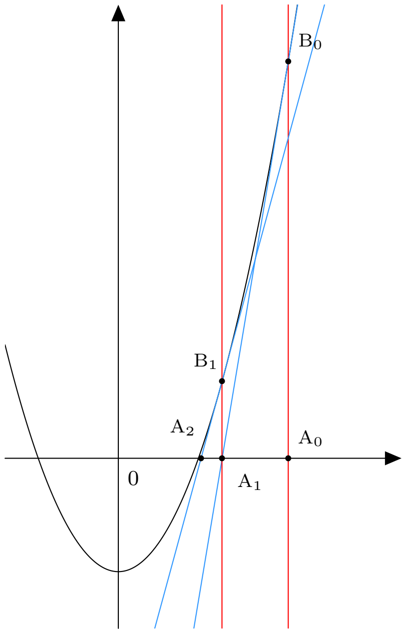
I Méthode de Newton



La méthode de Newton est une méthode numérique pour trouver une solution d'une équation de la forme $f(x) = 0$ d'inconnue x et où f est une **fonction**
Le principe est expliqué sur la figure : à partir d'un point d'abscisse x_0 (que l'on choisi a priori), tracer la tangente au point $\mathrm{B}_0 : (x_0, f(x_0))$, calculer le point d'intersection avec l'axe $(\mathrm{O}x)$ d'abscisse x_1 (on note $\mathrm{A}_1 : (x_1, 0)$ ce point) et recommencer pour trouver $x_2, \ldots$.

1.1 Première approche

Exercice 1
On souhaite ici trouver une valeur approchée de la racine positive de l'équation $x^2-x-1 = 0$.

1. Tracer la courbe d'équation $y = x^2 - x - 1$ pour des valeurs de x dans $[-3, 3]$.

```python
import numpy as np
import matplotlib.pyplot as plt
```

On utilisera la fonction `np.linspace` pour créer une liste d'abscisses convenables, puis le fait que la liste créée est en fait un vecteur numpy.

2. Déterminer l'expression de x_1 en fonction de x_0 puis plus généralement celle de x_{n+1} en fonction de x_n. Calculer ensuite grâce à python, pour différentes valeurs de x_0, la valeur de x_{10} obtenue.

   On pourra, pour changer facilement la valeur de x_0, créer une fonction dont le seul argument est la valeur de x_0.

Exercice 2
Créer une fonction **newton(f, fp, x0)** qui prend deux fonctions f et sa dérivée fp ainsi qu'un réel $x0$ comme paramètre et retourne le 10eme terme de cette suite.
On pourra dans un deuxième temps passer le rang du terme qui nous intéresse comme paramètre.

Test Tester votre fonction avec la fonction $f : x \mapsto x^2 - 3$. Quel est le nombre réel dont on vient d'obtenir une valeur approchée ? Trouver le nombre d'itération minimale pour obtenir une valeur approchée à 10^{-10} en partant de $x_0 = 2$.

1.2 Condition d'arrêt

Nous avons arbitrairement choisi de n'effectuer que 10 itérations (ou en tout cas un nombre fixé à l'avance) de la méthode (on "trace" 10 tangentes). Il existe au moins deux autres approches pour décider quand s'arrêter.

Exercice 3
On décide de passer un argument en plus, `eps` un réel que l'on choisira "proche" de 0 et qui représente la qualité de l'approximation.
Copier puis modifier le code de la fonction précédente pour obtenir une fonction `newton2(f, fp, x0, eps)` qui calcule le premier terme x_n de la suite définie en préambule et qui vérifie $|f(x_n)| \leqslant eps$. Pour garantir que l'algorithme termine toujours, on effectuera au maximum 30 itérations.
Test : résoudre l'équation x^3-2x-5 en utilisant comme choix de x_0 les valeurs $x_0 = 0.816$ puis $x_0 = 0$.
Ajouter un `print` pour savoir combien d'itération ont été effectuées.

Exercice 4
Une deuxième approche consiste non pas à mesurer si $f(x_n)$ est proche de 0 mais si x_{n+1} est proche de x_n. Cette fois la condition d'arrêt est $|x_n - x_{n-1}| \leqslant eps$, et on effectue encore au maximum 30 itérations. Répéter les tests précédents, sur une fonction `newton3`

II Illustration graphique

Exercice 5
On souhaite pouvoir obtenir un tracé semblable à la figure du début de l'énoncé, d'abord dans un cas particulier. On considère l'équation $x^2 - x - 1 = 0$ et on prend $x_0 = 2$.

On utilisera un nouveau fichier .py pour traiter cet exercice, pour exécuter plusieurs commandes de tracé avant le rendu graphique et ainsi pouvoir ajouter plusieurs courbes sur un même schéma.

1. Combien de points sont nécessaires au tracé d'une droite ? Calculer x_1 puis tracer les segments $[\mathrm{A}_0\mathrm{B}_0]$ et $[\mathrm{A}_1\mathrm{B}_1]$ en rouge en plus de la courbe.
2. On souhaite maintenant tracer la tangente T_0 au point B_0. Pour cela on va relier les points de cette tangentes d'abscisses $x_0 + 0.2$ et $x_1 - 0.2$.
3. Calculer x_2 puis tracer de même la tangente T_1 au point B_1.
4. Utiliser `plt.plot([...], [...], 'o')` pour placer les points $\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2, \mathrm{B}_0, \mathrm{B}_1$. Les listes à passer doivent contenir, comme d'habitude les coordonnées des points à placer, mais cette fois on ne les relie pas, d'où le troisième argument qui indique de placer seulement un disque à chaque emplacement.

Exercice 6

Généralisons l'approche précédente : on va créer une fonction `newton_graphique(f, fp, x0, n)` qui trace les n premières tangentes et les points correspondants.
Pour généraliser le tracé des tangentes, il faut déterminer qui de x_n et x_{n+1} est le plus grand à chaque étape, pour savoir à qui on retranche 0.2 et à qui on l'ajoute.
Utiliser cette fonction pour comprendre le comportement de la fonction utilisée à l'exercice 3 en tant qu'exemple.