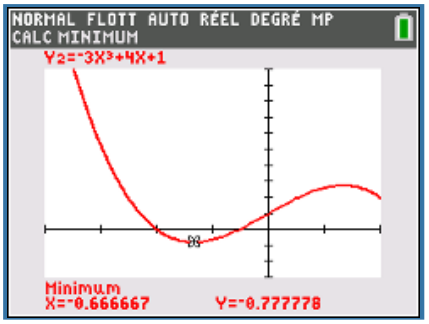
On trace les représentations graphiques des fonctions f et g définies par : $f(x) = 4x^3 - 20x^2 + 25x$ sur l'intervalle $[\,0, 2.5\,]$ et $g(x) = -3x^3 + 4x + 1$ sur l'intervalle $[\,-2\,, 1\,]$.

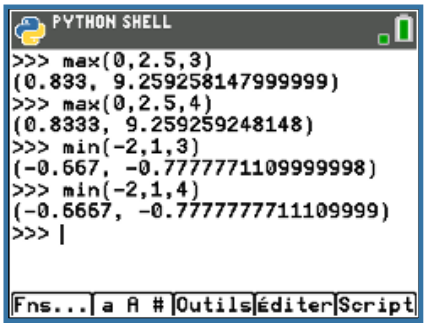
On observe les extrema dont on peut obtenir une approximation à l'aide des fonctions de la calculatrice. Comment fait elle pour les déterminer ? Elle possède un algorithme (similaire à celui proposé) qui permettra d'obtenir une approximation des coordonnées de l'extremum.

Ainsi, **on crée quatre fonctions dans ce script** :

- Une fonction **f** qui prend comme paramètre un réel x et qui renvoie l'image du réel x par la fonction **f**.
- Une fonction **max** qui prend comme paramètres a,b,n , avec a,b des réels et n un entier naturel et qui renvoie une approximation de l'abscisse du maximum de la fonction **f** sur l'intervalle [a,b] avec une précision de n chiffres après la virgule.
- Une fonction **g** qui prend comme paramètre x un réel et qui renvoie l'image du réel x par la fonction **g**.
- Une fonction **min** qui prend comme paramètres (a,b,n) , avec a,b des réels et n un entier naturel et qui permet de renvoyer une approximation du minimum de la fonction **g** sur l'intervalle [a,b] avec une précision de n chiffres après la virgule.



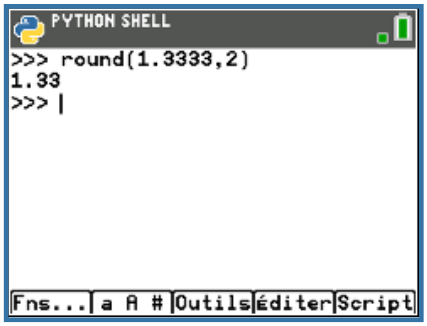




Problèmes liés aux nombres à virgule

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.



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

Thème: Fonctions

Niveau: 2nde

TI-83 Premium CE

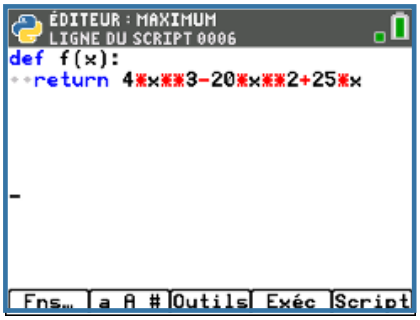
python

L.DIDIER

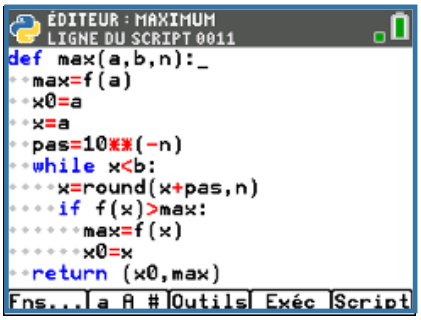
Fiche méthode

Etapes de résolution

Fonction **f** qui prend comme paramètre un réel x et qui renvoie l'image du réel x par la fonction **f**.

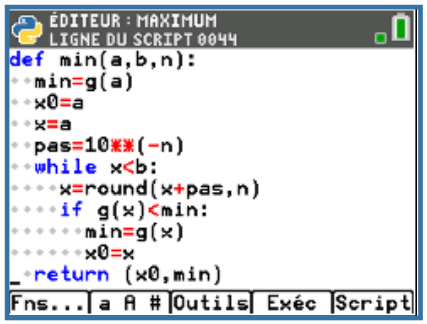


Fonction **max** qui prend comme paramètres a,b,n , avec a,b des réels et n un entier naturel et qui renvoie une approximation de l'abscisse du maximum de la fonction **f** sur l'intervalle [a,b] avec une précision de n chiffres après la virgule.
L'instruction `x=round(x+pas,n)` permet de conserver le pas souhaité.



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

Fonction **min** qui prend comme paramètres (a,b,n) , avec a,b des réels et n un entier naturel et qui permet de renvoyer une approximation de l'abscisse du minimum pour la fonction **g** sur l'intervalle [a,b] avec une précision de n chiffres après la virgule.
L'instruction `x=round(x+pas,n)` permet de conserver le pas souhaité.



Remarque pour gagner du temps : au lieu de chercher le minimum de la fonction g on pouvait chercher l'opposé du maximum de la fonction $-g$ et ainsi éviter le deuxième algorithme.

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