

Représentation graphique



Résumé : après avoir géré la construction du repère dans l'activité précédente, il s'agit ici de faire la représentation point par point ou pixel par pixel de la représentation graphique d'une fonction donnée.

Mots-clés : bibliothèque `turtle` ; fonction ; représentation graphique

Compétences visées

Chercher : « Observer, s'engager dans une démarche, expérimenter en utilisant éventuellement des outils logiciels » avec ici des résultats graphiques à observer directement.

Modéliser : « Utiliser, comprendre, élaborer une simulation numérique ».

Représenter : « passer d'un mode de représentation à un autre » en faisant le lien entre la représentation graphique d'une fonction que l'on peut faire sur une feuille et ce que l'on va exécuter par le code.

Calculer : « mettre en œuvre des algorithmes simples », en utilisant ici des fonctions à réutiliser au cours du programme principal.

Situation déclenchante

La calculatrice ou l'ordinateur est capable de tracer la représentation graphique d'une fonction. C'est souvent une boîte noire pour les élèves. Certains d'entre eux auront parfois du mal à tracer des fonctions à l'arrivée en seconde, pourtant déjà vu dans de nombreuses disciplines les années antérieures.

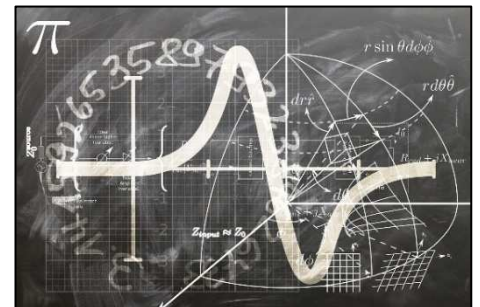


Image libre de droits d'après [Pixabay](#)

Problématique

Comment représenter graphiquement, point par point, une fonction donnée par programmation directe de la calculatrice ?

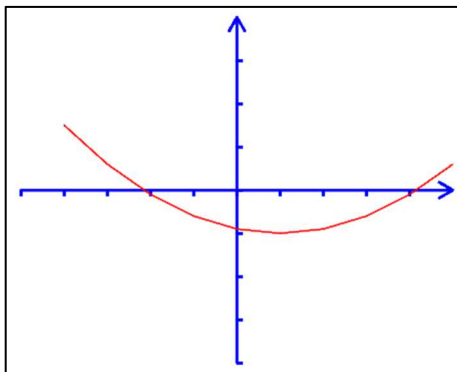


Scénario pédagogique

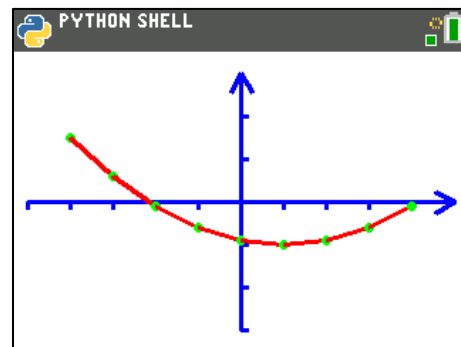
- **Avec la classe** : en relation avec l'activité précédente sur la construction d'un repère, ou en donnant le code correspondant, cette activité peut être proposée à l'ensemble de la classe, soit individuellement, soit en travail de groupe ou éventuellement dans un devoir hors classe. Il est possible de donner différentes fonctions à tracer pour inciter les élèves à réfléchir sur la problématique. L'exemple pris ici est la fonction $f: x \mapsto \frac{(x-1)^2}{10} - 1$, définie sur $[a; b]$ où a et b sont deux réels. Il faudra penser que par défaut la plage des ordonnées est de -3 à 3 , ce qui nécessite de tester les fonctions données.
- **Individuellement ou en groupe** : il est possible de donner le code du script `T05_REP.py` en amont si l'activité précédente n'a pas été réalisée et de demander ce que permet de faire ce script. Sinon, laisser les élèves créer l'algorithme puis le script correspondant en faisant le point avec eux régulièrement.
- **Mise en commun** : faire un point avec les élèves et évoquer les difficultés rencontrées dans les groupes.
- **Pour les élèves les plus en avance** : il est possible de leur proposer un ou plusieurs prolongements possibles, décrit en [fin de fiche](#).
- **Difficultés rencontrées** :
 - la notion de « sous-fonction » ;
 - le lien entre pixel et graduation.

Voici les visuels à l'issue des programmes :

en Scratch






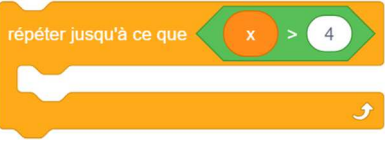

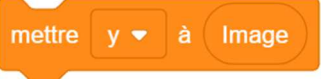

avec la TI-83 Premium CE Edition Python




Représentation graphique



Avec Scratch

Les briques de codes principales en Scratch pour ce programme	Explications	Traduction en langage Python sur la TI-83 Premium CE Edition Python
	<p>Cette instruction permet de :</p> <p>Définir un bloc utilisateur nommé fonction qui comporte ici le paramètre x.</p>	<pre>def f(X): ♦♦instructions</pre> <p>L'indentation avec ♦♦ est nécessaire pour définir les instructions à exécuter dans un bloc du code Python.</p>
	<p>Définir la fonction utilisée dans cet exemple.</p> <p>Attention à l'utilisation des blocs qui indiquent la priorité : chaque bloc correspond à un niveau de parenthésage, il n'y a pas ici de priorité de la multiplication sur l'addition par exemple.</p>	<pre>return (X-1)**2/10-1</pre>
	<p>Envoyer un message à un autre lutin afin qu'il trace le repère dans lequel la fonction sera représentée graphiquement.</p>	<pre>repere()</pre>
	<p>Faire une boucle qui permettra de calculer les coordonnées des points de la courbe. Par défaut, pour éviter d'alourdir les demandes, l'intervalle de définition est de -4 jusqu'à au moins 4.</p>	<pre>while x<x_max ♦♦instructions</pre> <p>Voir la partie « Pour mieux lire le code Python ».</p>
	<p>Définir les abscisses par incrément du pas que l'utilisateur aura donné.</p>	<pre>x+=pas</pre> <p>Cette instruction comporte un opérateur d'affectation augmenté. Il est aussi possible d'écrire <code>x=x+pas</code>.</p>
	<p>Définir l'image en se servant du bloc utilisateur fonction.</p>	<pre>y=f(x)</pre> <p>Le symbole égal signifie une affectation.</p>
	<p>Aller au point de coordonnées (x ; y)</p> <p>Le facteur « agrand » est expliqué dans la partie « Pour mieux lire le code Python ».</p>	<pre>turtle.goto(n(x),n(y))</pre>

A noter que dans la dernière version de Scratch, il faut chercher ce qui concerne le stylo dans les extensions : 

Une programmation possible est disponible sur le site de Scratch : scratch.mit.edu/studios/27615196/

Représentation graphique



Avec Python

Le code complet est à construire par les élèves.

Ce code est composé de plusieurs fonctions et nécessite l'importation de la bibliothèque `turtle` et le script `T05_REP.py`.

- La fonction `repere` sans paramètre. C'est la fonction déjà utilisée dans `T05_REP` à laquelle il faut enlever l'instruction `turtle.show()` car celle-ci arrête toute nouvelle programmation une fois lue.
- La fonction `f` de paramètre `X` qui est l'abscisse du point à créer. C'est cette fonction qu'il convient de modifier pour obtenir la fonction à représenter.

A noter ici qu'il est nécessaire « d'écraser » la courbe par un facteur un dixième afin de visualiser une courbe.

- La fonction `n` de paramètre `t` : cette fonction permet de renvoyer `t*30`. Elle sera traitée dans la partie « [Pour mieux lire le code Python](#) ».

- La fonction principale `rep_graph` de paramètres `x_min` ; `x_max` et `pas`. Les deux premiers désignent l'intervalle sur lequel la fonction sera représentée tandis que le `pas` est l'incrément pour les abscisses des points constituant la courbe.
- Cette fonction permet de tracer le repère, puis, au moyen d'une boucle `while`, la courbe est tracée par des segments de droite auquel un point vert est ajouté, grâce aux instructions :

```
turtle.color(0,255,0)
turtle.dot(3)
```

```
ÉDITEUR : T06_REPF
LIGNE DU SCRIPT 0020
def f(X):
    return (X-1)**2/10-1
```

```
ÉDITEUR : T06_REPF
LIGNE DU SCRIPT 0027
def n(t):
    return t*30
```

```
ÉDITEUR : T06_REPF
LIGNE DU SCRIPT 0030
def rep_graph(x_min,x_max,pas):
    repere()
    x=x_min
    y=f(x)
    gt(n(x),n(y))
    turtle.color(0,255,0)
    turtle.dot(3)
    while x<x_max:
        turtle.color(255,0,0)
        turtle.goto(n(x),n(y))
        turtle.color(0,255,0)
        turtle.dot(3)
    turtle.show()
```

Une programmation possible est disponible sur le site TI : education.ti.com/fr/scratch-python



Représentation graphique



Mode opératoire

Une fois le script T06_REPF.py exécuté, il faut appuyer sur la touche [var] : les fonctions définies dans le script apparaissent.

Par les flèches directionnelles, il faut sélectionner la fonction rep_graph, valider par Ok, puis ajouter les trois paramètres séparés par une virgule : l'abscisse de départ, celle d'arrivée ainsi que le pas.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de T06_REPF
>>> from T06_REPF import *
>>> rep_graph(-4,4,0.1)
```

Prolongements possibles

Voici des pistes pour les élèves les plus rapides ou qui ont envie de prolonger le travail :

- programmer la représentation graphique de la fonction racine carrée, en faisant attention à son ensemble de définition.

Pour mieux lire le code Python

Il faut faire attention avec les boucles « répéter tant que » (*while*) et « répéter jusqu'à ce que » (*until*). En effet, sans une condition d'arrêt, ces boucles peuvent créer une erreur en s'exécutant de façon infinie.

Il faudra appuyer sur la touche [on] pour arrêter le script.

Il n'y a pas de boucle « répéter tant que » dans Scratch, et pas de boucle « répéter jusqu'à ce que » dans le langage Python.

Le facteur « agrand » dans Scratch ou la fonction n dans Python sont nécessaires car l'écran est défini en pixel et on souhaite définir la fonction par rapport à une unité suffisamment visible à l'écran. Ainsi une unité vaudra 30 pixels dans le script Python.

Si un élève souhaite utiliser la fonction racine carrée, il faudra faire appel à la bibliothèque `math`. Il devra ensuite utiliser la fonction `sqrt()`.

