RESOLUTION D'EQUATIONS NON LINÉAIRES

(Dichotomie, point fixe et Newton)

1 Position du problème

Nous nous intéressons à la résolution numérique d'équations de la forme :

$f(x) = 0$, où f est une application de R dans R.

Les solutions notées $\beta_1, \beta_2, \beta_3, ...$ (Fig. 1) s'appellent les racines (ou zéros) de l'équation. Une telle équation peut admettre plusieurs solutions (voir une infinité), une seule solution, ou pas du tout

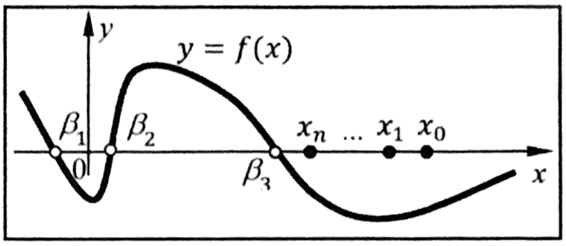


Figure 1 : Fonction, racines et suite numérique

2 Méthode de recherche d'une racine

Les méthodes utilisées sont basées sur des **algorithmes itératifs** qui construisent une suite de valeurs numériques : $x_0, x_1, x_2, \dots, x_n$ (Fig 1.) qui converge (ou pas) vers une solution de l'équation.

Nous allons étudier les propriétés et les vitesses de convergence de trois méthodes :

1. la méthode de dichotomie (dite aussi méthode de bissection),
2. la méthode de type point fixe
3. la méthode de Newton (dite aussi méthode de Newton-Raphson)

Signalons dès à présent que parmi les méthodes itératives permettant la résolution approchée d'équations non linéaires, celle de Newton figure parmi les plus puissantes. D'ailleurs, comme vous allez le voir, la méthode de Newton peut s'interpréter comme une « super» méthode de point fixe.

3 Erreur et ordre de convergence

Les méthodes étant itératives, on n'obtient, à chaque itération, qu'une estimation de la valeur de la racine. L'erreur au bout de n itérations, notée e_n, peut s'écrire :

$$e_n = |x_n - \beta| \quad (1)$$ marge d'erreur

A partir de cette définition, on dit qu'une méthode itérative est convergente à l'ordre p si il existe $C>0$ telle que :

$$|e_{n+1}| \leq C|e_n|^p \quad (2)$$

Ainsi, plus l'ordre de convergence est élevé, plus la suite x_n converge rapidement vers β, ce qui explique le penchant du numéricien pour les méthodes d'ordres élevés.

- Une convergence d'ordre 1 est dite linéaire. C'est en général le cas pour une méthode de type point fixe.
- La convergence à l'ordre 2 est dite quadratique. C'est le cas pour la méthode de Newton.
- Il peut arriver que la convergence soit établie sans pour autant avoir trouvé une constante C satisfaisant (2). La convergence est alors dite d'ordre 0. La méthode de dichotomie en est un exemple.

4 La dichotomie ou « l'art de couper en deux »

4.1 Construction de l'algorithme de dichotomie

Commençons par faire la remarque suivante : s'il existe a et b tels que $a < b$ et $f(a) * f(b) < 0$ alors il existe $\beta \in]a, b[$ tel que $f(\beta) = 0$. En d'autres termes la courbe $f(x)$ coupe l'axe des x entre a et b (Fig. 2).

Supposons que l'on ait localisé une racine β dans l'intervalle $]a, b[$ c'est-à-dire $f(a) * f(b) < 0$. Supposons de plus que cette racine soit unique sur cet intervalle. Calculons $f(\frac{a+b}{2})$ et comparons son signe avec $f(a)$. Les trois cas suivants sont possibles :

- Si $f(a) * f\left(\frac{a+b}{2}\right) < 0$ alors la racine β appartient à l'intervalle $\left]a, \frac{a+b}{2}\right[$ et on répète cette opération en considérant l'intervalle réduit à : $\left]a, \frac{a+b}{2}\right[$.
- Si $f(a) * f\left(\frac{a+b}{2}\right) > 0$ alors β appartient à l'intervalle $\left]\frac{a+b}{2}, b\right[$. On répète cette fois l'opération en considérant l'intervalle réduit à : $\left]\frac{a+b}{2}, b\right[$.
- Si $f(a) * f\left(\frac{a+b}{2}\right) = 0$ alors: $\beta = \frac{a+b}{2}$. Dans ce cas précis, mais très improbable, il est inutile de répéter le processus car la racine β a été trouvée !

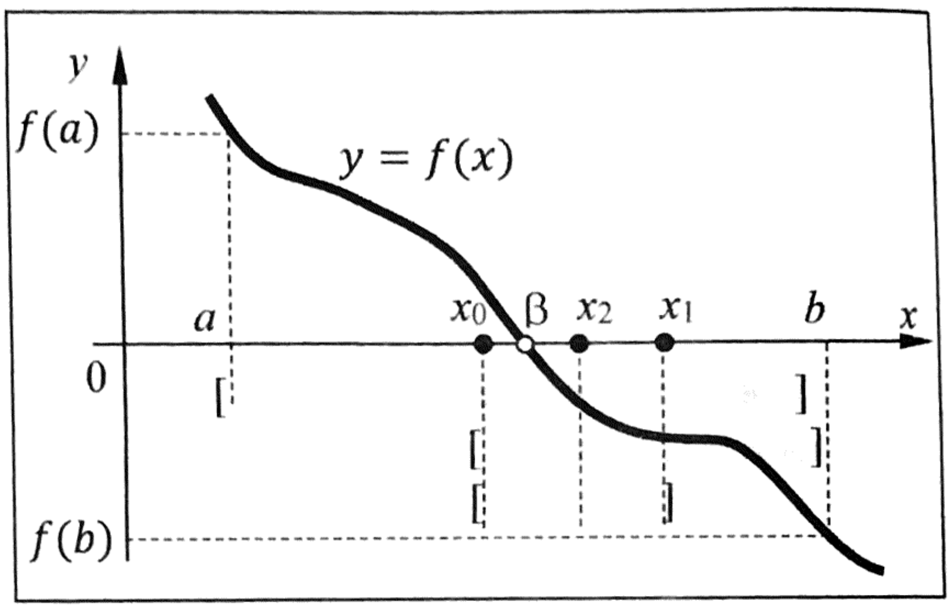


Figure 2 : Interprétation graphique de l'algorithme de dichotomie

On vient de décrire ici un algorithme de la méthode de dichotomie.

4.2 Etude de la convergence

A chaque étape, la longueur de l'intervalle noté I_n, est divisé par 2. Au bout de n itérations sa valeur s'est réduite à : $I_n = \frac{I_0}{2^n}$ (3).

On en déduit alors que : $\lim_{n\to\infty} I_n = 0$ et donc que la suite x_n converge vers la racine β.

4.3 Choix des valeurs de départ

Pour que le processus itératif de dichotomie puisse avoir une chance de réussir il faut choisir les valeurs de départ a et b tels que $a < b$ et $f(a) * f(b) < 0$

4.4 Choix d'un critère d'arrêt.

L'arrêt du processus itératif peut être défini à partir du moment où l'erreur devient plus petite qu'une valeur donnée ε appelée précision : $|x_n - \beta| \le \varepsilon$.

En pratique, on décide d'arrêter les itérations à l'étape n si la longueur de l'intervalle I_n correspondant est inférieure à la précision ε souhaitée, ce qui vérifie nécessairement la condition $|x_n - \beta| \le \varepsilon$.

Ceci nous donne d'ailleurs une information à priori sur le nombre d'itérations n à effectuer pour atteindre la précision ε. En effet, pour obtenir $|x_n - \beta| \le \varepsilon$, il faut : $\frac{b-a}{2^n} \le \varepsilon$

Travail à faire :

Q1. Calculez, littéralement, en fonction de a, b et ε le nombre n d'itérations à effectuer.

$\frac{b-a}{\varepsilon} \le 2^n \Leftrightarrow \log_2\left(\frac{b-a}{\varepsilon}\right) \le n$

$\Leftrightarrow \ln\left(\frac{b-a}{\varepsilon}\right) \le n.$

4.5 *Applications*

On considère le polynôme de Legendre de degré 5 d'expression : $f(x) = \frac{x}{8}(63x^4 - 70x^2 + 15)$
Une analyse montrerait que ses 5 racines sont toutes réelles et se situent dans l'intervalle] - 1,1[. On les note β_1 β_2 β_3 β_4 et β_5 ordonnées dans le sens croissant.

Travail à faire :

Q2. Tracer avec MatLab la courbe représentative du polynôme sur l'intervalle [-1,1] avec 500 points de tracé. La fonction f peut être définie à l'aide les outils MatLab de représentation des polynômes.
Q3. Vérifiez visuellement l'existence des 5 racines (utilisez « *grid on* » pour afficher un quadrillage).
Q4. Calculez les racines en utilisant la fonction *roots* (ou *fzero*). Stocker ces valeurs dans un tableau : beta.
Q5. Utilisez *sort* pour les trier.
Q6. Vérifiez que les valeurs calculées sont bien les racines de la fonction.

Application de la méthode à la recherche de la racine β_5.

Q7. Le choix de $a = 0.6$ *et* $b = 1$ comme intervalle de départ est-il correct ? $\ln\left(\frac{1-0,6}{\varepsilon}\right) \leq n$
a < β < b — oui
Q8. Pour différentes précisions ε dans le tableau ci-dessous, déterminez avec la réponse à la question Q1, le nombre d'itérations théoriques à effectuer pour atteindre cette valeur :

ε	10^{-6}	10^{-10}	10^{-12}	10^{-14}
Nombre d'itérations théoriques	10,6	19,8	24,4	29

Q9. Ecrire un programme MatLab mettant en œuvre la méthode de dichotomie. On limitera le nombre d'itérations à 100 pour éviter tout problème de bouclage infini.
Q10. Rechercher la racine β_5 avec une précision de $\varepsilon = 10^{-10}$. Stockez dans deux tableaux, à chaque itération la taille de l'intervalle : $I_n = b - a$ *avec* $b > a$, ainsi que l'erreur absolue définie par : $err = |x_n - \beta_5|$. On prendra pour β_5 la valeur donnée par MatLab en réponse à Q4. Utiliser la valeur stockée en mémoire pour conserver tous les chiffres significatifs disponibles dans MatLab.
Q11. Tracer I_n *et* err en fonction du nombre d'itérations n. Utiliser la fonction de tracer « semilogy ». Tracer I_n en trait en rouge discontinu 'r--', et err en trait continu '-'.
Q12. Ecrire la relation : $\log(I_n)$ en fonction de n à l'aide de la relation (3) du § 4.2. En déduire pourquoi la courbe en rouge discontinue est linéaire et décroissante. Quelle est la valeur de la pente ?
Q13. Pouvait-on s'attendre à l'allure « accidentée » de la courbe continue $err = f(n)$?
Q14. Pourquoi $\log(err) = f(n)$ est-elle toujours inférieure à $\log(I_n) = f(n)$?
Q15. Faites tourner votre programme avec différentes valeurs de ε et complétez le tableau suivant. Cela est-il en accord avec les estimations théoriques de la question Q8 ?

ε	10^{-6}	10^{-10}	10^{-12}	10^{-14}
Nb itérations numériques	19	32	39	46

5 La méthode par substitution (dite aussi du point fixe)

5.1 Définition d'un point fixe

Un point fixe d'une fonction $\varphi(x)$ est une valeur de x qui reste invariante par cette fonction et donc vérifie :

$$\varphi(x) = x$$

5.2 Application à la résolution des équations non linéaires

Soit l'équation $f(x) = 0$ et en supposant que f admet (au moins) une racine β, le point de départ de la méthode consiste à **réécrire** l'équation sous la forme équivalente :

$$f(x) = 0 \iff x = \varphi(x) \quad (4)$$

La fonction φ est appelée fonction d'itération. Elle doit être choisie de telle façon que si β est une racine de f alors β est un point fixe de φ.
La définition de φ est toujours possible et même de façons très différentes. Cependant, du choix de φ dépends la convergence ou la divergence de l'algorithme.

5.3 Convergence

Lorsqu'un algorithme de point fixe converge, la convergence est (au moins) linéaire. On montre que l'erreur $e_n = |x_n - \beta|$ à l'itération n et $e_{n+1} = |x_{n+1} - \beta|$ à l'itération $n + 1$ sont liées par la relation :

$$e_{n+1} \cong \varphi'(\beta)\, e_n \quad (5)$$

On constate que l'erreur ne peut diminuer que si :

$$|\varphi'(\beta)| < 1 \quad (6)$$

On constate facilement que la convergence est d'autant plus rapide que $\varphi'(\beta)$ est petit devant 1. Le cas limite est celui où : $\varphi'(\beta) = 0$. Dans ce cas particulier, on peut vérifier que la convergence est quadratique (d'ordre 2) : $e_{n+1} \cong k\, {e_n}^2$

5.4 Bassin d'attraction

La condition (4) est une condition nécessaire mais pas suffisante. En effet, la convergence est aussi assujettie au choix de la valeur initiale x_0. Un mauvais choix peut conduire à un algorithme divergent même si la condition (4) est remplie.
On appel bassin d'attraction de la racine β, l'ensemble des valeurs initiales pour lesquels l'algorithme converge vers β.
Il est rare que bassin d'attraction soit R tout entier. Cependant il suffit qu'il existe sur un voisinage de la racine β et que le choix de x_0 soit fait dans ce voisinage.

5.5 Point fixe attractif et répulsif

D'après le résultat du § 5.3 on peut définir qu'un point fixe β de la fonction $\varphi(x)$ est appelé attractif si $|\varphi'(\beta)| < 1$. Dans le cas contraire $|\varphi'(\beta)| > 1$, il est dit répulsif et l'algorithme diverge. Le cas ou $|\varphi'(\beta)| = 1$ est indéterminé.

5.6 *Visualisation graphique de l'algorithme de point fixe*

On peut visualiser graphiquement si la suite x_n converge ou diverge. Il suffit pour cela de représenter les fonctions $x \longrightarrow x$ et $x \longrightarrow \varphi(x)$ sur le même graphe, leur(s) intersection(s) permettant de localiser le(s) point(s) fixe(s). On remarque que le signe de φ' a une influence sur le comportement de la suite x_n.

- 1er cas: $\varphi'(\beta)$ négatif. La suite x_n converge ou diverge en oscillant de part et d'autre de β.

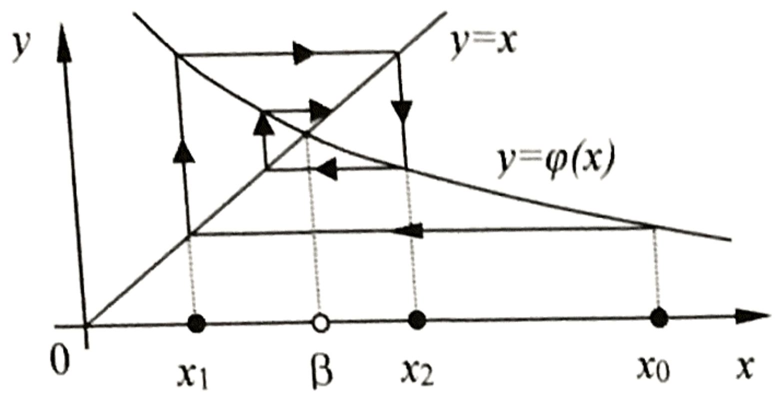
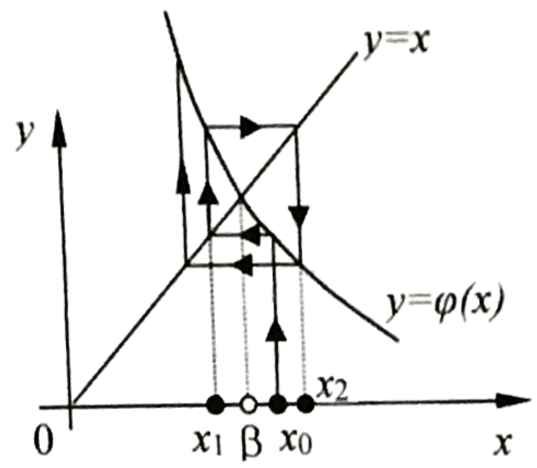


Figure 3 : Convergence en spirale : $\mathbf{-1 < \varphi'(\beta) < 0}$



Divergence en spirale : $\boldsymbol{\varphi'(\beta) < -1}$

- 2ème cas: $\varphi'(\beta)$ positif. La convergence ou la divergence de la suite x_n est monotone.

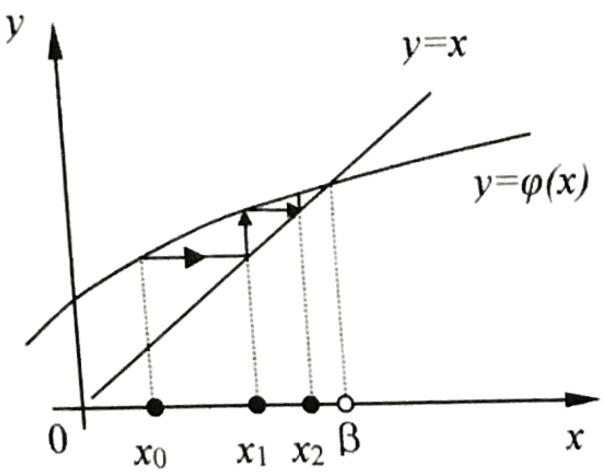
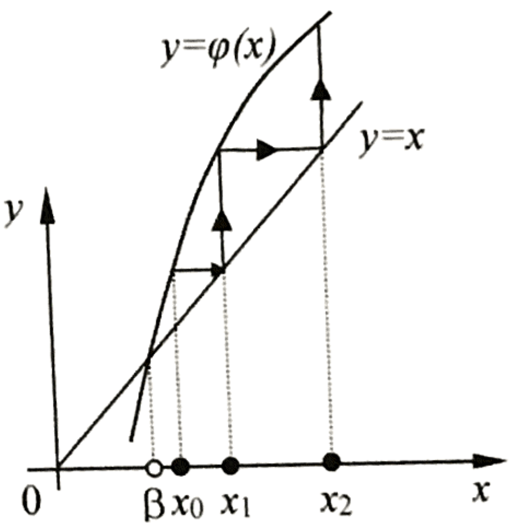


Figure 4 : Convergence en escalier : $\mathbf{0 < \varphi'(\beta) < 1}$



Divergence en escalier : $\boldsymbol{\varphi'(\beta) > 1}$

5.7 *Que faire lorsque la suite diverge ?*

Supposons que β soit un point fixe répulsif pour la fonction d'itération φ, c'est-à-dire $|\varphi'(\beta)| > 1$. Il est alors toujours possible de modifier l'équation $x = \varphi(x)$ pour nous retrouver un point fixe attractif. En effet, nous remarquons que puisque $\beta = \varphi(\beta)$ alors $\varphi^{-1}(\beta) = \beta$ et donc que : $(\varphi^{-1})'(\beta) = \frac{1}{|\varphi'(\beta)|} < 1$.

Le point fixe β qui était répulsif pour φ est maintenant attractif pour φ^{-1}. Ainsi la suite x_n définie par $x_{n+1} = \varphi^{-1}(x_n)$ avec x_0 convenablement choisi dans un voisinage de β converge vers β.

5.8 *L'algorithme du point fixe*

L'algorithme itératif, permettant de déterminer un point fixe, est construit à partir de la relation (4). Il est extrêmement simple car il s'agit de calculer successivement : $x_{n+1} = \varphi(x_n)$

```
soit x₀ une estimation de départ

Faire
        x_{n+1} = φ(x_n)
Tant que (critère de convergence non vérifié)
```

5.9 *Choix d'un critère d'arrêt*

L'arrêt du processus itératif peut être défini à partir du moment où l'erreur devient plus petite qu'une valeur donnée ε appelée précision : $|x_n - \beta| \leq \varepsilon$. **En pratique, on décide d'arrêter les itérations à l'étape n+1** si : $\frac{|x_{n+1}-x_n|}{|x_n|} < \varepsilon$. On montre que si ce critère est vérifié, le précédent l'est forcément.

5.10 *Application*

On considère la fonction non linéaire suivante : $f(x) = x^3 - 4x + 1$ (7)

Travail à faire :

Q16. Représentez graphiquement avec Matlab la fonction f sur l'intervalle [-3; 3].

Q17. On note β_1, β_2 et β_3 les trois racines réelles de f ordonnées dans le sens croissant. Calculez-les avec Matlab. Trier les et rangez-les dans un tableau nommé beta.

Q18. On désire maintenant résoudre l'équation (7) en utilisant la méthode du point fixe. Vérifiez que l'équation (7) peut se réécrire sous la forme : $x = \varphi(x) = \frac{1}{4}(x^3 + 1)$

Q19. Tracez la fonction d'itération φ sur l'intervalle [-3; 3] sur la même figure que f.

Q20. Combien de point(s) fixe(s) possède cette fonction ? Vérifiez visuellement qu'ils correspondent aux racines trouvées à la Q17.

Q21. Implémentez l'algorithme du point fixe en Matlab.

a. Utiliser le critère de convergence défini au § 5.9. Ce critère sera comparé à une valeur ε donnée.
b. Par sécurité vous limiterez le nombre d'itération à une valeur fixe.
c. Utilisez aussi la commande "*format long*" pour afficher le résultat.

Q22. Pour les différentes valeurs initiales x_0 du tableau suivant et à l'aide de votre programme, indiquez si l'algorithme converge (dans l'affirmative, indiquez vers quelle valeur).

x_0	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3
Convergence ?	non	non	oui	non	non	non	oui	non	oui	non	oui	non	non
Valeur ?			1				1		-2		1		

Q23. En considérant $x_0 = 0$, complétez les tableaux :

ε	10^{-10}	10^{-12}	10^{-15}
nombre d'itérations	/	/	/

	$\|x_{n+1} - \beta_2\|/\|x_n - \beta_2\|$
$n = 2$	
$n = 4$	
$n = 6$	
$n = 8$	

Les résultats du tableau ci-dessus sont-ils en accord avec la théorie §5.3 ?

Q24. Essayez de trouver β_1 en partant de $x_0 = -3$ et $x_0 = -2$. β_1 est-il un point fixe attractif pour φ ?

6 La méthode de Newton

Motivation : obtenir un point fixe super attractif. Dans la méthode du point fixe, la première étape consiste à réécrire $f(x) = 0$ sous la forme :

$$x = \varphi(x)$$

Cette étape (cruciale) peut se faire suivant une infinité de façons possibles et la valeur de φ' a une influence sur la convergence, plus celle-ci est petite, plus la convergence est rapide. Le choix de φ étant libre, pourquoi alors ne pas la choisir telle que $\varphi'(\beta) = 0$?

Nous obtiendrions:

- d'une part un algorithme itératif convergent à tout coup.
- d'autre part nous améliorerons grandement la vitesse de convergence.

6.1 Construction graphique de l'algorithme de Newton

Représentons l'application f sur un voisinage de β (Figure 5). Supposons que l'on dispose d'une valeur approchée x_0 de la racine β. On suppose de plus que $f'(x) \neq 0$ pour tout x dans le voisinage de la racine.

❖ étape 1 :

- on remplace la courbe représentative de f par sa tangente au point x_0. L'équation de cette droite est alors :

$$y = f'(x_0)(x - x_0) + f(x_0) \text{ avec } f'(x_0) \neq 0$$

- on calcule l'abscisse x_1 du point d'intersection de cette tangente avec l'axe $y = 0$.

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Si x_0 est suffisamment proche de β alors x_1 constitue une meilleure approximation de β.

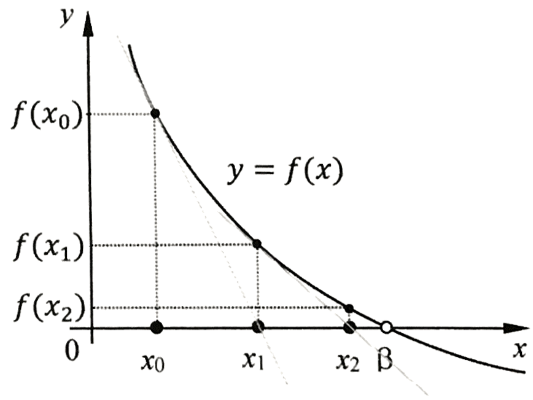


Figure 5 : Algorithme de Newton

❖ étape 2 :

A partir du nouveau point x_1, on réitère le processus. On remplace la courbe représentative de f par sa tangente au point x_1 et on calcule l'abscisse x_2 du point d'intersection de cette tangente avec l'axe y=0, c'est-à-dire:

$$y = f'(x_1)(x - x_1) + f(x_1) \quad \text{avec} \quad f'(x_1) \neq 0$$

et donc :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

De même, x_2 est une meilleure approximation de β que ne l'était x_1. Et ainsi de suite !

❖ étape n + 1 : on généralise le processus

$$y = f'(x_n)(x - x_n) + f(x_n) \quad \text{avec} \quad f'(x_n) \neq 0 \quad \text{et donc : } x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

6.2 Algorithme de Newton

L'algorithme de Newton se construit de la même manière que celui du point fixe en choisissant comme fonction d'itération la fonction : $\varphi(x) = x - \frac{f(x)}{f'(x)}$

6.3 *Convergence de l'algorithme de Newton*

En supposant f continue dérivable jusqu'à l'ordre deux et en rappelant que $f(\beta) = 0$ et que $f'(x) \neq 0$ pour tout x dans le voisinage de β, et donc a fortiori en β, on a :

$$\varphi'(x) = 1 - \frac{\left(f'(x)\right)^2 - f(x)f''(x)}{\left(f'(x)\right)^2} \qquad \text{soit} \qquad \varphi'(x) = \frac{f(x)f''(x)}{\left(f'(x)\right)^2}$$

Et on vérifie bien : $\varphi(\beta) = \beta$ et $\varphi'(\beta) = 0$

La fonction d'itération φ répond bien à nos attentes:

1. la racine β est un point fixe de puisque $\varphi(\beta) = \beta$
2. de plus β est super attractif puisque $\varphi'(\beta) = 0$

A condition de choisir x_0 convenablement dans le voisinage de β, on montre que la convergence est quadratique, **ce qui signifie que le nombre de décimales exactes double environ à chaque itération**. La convergence est donc très rapide.

Remarque : Ce point est particulièrement délicat. En théorie, comme en pratique, la valeur de x_0 doit être « assez proche » de β que l'on cherche à calculer ! Or justement, on ne connaît pas β, et donc a priori on ne connaît pas ce voisinage. En pratique, on utilise une méthode globalement convergente (comme la méthode de dichotomie présentée précédemment) pour se rapprocher de la racine. En effet, après quelques itérations, on obtient une approximation raisonnable de β. On arrête alors les itérations de dichotomie car on sait que cette méthode est relativement lente. On peut alors utiliser la valeur obtenue comme point de départ pour l'algorithme de Newton qui possède un ordre de convergence supérieur.

def Newton(x,n):
a = x
for i in range(n)):
a = a/2 + 1/a
expl
return a

6.4 *Applications*

Nous reprenons l'exemple que nous avons utilisé pour tester la méthode du point fixe. Soit f la fonction non linéaire : $f(x) = x^3 - 4x + 1$ (7)

Q25. Implémentez l'algorithme de Newton.

a. La fonction f et sa dérivée peuvent être définies à l'aide des outils de représentation des polynômes (utilisez *polyder* pour obtenir les coefficients du polynôme dérivé)
b. Le critère de convergence est le même que pour le point fixe.
c. Par sécurité vous limiterez le nombre d'itération à une valeur fixe.
d. Utilisez aussi la commande "*format long*" pour afficher le résultat.

Q26. Pour les différentes valeurs de x_0 du tableau suivant, faîtes tourner votre programme et indiquez si l'algorithme converge (dans l'affirmative, indiquez vers quelle valeur).

x_0	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3
Convergence ?													
Valeur ?	-1,8	-1,6	-1,5	-1,4									

Q27. Expliquez pourquoi l'algorithme converge vers l'une ou l'autre des racines ?

Q28. En considérant $x_0 = -3$, complétez le tableau suivant :

ε	10^{-10}	10^{-12}	10^{-15}
nombre d'itérations			

Q29. En prenant $x_0 = -3$ et en affichant la solution à chaque itération, vérifiez que le nombre de décimales justes doublent à chaque itération comme prévu au § 6.3