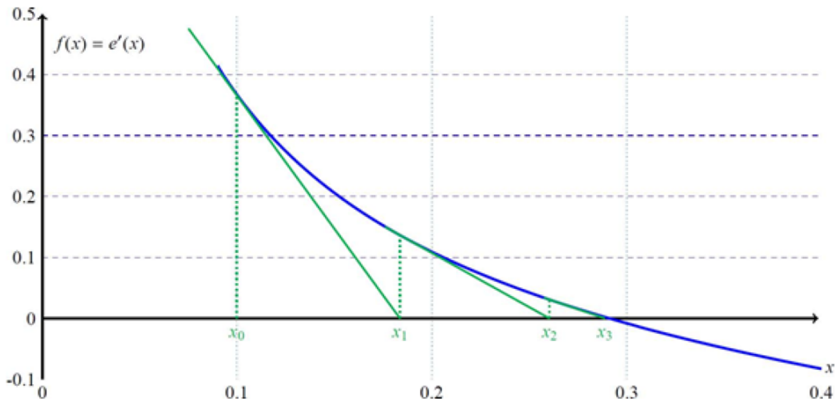
Algorithmes de résolution de f(x)=0
Partie 2 : Méthode de Newton

1 Principe de la méthode

Le principe est d'approximer la fonction f, suffisamment régulière, par sa tangente en x.
Soit f une fonction, et r une racine de l'équation $f(x) = 0$ et x_0 une valeur initiale (first guess)

Exercice 1 : Écriture de la méthode de Newton

1. Écrire l'équation de la tangente à la courbe représentative de f au point d'abscisse x_0.
2. Déterminer l'abscisse x_1 du point d'intersection de cette tangente avec l'axe des abscisses (on supposera que $f'(x) \neq 0$).
3. On réitère cette méthode par récurrence, en supposant que f' ne s'annule jamais. Donner une expression de x_n définie par récurrence.
4. Montrer que si cette suite converge, c'est nécessairement vers une limite l telle que $f(l) = 0$.



Pour mettre en œuvre cet algorithme nous utiliserons :

- en arguments d'entrée : l'expression de $f(x)$, celle de $f'(x)$, une valeur initiale x_0, un critère de précision ε. Comme nous n'avons pas de certitude de convergence, nous utiliserons également un nombre maximum d'itérations N_0 en argument, après lequel nous arrêterons notre boucle à défaut de convergence.
- en conditions d'arrêt :
  - un test de convergence : $|f(x_n)| < \varepsilon$
  - un test d'arrêt de l'algorithme en cas de suspicion de non-convergence : $n > N_0$

2 Algorithme

Entrées : f,f',x_0,ε,N_0
xi ← x_0 fx ← $f(x_0)$ c ← 0 **Tant que** abs(fx) $> \varepsilon$ **et** $c < N_0$ **faire** : fpx ← f'(xi) **si** fpx = 0 **alors** : arrêt algorithme **sinon** : xi ← xi-fx/fpx fx ← fxi c ← c+1
Sortie : xi

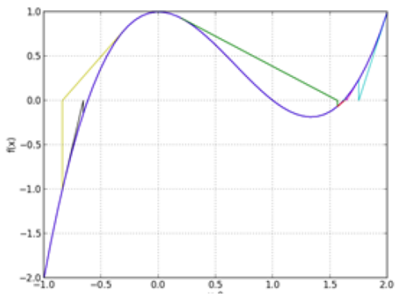
3 Implémentation en Python

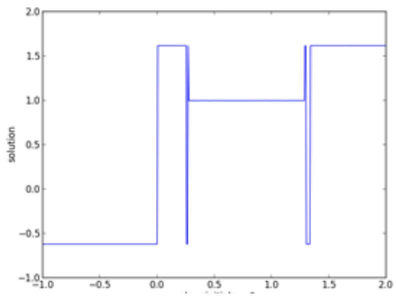
Exercice 2 : Implémenter cet algorithme en Python :

4 Étude de cas

4.1 Cas où la fonction a plusieurs racines

Considérons la fonction $f(x) = x^3 - 2x^2 + 1$ dont les racines sont $r_1 = 1$, $r_2 = 1 - \sqrt{2}$ et $r_3 = 1 + \sqrt{2}$
Suivant le choix de x_0 la méthode de Newton va converger vers une de ces trois racines.
Graphiquement on illustre à gauche les choix de $x_0 = -0.3$ $x_0 = 0.2$ et $x_0 = 2$ et à droite la racine vers laquelle on converge en fonction de x_0



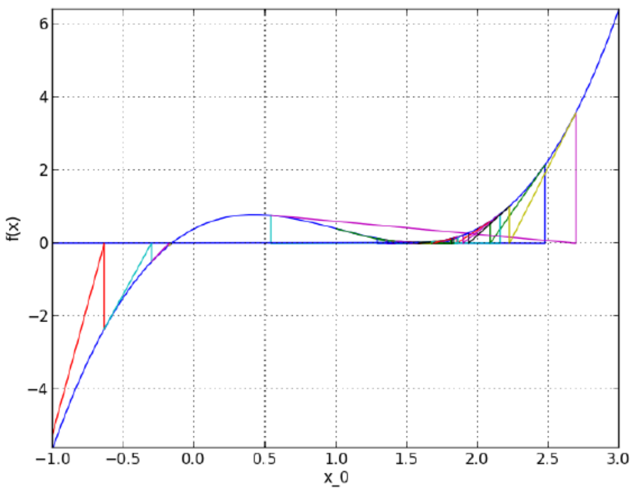


Comme on le constate, une valeur de x_0 proche d'une racine donnée ne garantit aucunement la convergence vers cette racine, ce qui posera potentiellement des problèmes de validation des résultats.

4.2 Cas où la pente est très faible

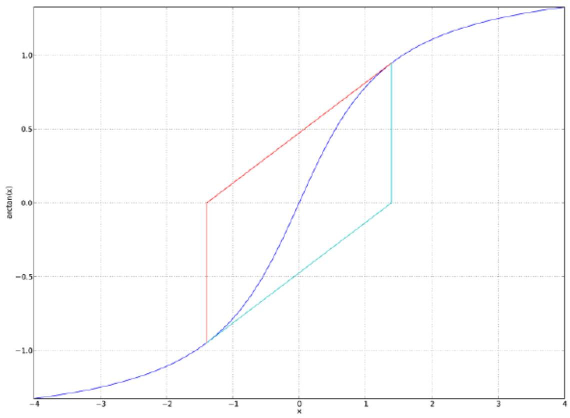
L'algorithme n'est valable que si $f'(x_i)$ est non nulle.
Si c'est le cas, un message d'erreur apparait, qui est du type saturation de la mémoire, résultat de type NaN (pour Not A Number) ou ZeroDivisionError : float division by zero.
Numériquement, si $f'(x_i)$ est proche de zéro, alors la valeur $x_(i+1)$ part à l'infini.
Illustrons ceci avec $f(x) = x^3 - 3x^2 + 2x + 0.4$ et $x_0 = 1$

Cette fonction n'a qu'une racine. Néanmoins elle a une pente nulle (et une valeur faible) en $x = 1 + \dfrac{\sqrt{2}}{3}$



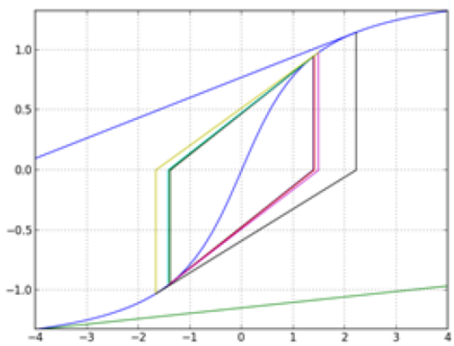
4.3 Cas d'oscillations

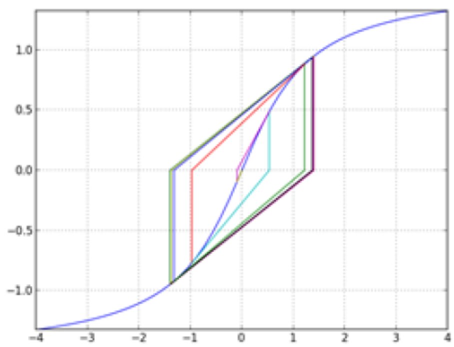
La fonction $f(x) = \arctan(x)$ admet une valeur particulière x_0 pour laquelle des oscillations se produisent :



Exercice 3 : Déterminer cette valeur

Pour des valeurs de x_0 proches de cette valeur, après quelques oscillations, il y a convergence :





4.4 Autres problèmes

D'autres problèmes peuvent encore survenir :
- La tangente à la courbe peut couper l'axe des abscisses hors du domaine de définition de la fonction.
- Si l'on utilise la méthode de Newton pour trouver l'unique zéro de la fonction définie sur $\mathbb{R}$ par $f(x) = \sqrt{|x|}$ on constate que si $x_0 \neq 0$ on a $x_{k+1} = -x_k$ pour tout k. Le problème vient de la non-dérivabilité de f en 0

5 Vitesse de convergence

Nous avons vu précédemment la notion de vitesse de convergence linéaire en analyse numérique. Lorsque la méthode de Newton converge, elle le fait à une vitesse nettement supérieure, on dit qu'elle converge quadratiquement vers sa limite. Intuitivement, cela veut dire que le nombre de décimales exactes double à chaque itération. En pratique, compte tenu de la précision permise par les calculs flottants, nous n'aurons pas besoin d'avoir des méthodes qui convergent plus rapidement.

De manière plus mathématique, et par analogie avec la définition de la vitesse de convergence linéaire ; on dit qu'une suite x_k qui converge vers une limite l **converge quadratiquement** vers l si il existe un réel 0<λ<1 tel que :

$$\lim_{x \to +\infty} \frac{|x_{k+1} - l|}{(x_k - l)^2} = \lambda$$

C'est le cas de la méthode de Newton.

6 Bilan et prolongements

6.1 Bilan

La méthode de Newton est très efficace *lorsque tout se passe bien.* Hélas, tout ne se passe pas toujours comme prévu et :
- il est bon d'utiliser un compteur d'itérations et d'arrêter la méthode si l'on pense avoir besoin de trop d'itérations
- Dans le même registre, on peut utiliser des méthodes mixtes du type dichotomie pour trouver une valeur de x_0 qui paraît prometteuse, puis une méthode de Newton à partir de cette valeur avec un faible nombre d'itérations, éventuellement suivie d'une nouvelle dichotomie etc...
- La méthode de Newton illustre un problème récurrent en Ingénierie Numérique, à savoir la validation des résultats trouvés. Il y aura en particulier lieu de s'interroger si la racine trouvée est bien celle cherchée a priori. Ce qui suivant le problème nécessite une intervention humaine éclairée, et sort du cadre de la simple algorithmique.

6.2 Utilisation d'une dérivée approchée

Occasionnellement, il sera trop fastidieux voire très difficile d'exprimer de manière exacte une dérivée. On pourra alors recourir à une dérivée numérique approchée, en assimilant le nombre dérivé qui apparaît dans la méthode de Newton à un taux de variation.

La relation de récurrence s'écrit alors :

$$x_{i+1} = x_i - \frac{f(x_i)}{(f'(x_i)} \approx x_i - \frac{f(x_i)}{\frac{f(x_i)-f(x_{i-1})}{x_i - x_{i-1}}} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

La mise en oeuvre de la méthode nécessitera alors deux valeurs initiales : x_0 et x_1

6.3 Système d'équations

Il est possible d'étendre la méthode de Newton à la recherche de solutions de systèmes d'équations.
Considérons par exemple l'équation complexe $z^3 = 1$
Posons $f(z) = z^3 - 1$ de partie réelle x et de partie imaginaire y.
L'équation s'écrit :

$$\begin{cases} x^3 - 3xy^2 - 1 = 0 \\ 3x^2y - y^3 = 0 \end{cases}$$

On définit une fonctions à deux variables et à valeurs dans $\mathbb{R}$ par

$$f(x, y) = \begin{bmatrix} f_1(x, y) = x^3 - 3xy^2 - 1 \\ f_2(x, y) = 3x^2y - y^3 \end{bmatrix}$$

Sa matrice Jacobienne, constituée de ses dérivées partielles (sera étudiée en deuxième année en Mathématiques) est :

$$Df(x, y) = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 3x^2 - 3y^2 & -6xy \\ 6xy & 3x^2 - 3y^2 \end{bmatrix}$$

La relation de récurrence s'écrit alors :

$$z_{i+1} = \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - Df(x, y)_i^{-1} . \begin{bmatrix} f(x_i) \\ f(y_i) \end{bmatrix}$$

La méthode de Newton appliquée au polynôme $z^3 - 1$ à variable complexe z converge à partir de tous les points du plan (des nombres complexes) colorés en rouge, vert ou bleu vers l'une des trois racines de ce polynôme, chacune des couleurs correspondant à une racine différente. Les points restants, se trouvant sur la structure plus claire - appelée fractale de Newton - sont les points de départ pour lesquels la méthode ne converge pas.