Résolution numérique de $f(x) = 0$

Ivan Noyer

Lycée Thiers

Objectifs

- On veut donner une solution approchée d'une équation de type $f(x) = 0$ où est f est une fonction numérique continue.

Objectifs

- On veut donner une solution approchée d'une équation de type $f(x) = 0$ où est f est une fonction numérique continue.
- Deux méthodes :

Objectifs

- On veut donner une solution approchée d'une équation de type $f(x) = 0$ où est f est une fonction numérique continue.
- Deux méthodes :
  - Dichotomie : facile à mettre en œuvre. Peu de vérifications préalables.

Objectifs

- On veut donner une solution approchée d'une équation de type $f(x) = 0$ où est f est une fonction numérique continue.
- Deux méthodes :
  - Dichotomie : facile à mettre en œuvre. Peu de vérifications préalables.
  - Méthode de Newton. Souvent plus performante. Davantage de prérequis.

Principe

- Basée sur le théorème des valeurs intermédiaires : Si f est une fonction numérique continue sur $[a, b]$ et si $f(a)$ et $f(b)$ sont de signes contraires et non nuls, alors il existe $c \in]a, b[$ tel que $f(c) = 0$.

Principe

- Basée sur le théorème des valeurs intermédiaires : Si f est une fonction numérique continue sur $[a, b]$ et si $f(a)$ et $f(b)$ sont de signes contraires et non nuls, alors il existe $c \in]a, b[$ tel que $f(c) = 0$.
- $f(a)f(b) < 0$.

Principe

- Basée sur le théorème des valeurs intermédiaires : Si f est une fonction numérique continue sur $[a, b]$ et si $f(a)$ et $f(b)$ sont de signes contraires et non nuls, alors il existe $c \in]a, b[$ tel que $f(c) = 0$.
- $f(a)f(b) < 0$.
- Voir `dichot.pdf`.

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$
- $f(\frac{1+2}{2}) = f(1.5) = 0.25$. $I_1 = [1; 1.5]$.

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$
- $f(\frac{1+2}{2}) = f(1.5) = 0.25$. $I_1 = [1; 1.5]$.
- $f(\frac{1+1.5}{2}) = f(1.25) = -0.4375$, $I_2 = [1.25; 1.5]$. Précision 0.25.

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$
- $f(\frac{1+2}{2}) = f(1.5) = 0.25$. $I_1 = [1; 1.5]$.
- $f(\frac{1+1.5}{2}) = f(1.25) = -0.4375$, $I_2 = [1.25; 1.5]$. Précision 0.25.
- $f(1.375) = -0.109375$. $I_3 = [1.375, 1.5]$. Précision 0.125.

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$
- $f(\frac{1+2}{2}) = f(1.5) = 0.25$. $I_1 = [1; 1.5]$.
- $f(\frac{1+1.5}{2}) = f(1.25) = -0.4375$, $I_2 = [1.25; 1.5]$. Précision 0.25.
- $f(1.375) = -0.109375$. $I_3 = [1.375, 1.5]$. Précision 0.125.
- $(1.375 + 1.5)/2 = 1.4375$; $f(1.4375) = 0.06640625$.
  $I_4 = [1.375, 1.4375]$. Précision : 0.0625.

Exemple

Approcher $\sqrt{2}$ à 0.1 près

- $f(x) = x^2 - 2$, $I = [1, 2]$, $f(1) = -1$, $f(2) = 2$. Précision 1. $I_0 = [1, 2]$
- $f(\frac{1+2}{2}) = f(1.5) = 0.25$. $I_1 = [1; 1.5]$.
- $f(\frac{1+1.5}{2}) = f(1.25) = -0.4375$, $I_2 = [1.25; 1.5]$. Précision 0.25.
- $f(1.375) = -0.109375$. $I_3 = [1.375, 1.5]$. Précision 0.125.
- $(1.375 + 1.5)/2 = 1.4375$; $f(1.4375) = 0.06640625$.
  $I_4 = [1.375, 1.4375]$. Précision : 0.0625.
- On retourne 1.4375 ou bien 1.375 ou bien $(1.375 + 1.4375)/2$ (sans doute le mieux, sauf si la bonne valeur est très proche des bornes de I_4).

Algorithme

- Le pseudo-code :

```
Données : f,a,b,p # p : precision voulue
g,d:=a,b # par hyp f(a)f(b)<=0

tant que d-g>2p faire
    m:=(g+d)/2 #milieu
    si f(g)f(m)<= 0 alors
        d:=m
    sinon
        g:=m

retourner (g+d)/2
```

Algorithme

- Le pseudo-code :

```
Données : f,a,b,p # p : precision voulue
g,d:=a,b # par hyp f(a)f(b)<=0

tant que d-g>2p faire
    m:=(g+d)/2 #milieu
    si f(g)f(m)<= 0 alors
        d:=m
    sinon
        g:=m

retourner (g+d)/2
```

- On retourne $(g+d)/2$ et pas m car à la sortie de boucle $d-g \leq 2p$ et m est égal à d ou g. Et donc, si x_0 est la solution exacte, on peut avoir $x_0 - m > p$, alors que x_0 est au plus à la distance p du milieu.

Analyse : Terminaison

On numérote les passages dans la boucle à partir de 1.
On vérifie par récurrence qu'à la fin du k-ième passage dans la boucle, on a $d_k - g_k = \dfrac{b-a}{2^k}$.

- Vrai à la fin du passage zéro : $d_0 - g_0 = b - a = \dfrac{b-a}{2^0}$.

Analyse : Terminaison

On numérote les passages dans la boucle à partir de 1.
On vérifie par récurrence qu'à la fin du k-ième passage dans la boucle, on a $d_k - g_k = \dfrac{b-a}{2^k}$.

- Vrai à la fin du passage zéro : $d_0 - g_0 = b - a = \dfrac{b-a}{2^0}$.
- Hérédité : Si $d_{k-1} - g_{k-1} > 2p$ à l'entrée de la boucle k, alors $d_k = \dfrac{d_{k-1} + g_{k-1}}{2}$ et $g_k = g_{k-1}$ ou ($g_k = \dfrac{d_{k-1} + g_{k-1}}{2}$ et $d_k = d_{k-1}$).

  Dans les deux cas : $d_k - g_k = \dfrac{d_{k-1} - g_{k-1}}{2} = \dfrac{d_0 - g_0}{2^k}$ (en utilisant HR) à l'entrée de la boucle $k+1$. Hérédité : OK

Analyse : Terminaison

On numérote les passages dans la boucle à partir de 1.
On vérifie par récurrence qu'à la fin du k-ième passage dans la boucle, on a $d_k - g_k = \dfrac{b-a}{2^k}$.

- Vrai à la fin du passage zéro : $d_0 - g_0 = b - a = \dfrac{b-a}{2^0}$.
- Hérédité : Si $d_{k-1} - g_{k-1} > 2p$ à l'entrée de la boucle k, alors $d_k = \dfrac{d_{k-1} + g_{k-1}}{2}$ et $g_k = g_{k-1}$ ou ($g_k = \dfrac{d_{k-1} + g_{k-1}}{2}$ et $d_k = d_{k-1}$).

  Dans les deux cas : $d_k - g_k = \dfrac{d_{k-1} - g_{k-1}}{2} = \dfrac{d_0 - g_0}{2^k}$ (en utilisant HR) à l'entrée de la boucle $k+1$. Hérédité : OK
- Donc $d_k - g_k$ est une quantité positive qui tend vers zéro et passe donc au bout d'un nombre fini d'itérations sous la barre des $2p$ (qui est la condition d'arrêt). Terminaison OK.

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$
- Cas de base : OK.

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$
- Cas de base : OK.
- Hérédité : au début de l'itération k, on suppose que $f(d_{k-1})f(g_{k-1}) \leq 0$. On pose $m = \dfrac{d_{k-1} + g_{k-1}}{2}$.

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$
- Cas de base : OK.
- Hérédité : au début de l'itération k, on suppose que $f(d_{k-1})f(g_{k-1}) \leq 0$. On pose $m = \dfrac{d_{k-1} + g_{k-1}}{2}$.
  1. Si $f(g_{k-1})f(m) \leq 0$, alors $d_k = m$, et $g_k = g_{k-1}$. Alors $f(g_k)f(d_k) = f(g_{k-1})f(m) \leq 0$. Hérédité : OK

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$
- Cas de base : OK.
- Hérédité : au début de l'itération k, on suppose que $f(d_{k-1})f(g_{k-1}) \leq 0$. On pose $m = \dfrac{d_{k-1} + g_{k-1}}{2}$.
  1. Si $f(g_{k-1})f(m) \leq 0$, alors $d_k = m$, et $g_k = g_{k-1}$. Alors $f(g_k)f(d_k) = f(g_{k-1})f(m) \leq 0$. Hérédité : OK
  2. Si $f(g_{k-1})f(m) > 0$, alors $g_k = m$, $d_k = d_{k-1}$ et $f(m)$ de même signe que $f(g_{k-1})$. Alors $f(g_k)f(d_k) = f(m)f(d_{k-1})$.
     Donc $f(m)f(d_{k-1})$ est du même signe que $f(g_{k-1})f(d_{k-1})$, c.a.d négatif. Hérédité : OK

Analyse : Correction

- On veut montrer que le résultat retourné est un réel r tel qu'il existe une solution x_0 de $f(x) = 0$ qui vérifie $|x_0 - r| \leq p$.
- Invariant : à chaque itération $f(g)f(d) \leq 0$
- Cas de base : OK.
- Hérédité : au début de l'itération k, on suppose que $f(d_{k-1})f(g_{k-1}) \leq 0$. On pose $m = \dfrac{d_{k-1} + g_{k-1}}{2}$.
  1. Si $f(g_{k-1})f(m) \leq 0$, Hérédité : OK
  2. Si $f(g_{k-1})f(m) > 0$, Hérédité : OK
- L'invariant est donc vérifié lorsqu'on sort de la boucle à l'étape φ (fin).
  On a $f(d_\varphi)f(g_\varphi) \leq 0$ et $d_\varphi - g_\varphi \leq 2p$. Un zéro x_0 se trouve entre les deux, soit parce que $f(d_\varphi)f(g_\varphi) = 0$, soit par TVI. Le réel r renvoyé est le milieu de $[g_\varphi, d_\varphi]$. Du coup $|x_0 - r| < p$.

Complexité

- On compte les passages dans la boucle.

Complexité

- On compte les passages dans la boucle.
- Avant l'itération k, on a $d_{k-1} - g_{k-1} = \dfrac{b-a}{2^{k-1}}$ (par récurrence). On fait un passage k si et seulement si $\dfrac{b-a}{2^{k-1}} > 2p$, donc si et seulement si $2^k < \dfrac{b-a}{p}$.

Complexité

- On compte les passages dans la boucle.
- Avant l'itération k, on a $d_{k-1} - g_{k-1} = \dfrac{b-a}{2^{k-1}}$ (par récurrence). On fait un passage k si et seulement si $\dfrac{b-a}{2^{k-1}} > 2p$, donc si et seulement si $2^k < \dfrac{b-a}{p}$.
- Ceci est vrai si et seulement si $k < \log_2(\dfrac{b-a}{p})$.

Complexité

- On compte les passages dans la boucle.
- Avant l'itération k, on a $d_{k-1} - g_{k-1} = \frac{b-a}{2^{k-1}}$ (par récurrence). On fait un passage k si et seulement si $\frac{b-a}{2^{k-1}} > 2p$, donc si et seulement si $2^k < \frac{b-a}{p}$.
- Ceci est vrai si et seulement si $k < \log_2(\frac{b-a}{p})$.
- Il faut donc $k \leq \lfloor \log_2(\frac{b-a}{p}) \rfloor$. Si $k = \lfloor \log_2(\frac{b-a}{p}) \rfloor + 1$, on sort de la boucle. Le nombre de passage dans la boucle est en $O\left(\ln(\frac{b-a}{p})\right)$.

Complexité

- On compte les passages dans la boucle.
- Avant l'itération k, on a $d_{k-1} - g_{k-1} = \frac{b-a}{2^{k-1}}$ (par récurrence). On fait un passage k si et seulement si $\frac{b-a}{2^{k-1}} > 2p$, donc si et seulement si $2^k < \frac{b-a}{p}$.
- Ceci est vrai si et seulement si $k < \log_2(\frac{b-a}{p})$.
- Le nombre de passage dans la boucle est en $O\left(\ln(\frac{b-a}{p})\right)$.
- Nombre d'opérations élémentaires en $O\left(\ln(\frac{b-a}{p})\right)$ et nombre d'appel à f en $O\left(\ln(\frac{b-a}{p})\right)$ (temps constant ?)La complexité dépend de celle de f.

Complexité

- On compte les passages dans la boucle.
- Avant l'itération k, on a $d_{k-1} - g_{k-1} = \dfrac{b-a}{2^{k-1}}$ (par récurrence). On fait un passage k si et seulement si $\dfrac{b-a}{2^{k-1}} > 2p$, donc si et seulement si $2^k < \dfrac{b-a}{p}$.
- Ceci est vrai si et seulement si $k < \log_2(\dfrac{b-a}{p})$.
- Le nombre de passage dans la boucle est en $O\left(\ln(\dfrac{b-a}{p})\right)$.
- La complexité dépend de celle de f.
- La complexité dans le meilleur des cas est la même : pas de test dans la boucle conduisant à une sortie anticipée.

La dichotomie dans les fonctions de bibliothèque

Bien sûr, elle est déjà implantée (la méthode est appelée *bisection* en anglais :

```
>>> import scipy.optimize as so
>>> dichotomie(lambda x : x**2-2, 1, 2, 0.000001)
1.4142141342163086
>>> so.bisect(lambda x : x**2-2, 1, 2)
1.4142135623724243
>>> import numpys as np
>>> so.bisect(np.sin, 3, 4)#annule sinus entre 3 et 4
3.141592653589214
```

Code

```
1  def dichotomie(f, a, b, epsilon):
2      """
3      f : fonction, a,b : bornes
4      epsilon : précision cherchée
5      """
6      assert f(a) * f(b) <= 0 and epsilon > 0
7      g, d = a, b
8      while d - g > 2 * epsilon:
9          m = (g + d) / 2
10         if f(g) * f(m) <= 0:
11             d = m
12         else:
13             g = m
14     return (g+d)/2.
```

Historique

- Méthode de Newton ou méthode de Newton-Raphson : trouver une approximation précise d'un zéro (ou racine) d'une fonction réelle d'une variable réelle.

Historique

- Méthode de Newton ou méthode de Newton-Raphson : trouver une approximation précise d'un zéro (ou racine) d'une fonction réelle d'une variable réelle.
- Isaac Newton (1643-1727) et Joseph Raphson (peut-être 1648-1715) : recherche des zéros d'une équation polynomiale.

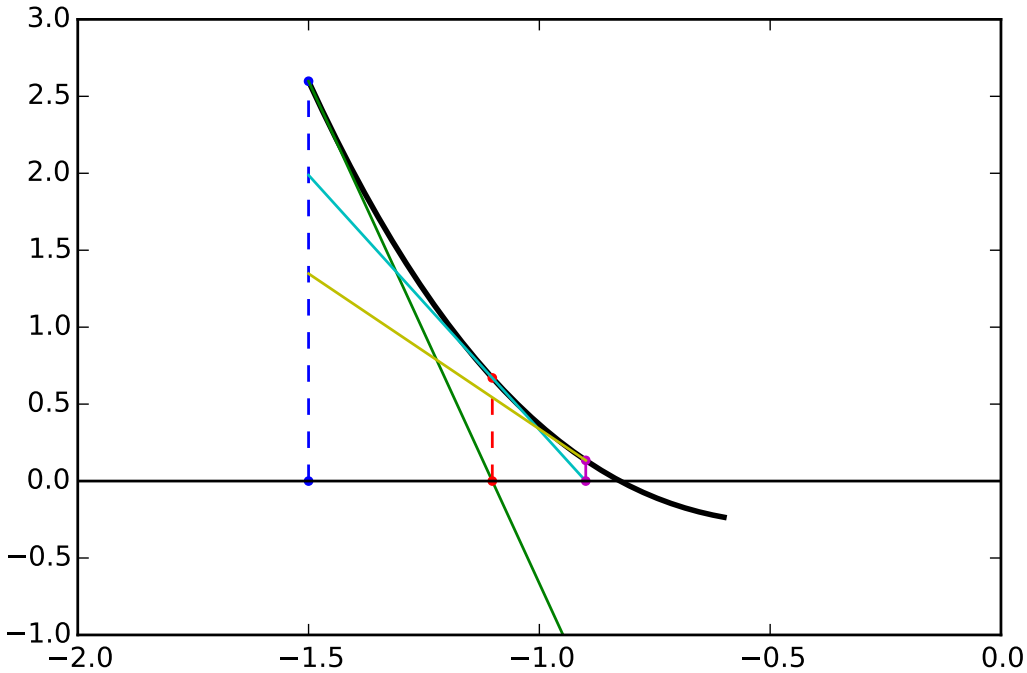
Historique

- Méthode de Newton ou méthode de Newton-Raphson : trouver une approximation précise d'un zéro (ou racine) d'une fonction réelle d'une variable réelle.
- Isaac Newton (1643-1727) et Joseph Raphson (peut-être 1648-1715) : recherche des zéros d'une équation polynomiale.
- Élargie par Thomas Simpson (1710-1761) (grâce à la notion de dérivée) : calculer un zéro d'une équation non linéaire, pouvant ne pas être un polynôme, et d'un système formé de telles équations. (source Wikipedia)

Principe général

À chaque itération, la fonction dont on cherche un zéro est linéarisée (on confond la fonction et sa tangente) en l'itéré (ou abscisse) courant(e) et l'itéré suivant est pris égal au zéro de la fonction linéarisée.

Figure – Abscisses successives dans la méthode de Newton



Principe général

- Deux conditions sont requises pour la bonne marche de l'algorithme : la fonction doit être dérivable aux points visités (pour pouvoir y linéariser la fonction) et les dérivées ne doivent pas s'y annuler (pour que la tangente rencontre l'axe des absisses)

Principe général

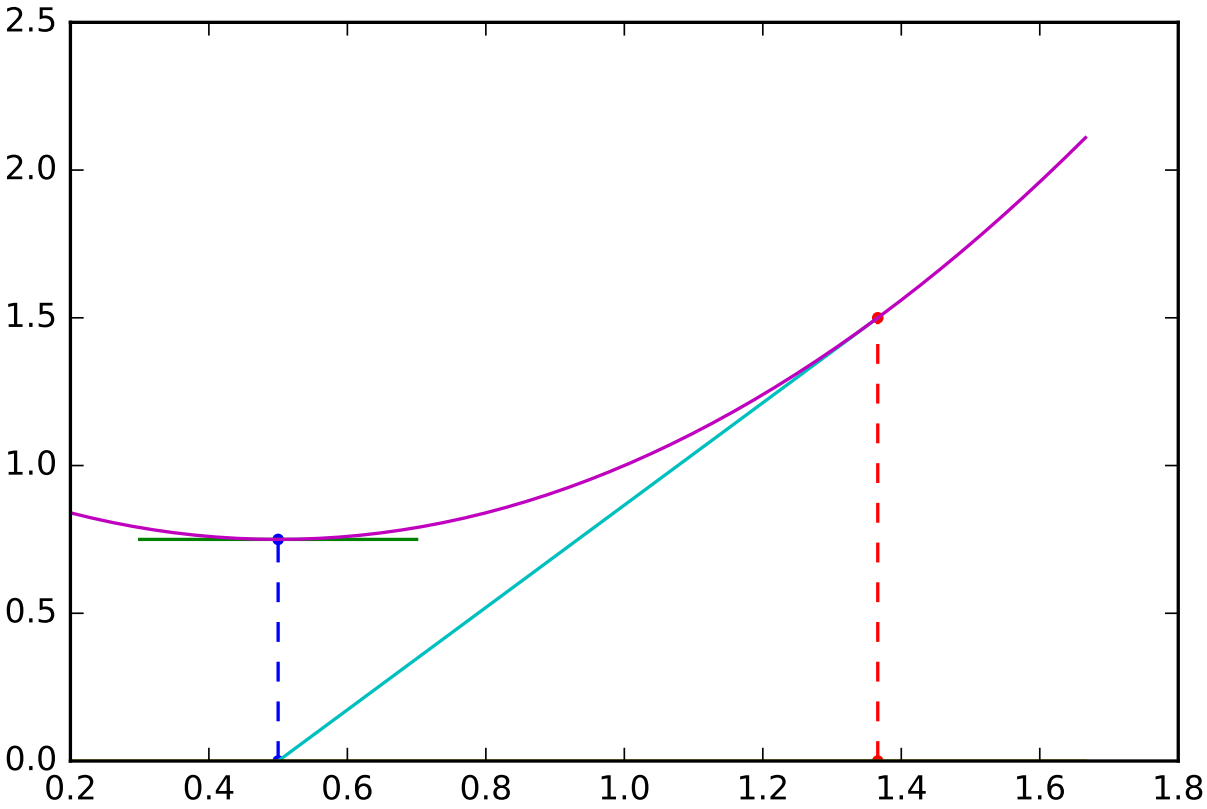
- Deux conditions sont requises pour la bonne marche de l'algorithme : la fonction doit être dérivable aux points visités (pour pouvoir y linéariser la fonction) et les dérivées ne doivent pas s'y annuler (pour que la tangente rencontre l'axe des absisses)
- S'ajoute à ces conditions la contrainte forte de devoir prendre le premier itéré assez proche d'un zéro régulier de la fonction (i.e., en lequel la dérivée de la fonction ne s'annule pas), pour que la convergence du processus soit assurée.

Principe général

- Deux conditions sont requises pour la bonne marche de l'algorithme : la fonction doit être dérivable aux points visités (pour pouvoir y linéariser la fonction) et les dérivées ne doivent pas s'y annuler (pour que la tangente rencontre l'axe des absisses)
- S'ajoute à ces conditions la contrainte forte de devoir prendre le premier itéré assez proche d'un zéro régulier de la fonction (i.e., en lequel la dérivée de la fonction ne s'annule pas), pour que la convergence du processus soit assurée.
- Autrement dit : Il est nécessaire (on l'espère...) de ne jamais rencontrer de point en lequel la dérivée de f s'annule.

Cas d'une tangente horizontale

Figure – Le second itéré est l'abscisse d'un extrêmum



Description

1. f dérivable. On choisit x_0 proche du zéro à trouver (évaluation grossière de ce zéro).

Description

1. f dérivable. On choisit x_0 proche du zéro à trouver (évaluation grossière de ce zéro).
2. On a au voisinage de x_0 : $f(x) \simeq f(x_0) + f'(x_0)(x - x_0)$. Intersection tangente/axe des abscisses en $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$.

Description

1. f dérivable. On choisit x_0 proche du zéro à trouver (évaluation grossière de ce zéro).
2. On a au voisinage de x_0 : $f(x) \simeq f(x_0) + f'(x_0)(x - x_0)$. Intersection tangente/axe des abscisses en $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$.
3. On itère.

Description

1. f dérivable. On choisit x_0 proche du zéro à trouver (évaluation grossière de ce zéro).
2. On a au voisinage de x_0 : $f(x) \simeq f(x_0) + f'(x_0)(x - x_0)$. Intersection tangente/axe des abscisses en $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$.
3. On itère.
4. Critère d'arrêt : deux possibles $|f(x_k)| \leq \varepsilon_1$ ou bien $|x_{k+1} - x_k| \leq \varepsilon_2$ où $\varepsilon_1, \varepsilon_2 \in \mathbb{R}^+$ représentent des erreurs d'approximations caractérisant la qualité de la solution numérique.
   Dans tous les cas, il se peut que le critère d'arrêt soit vérifié en des points ne correspondant pas à des solutions de l'équation à résoudre.

Justification

On peut montrer que :

- Si le zéro inconnu α est isolé (seul dans un intervalle), alors il existe un voisinage de α tel que pour toutes les valeurs de départ x_0 dans ce voisinage, la suite $(x_k)_k$ va converger vers α.

Justification

On peut montrer que :

- Si le zéro inconnu α est isolé (seul dans un intervalle), alors il existe un voisinage de α tel que pour toutes les valeurs de départ x_0 dans ce voisinage, la suite $(x_k)_k$ va converger vers α.
- De plus, si $f'(\alpha)$ est non nul, alors la convergence est quadratique, ce qui signifie intuitivement que le nombre de chiffres corrects est approximativement doublé à chaque étape.

Exemple

- On cherche un nombre positif x vérifiant $\cos(x) = x^3$.

Exemple

- On cherche un nombre positif x vérifiant $\cos(x) = x^3$.
- Reformulation : Racine positive de $f(x) = \cos(x) - x^3$. Dérivée $f'(x) = -\sin(x) - 3x^2$.

Exemple

- On cherche un nombre positif x vérifiant $\cos(x) = x^3$.
- Reformulation : Racine positive de $f(x) = \cos(x) - x^3$. Dérivée $f'(x) = -\sin(x) - 3x^2$.
- Comme $\cos(x) \leqslant 1$ pour tout x et $x^3 > 1$ pour $x > 1$, le zéro se situe entre 0 et 1. Essayons $x_0 = 0,5$.

$$
\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 0{,}5 - \frac{\cos(0{,}5) - 0{,}5^3}{-\sin(0{,}5) - 3 \times 0{,}5^2} &&\simeq 1{,}112\,141\,637\,1 \\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} &&\simeq 0{,}909\,672\,693\,736 \\
x_3 &\vdots &&\simeq 0{,}866\,263\,818\,209 \\
x_4 &\vdots &&\simeq 0{,}865\,477\,135\,298 \\
x_5 &\vdots &&\simeq 0{,}865\,474\,033\,111 \\
x_6 &\vdots &&\simeq 0{,}865\,474\,033\,101 \\
x_7 &\vdots &&\simeq 0{,}865\,474\,033\,102
\end{aligned}
$$

Algorithme

```
1  entree : f : application dérivable,
2           x0 : abscisse proche d'un zéro
3           ε : précision voulue
4  sortie : un itéré proche du zéro cherché à la précision voulue
5
6  /*ancien itéré, nouvel itéré*/
7  a, n := x0, x0 − f(x0)/f'(x0)
8  /*critère d'arrêt 'horizontal' */
9  tant_que |n − a| > ε faire
10       a := n
11       n := n − f(n)/f'(n) /*intersection tgte avec axe abscisses*/
12 fin_faire
13 renvoyer n
```

Problèmes

- On peut rencontrer des divisions par zéro.

Problèmes

- On peut rencontrer des divisions par zéro.
- La terminaison n'est pas assurée.

Problèmes

- On peut rencontrer des divisions par zéro.
- La terminaison n'est pas assurée.
- Même si un résultat est renvoyé, il peut être éloigné d'un *véritable* zéro de f .

Stabilité

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 0 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Par théorème, on sait qu'une fonction dérivable est convexe si et seulement si la courbe est au dessus de toutes les tangentes.

Stabilité

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 0 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Par théorème, on sait qu'une fonction dérivable est convexe si et seulement si la courbe est au dessus de toutes les tangentes.
- Comme les cordes sont au dessus de la courbe, les cordes sont au dessus des tangentes.

Stabilité

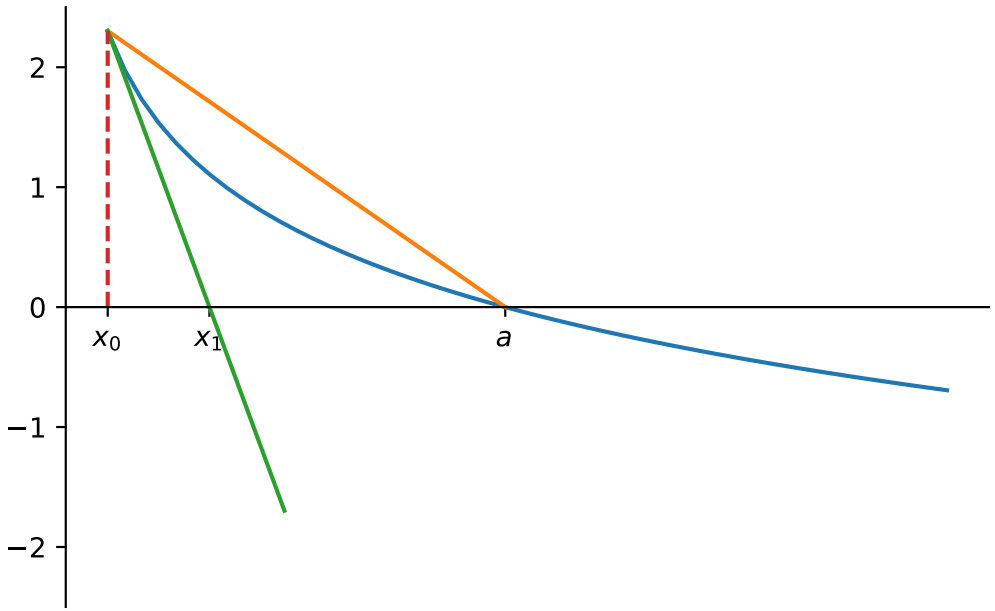
Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 0 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Par théorème, on sait qu'une fonction dérivable est convexe si et seulement si la courbe est au dessus de toutes les tangentes.
- Comme les cordes sont au dessus de la courbe, les cordes sont au dessus des tangentes.
- La tangente en x_0 est donc comprise entre la verticale $y = x_0$ et la corde qui joint $(x_0, f(x_0))$ à $(a, f(a))$. on en déduit que x_1 est entre x_0 et a (voir figure page suivante)

Stabilité

f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 0 ($f'(a) \neq 0$).

Figure – Les itérés restent dans I



Convergence quadratique locale

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Pour tout $x \in I$, la formule de Taylor-Lagrange donne l'existence de ξ tel que : $0 = f(a) = f(x) + f'(x)(a-x) + \frac{f''(\xi)}{2}(a-x)^2$, avec $\xi \in]a; x[$ donc $\xi \in I$.

  Donc $x - a = \dfrac{f(x)}{f'(x)} + \frac{f''(\xi)}{2f'(x)}(a-x)^2$ si $f'(x) \neq 0$.

Convergence quadratique locale

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Pour tout $x \in I$, la formule de Taylor-Lagrange donne l'existence de ξ tel que : $0 = f(a) = f(x) + f'(x)(a-x) + \frac{f''(\xi)}{2}(a-x)^2$, $\xi \in I$.
  Donc $x - a = \dfrac{f(x)}{f'(x)} + \dfrac{f''(\xi)}{2f'(x)}(a-x)^2$ si $f'(x) \neq 0$.
- Partant de $x_0 \in I$, au bout d'une itération :
  $x_1 - a = x_0 - \dfrac{f(x_0)}{f'(x_0)} - a = \dfrac{f^{(2)}(\xi)}{2\,f'(x_0)}(x_0 - a)^2$ avec $\xi \in I$.

Convergence quadratique locale

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Pour tout $x \in I$, la formule de Taylor-Lagrange donne l'existence de ξ tel que : $0 = f(a) = f(x) + f'(x)(a-x) + \frac{f''(\xi)}{2}(a-x)^2$, $\xi \in I$.
  Donc $x - a = \dfrac{f(x)}{f'(x)} + \frac{f''(\xi)}{2f'(x)}(a-x)^2$ si $f'(x) \neq 0$.
- Partant de $x_0 \in I$, au bout d'une itération :
  $x_1 - a = x_0 - \frac{f(x_0)}{f'(x_0)} - a = \frac{f^{(2)}(\xi)}{2\,f'(x_0)}(x_0 - a)^2$ avec $\xi \in I$.
- On pose $M = \max_{x \in I} |f^{(2)}(x)| \neq 0$ (sinon f' constante et f affine, Newton est sans intérêt) et $m = \min_{x \in I} |f'(x)| \neq 0$. La continuité de f', $f^{(2)}$ et l'aspect fermé de I entraîne l'existence de M, m. Posons $K = \dfrac{M}{2m}$. Alors $|x_1 - a| \leq K|x_0 - a|^2$.

Convergence quadratique locale

Hypothèse : f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- Pour tout $x \in I$, la formule de Taylor-Lagrange donne l'existence de ξ tel que : $0 = f(a) = f(x) + f'(x)(a-x) + \frac{f''(\xi)}{2}(a-x)^2$, $\xi \in I$.
  Donc $x - a = \dfrac{f(x)}{f'(x)} + \frac{f''(\xi)}{2f'(x)}(a-x)^2$ si $f'(x) \neq 0$.
- Partant de $x_0 \in I$, au bout d'une itération :
  $x_1 - a = x_0 - \frac{f(x_0)}{f'(x_0)} - a = \frac{f^{(2)}(\xi)}{2\,f'(x_0)}(x_0 - a)^2$ avec $\xi \in I$.
- On pose $M = \max_{x \in I} |f^{(2)}(x)| \neq 0$ (sinon f' constante et f affine, Newton est sans intérêt) et $m = \min_{x \in I} |f'(x)| \neq 0$. Posons $K = \dfrac{M}{2m}$.
  Alors $|x_1 - a| \leq K|x_0 - a|^2$.
- On choisit x_0 assez proche de a pour que $|x_0 - a| < \frac{1}{K}$. Alors $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$ (mais on le sait déjà).

Convergence quadratique locale

Hypothèse f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$.

Convergence quadratique locale

Hypothèse f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$.
- Par récurrence, les itérés de Newton vérifient $|x_k - a| \leq |x_0 - a|$ et donc $\forall \mathbf{k} \in \mathbb{N}$, $\mathbf{x_k} \in \mathbf{I}$ (on retrouve ainsi le résultat de **stabilité**).

Convergence quadratique locale

Hypothèse f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$.
- Par récurrence, les itérés de Newton vérifient $|x_k - a| \leq |x_0 - a|$ et donc $\forall \mathbf{k} \in \mathbb{N}$, $\mathbf{x_k} \in \mathbf{I}$ (on retrouve ainsi le résultat de **stabilité**).
- Reprenons Taylor-Lagrange, pour tout k on a $|x_{k+1} - a| \leq K|x_k - a|^2$ (c'est ce qu'on appelle *convergence quadratique locale* (CQL)).

Convergence quadratique locale

Hypothèse f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$.
- Par récurrence, les itérés de Newton vérifient $|x_k - a| \leq |x_0 - a|$ et donc $\forall \mathbf{k} \in \mathbb{N}$, $\mathbf{x_k} \in \mathbf{I}$ (on retrouve ainsi le résultat de **stabilité**).
- Reprenons Taylor-Lagrange, pour tout k on a $|x_{k+1} - a| \leq K|x_k - a|^2$ (c'est ce qu'on appelle *convergence quadratique locale* (CQL)).
- En utilisant 3 fois cette formule de CQL, on obtient :

$$|x_n - a| \leq K|x_{n-1} - a|^2 \leq K\left(K^2|x_{n-2} - a|^4\right) \leq K^3\left(K^4|x_{n-3} - a|^8\right)$$

Convergence quadratique locale

Hypothèse f est une fonction définie sur un intervalle I fermé borné centré en a, de classe $\mathcal{C}^2$ telle que $f^{(2)} \geq 0$ (fonction convexe). On suppose que a est un zéro de f d'ordre 1 ($f'(a) \neq 0$). Quitte à restreindre I, on suppose que f' ne s'annule pas sur I.

- $|x_1 - a| \leq |x_0 - a|$ et donc $x_1 \in I$.
- Par récurrence, les itérés de Newton vérifient $|x_k - a| \leq |x_0 - a|$ et donc $\forall \mathbf{k} \in \mathbb{N}$, $\mathbf{x_k} \in \mathbf{I}$ (on retrouve ainsi le résultat de **stabilité**).
- Reprenons Taylor-Lagrange, pour tout k on a $|x_{k+1} - a| \leq K|x_k - a|^2$ (c'est ce qu'on appelle *convergence quadratique locale* (CQL)).
- En utilisant 3 fois cette formule de CQL, on obtient :

$$|x_n - a| \leq K|x_{n-1} - a|^2 \leq K\left(K^2|x_{n-2} - a|^4\right) \leq K^3\left(K^4|x_{n-3} - a|^8\right)$$

- Par une simple récurrence, on trouve :

$$|x_n - a| \leq K^{2^n-1}|x_0 - a|^{2^n}.$$

Convergence de la suite des itérés vers a

Comme $|x_0 - a| < \frac{1}{K}$, posons $\mu = K|x_0 - a|$. On a donc $\mu < 1$. Il vient :

$$|x_n - a| \leq \frac{1}{K}\mu^{2^n} \underset{n\to+\infty}{\longrightarrow} 0.$$

Convergence de la suite des itérés vers a

Comme $|x_0 - a| < \frac{1}{K}$, posons $\mu = K\,|x_0 - a|$. On a donc $\mu < 1$. Il vient :

$$|x_n - a| \leq \frac{1}{K}\mu^{2^n} \underset{n\to+\infty}{\longrightarrow} 0.$$

On a donc montré qu'en prenant x_0 « assez » proche du zéro a, la suite des itérés reste dans un petit intervalle fixe autour de a et converge vers a.

Nombre d'itération

- On peut montrer que en cas de CQL, le « nombre de chiffres significatifs corrects » des itérés double en gros (définir « en gros ») à chaque itération.

Nombre d'itération

- On peut montrer que en cas de CQL, le « nombre de chiffres significatifs corrects » des itérés double en gros (définir « en gros ») à chaque itération.
- Or, le nombre de chiffres significatifs représentables par un ordinateur est limité (environ 15 chiffres décimaux sur un ordinateur avec un processeur 32-bits).

Nombre d'itération

- On peut montrer que en cas de CQL, le « nombre de chiffres significatifs corrects » des itérés double en gros (définir « en gros ») à chaque itération.
- Or, le nombre de chiffres significatifs représentables par un ordinateur est limité (environ 15 chiffres décimaux sur un ordinateur avec un processeur 32-bits).
- Donc, en 10 itérations maximum, on est arrivé au nombre représentable le plus proche possible du zéro.

Nombre d'itération

- On peut montrer que en cas de CQL, le « nombre de chiffres significatifs corrects » des itérés double en gros (définir « en gros ») à chaque itération.
- Or, le nombre de chiffres significatifs représentables par un ordinateur est limité (environ 15 chiffres décimaux sur un ordinateur avec un processeur 32-bits).
- Donc, en 10 itérations maximum, on est arrivé au nombre représentable le plus proche possible du zéro.
- Si la contrainte de sortie de boucle n'est pas vérifiée, c'est vraisemblablement que la suite des itérés diverge et donc qu'on n'a pas pris x_0 assez proche du véritable zéro.

Nombre d'itération

- On peut montrer que en cas de CQL, le « nombre de chiffres significatifs corrects » des itérés double en gros (définir « en gros ») à chaque itération.
- Or, le nombre de chiffres significatifs représentables par un ordinateur est limité (environ 15 chiffres décimaux sur un ordinateur avec un processeur 32-bits).
- Donc, en 10 itérations maximum, on est arrivé au nombre représentable le plus proche possible du zéro.
- Si la contrainte de sortie de boucle n'est pas vérifiée, c'est vraisemblablement que la suite des itérés diverge et donc qu'on n'a pas pris x_0 assez proche du véritable zéro.
- Enfonçons le clou : si l'itéré initial x_0 n'est pas pris suffisamment proche d'un zéro, la suite des itérés générée par l'algorithme a un comportement erratique, dont la convergence éventuelle ne peut être que le fruit du hasard (un des itérés est par chance proche d'un zéro).

Approximation de la dérivée

Parfois, on ne dispose pas de (ou on ne veut pas calculer) la dérivée.

- $f'(x) \simeq \dfrac{f(x+h)-f(x)}{h}$ (Approximation linéaire, DL1).

Approximation de la dérivée

Parfois, on ne dispose pas de (ou on ne veut pas calculer) la dérivée.

- $f'(x) \simeq \dfrac{f(x+h)-f(x)}{h}$ (Approximation linéaire, DL1).
- **Approximation par taux d'accroissement** Si f de classe $\mathcal{C}^2$, par la formule de Taylor-Young :

$$\begin{array}{rcl} f(x_0+h) & = & f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+o_0\left(h^2\right) \\ \frac{f(x_0+h)-f(x_0)}{h}-f'(x_0) & = & \frac{h}{2}f''(x_0)+o_0(h) \end{array}$$

  Incertitude linéaire.

Approximation de la dérivée

Parfois, on ne dispose pas de (ou on ne veut pas calculer) la dérivée.

- $f'(x) \simeq \dfrac{f(x+h)-f(x)}{h}$ (Approximation linéaire, DL1).
- **Approximation par taux d'accroissement** Si f de classe $\mathcal{C}^2$, par la formule de Taylor-Young :

$$\begin{aligned} f(x_0+h) &= f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+o_0\left(h^2\right) \\ \frac{f(x_0+h)-f(x_0)}{h}-f'(x_0) &= \frac{h}{2}f''(x_0)+o_0(h) \end{aligned}$$

  Incertitude linéaire.
- **Approximation par dérivée symétrique** Si f de classe $\mathcal{C}^3$, par la formule de Taylor-Young (observer l' incertitude quadratique) :

$$\begin{aligned} f(x_0+h) &= f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3f^{(3)}(x_0)}{6}+o_0\left(h^3\right) \\ f(x_0-h) &= f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)-\frac{h^3f^{(3)}(x_0)}{6}+o_0\left(h^3\right) \\ f(x_0+h)-f(x_0-h) &= 2hf'(x_0)+\frac{2h^3f^{(3)}(x_0)}{6}+o_0\left(h^3\right) \\ \frac{f(x_0+h)-f(x_0-h)}{2h}-f'(x_0) &= \frac{h^2f^{(3)}(x_0)}{6}+o_0\left(h^2\right) \end{aligned}$$

Dérivation discrète

- Deux méthodes de dérivations discrète :

```
1 #donne une approx. de la dérivée de f en x0
2 #prendre h petit
3 def derivee(f, x0, h):
4     """taux d'accroissement"""
5     return (f(x0+h) - f(x0)) / h
6
7 def derivee_symetrique(f, x0, h):
8     """dérivée symétrique"""
9     return (f(x0+h) - f(x0-h)) / (2*h)
```

Code python

- Code basique (aucune vérification des hypothèses de convergence) :

```
1 #fp : la dérivée de f
2 def newton(f, fp, x0, epsilon):
3     u = x0
4     v = u - f(u)/fp(u)
5     while abs(v-u) > epsilon:
6         u, v = v, v - f(v)/fp(v)
7     return v
```

Code python

- Code basique (aucune vérification des hypothèses de convergence) :

```
1 #fp : la dérivée de f
2 def newton(f, fp, x0, epsilon):
3     u = x0
4     v = u - f(u)/fp(u)
5     while abs(v-u) > epsilon:
6         u, v = v, v - f(v)/fp(v)
7     return v
```

- Utilisation réelle : vérifier que u, v ne prennent pas de trop grandes valeurs (sortie de l'intervalle d'étude) ou que $|f'(v)|$ n'est pas trop petite (division par un proche de zéro).

Trace d'éxecution

Trace d'exécution :

```
>>> import math
>>> newton(math.sin, math.cos, 3, 10**(-3))
3.141592653300477
>>> newton(lambda x : x**2-2, lambda x : 2*x, 2., 10**(-2))
1.4142156862745099
```

Recherche de zéro d'une fonction : idées générales

- Résoudre $f(x) = 0$ ou équivalent $x = g(x)$.
- Plusieurs méthodes disponibles dans `scipy.optimize`.
- Méthode de point fixe, Newton, Brent, méthode générale.

Point fixe $x = g(x)$

- Itération explicite : $x_{n+1} = g(x_n)$.
- Peut être codée manuellement.
- Convergence non garantie, dépend de g.

```
x = x0
for i in range(max_iter):
    x_new = g(x)
    if abs(x_new - x) < tol:
        break
    x = x_new
```

scipy.optimize.root

- Méthode générale pour $f(x) = 0$.
- Plusieurs algorithmes au choix (hybride par défaut).
- Supporte aussi les systèmes vectoriels.

```
1 from scipy.optimize import root
2
3 res = root(f, x0)
4 print(res.x)
```

scipy.optimize.newton

- Implémentation de Newton-Raphson.
- Supporte aussi la méthode de la sécante (si dérivée non fournie).
- Très rapide quand $f'(x)$ est connue.

```
1 from scipy.optimize import newton
2
3 # Avec dérivée
4 x_sol = newton(f, x0, fprime=f_prime)
5
6 # Sans dérivée (méthode de la sécante)
7 x_sol = newton(f, x0)
```

scipy.optimize.brentq

- Méthode robuste (Brent) pour $f(x) = 0$ sur un intervalle $[a, b]$.
- Nécessite un changement de signe : $f(a) \times f(b) < 0$.
- Très fiable, pas besoin de dérivée.

```
1 from scipy.optimize import brentq
2
3 x_sol = brentq(f, a, b)
```

Résumé des choix

- `point fixe` : simple mais convergence incertaine.
- `root` : méthode générale pour $f(x) = 0$, même pour systèmes.
- `newton` : très rapide si dérivée connue.
- `brentq` : très fiable dans un intervalle connu, pas besoin de dérivée.

Pas de méthode universelle de résolution

- Si on cherche la simplicité et la robustesse, la méthode dichotomique s'impose : la continuité de la fonction et un premier intervalle $[a, b]$ tel que $f(a)f(b) \leq 0$ sont suffisants. La $\ll$ lenteur $\gg$ de cette méthode est toute relative quand on veut seulement quelques décimales.

Pas de méthode universelle de résolution

- Si on cherche la simplicité et la robustesse, la méthode dichotomique s'impose : la continuité de la fonction et un premier intervalle $[a, b]$ tel que $f(a)f(b) \leq 0$ sont suffisants. La « lenteur » de cette méthode est toute relative quand on veut seulement quelques décimales.
- Si on connaît une bonne approximation d'un zéro et qu'on cherche la rapidité, alors on peut appliquer une méthode de la famille de Newton : la méthode de Newton proprement dite si on connaît f' ou qu'on décide de l'approcher numériquement ; la méthode de la sécante (voir TD) si on ne peut/veut pas évaluer f' ou encore la méthode de Halley (qui est cubique) si on connaît f''.

Pas de méthode universelle de résolution

- Si on cherche la simplicité et la robustesse, la méthode dichotomique s'impose : la continuité de la fonction et un premier intervalle $[a, b]$ tel que $f(a)f(b) \leq 0$ sont suffisants. La « lenteur » de cette méthode est toute relative quand on veut seulement quelques décimales.
- Si on connaît une bonne approximation d'un zéro et qu'on cherche la rapidité, alors on peut appliquer une méthode de la famille de Newton : la méthode de Newton proprement dite si on connaît f' ou qu'on décide de l'approcher numériquement ; la méthode de la sécante (voir TD) si on ne peut/veut pas évaluer f' ou encore la méthode de Halley (qui est cubique) si on connaît f''.
- Dans des situations intermédiaires, on peut pratiquer des méthodes mixtes, avec une première phase de localisation de zéros par dichotomie, puis des itérations de Newton. Si ces itérations de Newton ne semblent pas converger, on peut reprendre une phase de dichotomie ; etc.

Comparaison avec Numpy

La méthode de Newton est programmée dans scipy.optimize.newton. À noter que si on ne donne pas la dérivée, c'est en fait la méthode de la sécante (voir TD) qui est appliquée.

```
>>> import scipy.optimize
>>> scipy.optimize.newton(math.sin, 3, math.cos)# Newton
3.141592653589793
>>> math.pi
3.141592653589793
>>> scipy.optimize.newton(math.sin, 3)# secante
3.141592653589793
```