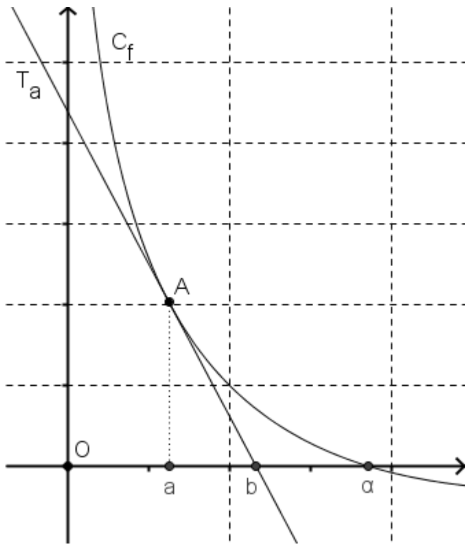
# La méthode de Newton

## 1) Position du problème

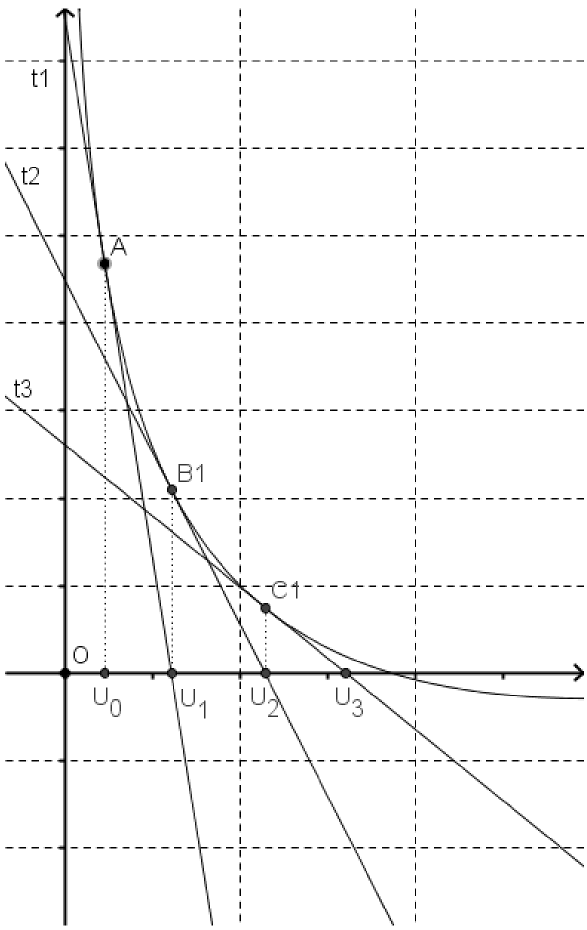
$f$ est une fonction dérivable sur un intervalle I.
L'équation $f(x) = 0$ admet une racine unique $\alpha$ sur l'intervalle I.
**La méthode**




Soit $a \in I$ une valeur approchée (grossière) de $\alpha$.
On va utiliser l'approximation affine $g$ de $f$ au point $a$.
On aura donc $g(x) = f(a) + (x - a)f'(a)$ (tangente $T_a$).
La droite $T_a$ coupe l'axe des abscisses en $b = a - \dfrac{f(a)}{f'(a)}$.
Sous certaines conditions, le nombre $b$ peut représenter une meilleure approximation de $\alpha$ que $a$.
La méthode de Newton consiste à itérer le processus en repartant de $b$ et ainsi de suite.

## 2) Visualisation avec Geogebra

On cherche à résoudre l'équation $f(x) = 0$ avec $f(x) = x - 3\ln x$



# 3) Mise en place de la suite récurrente

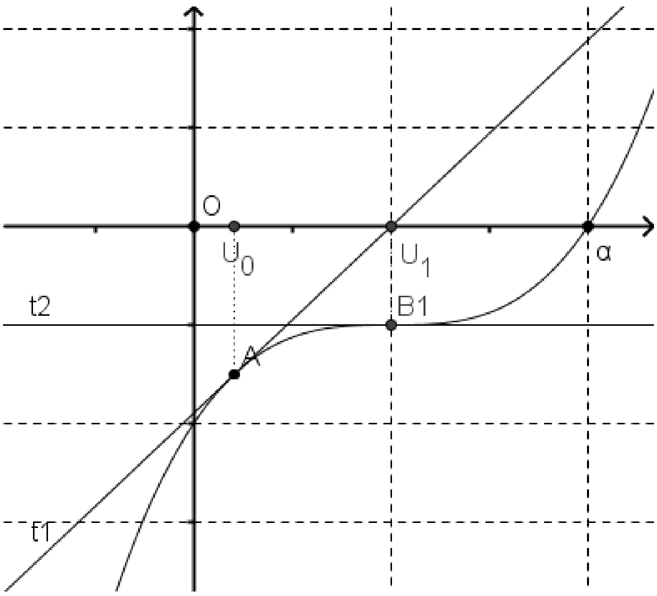
$u_{n+1} = u_n - \dfrac{f(u_n)}{f'(u_n)}$ et $u_0 = a$ avec $a \in I$

## a) Un cas favorable

Lorsque $f$ est monotone sur l'intervalle I et que $f$ et $f''$ sont de même signe entre $a$ et $\alpha$. On obtient dans ce cas une suite monotone.
En effet si on appelle $g$ la fonction sur laquelle est basée la récurrence, on a $g(x) = x - \dfrac{f(x)}{f'(x)}$ et la dérivée de $g$ s'écrit : $g' = 1 - \dfrac{(f')^2 - f \times f''}{(f')^2} = \dfrac{f \times f''}{(f')^2}$ et si $f$ et $f''$ sont de même signe alors $g' > 0$ et $g$ est croissante.
Dans ce cas $u_n - u_{n-1}$ et $g(u_n) - g(u_{n-1}) = u_{n+1} - u_n$ sont de même signe et la suite $(u_n)$ est monotone.

## b) Un cas ennuyeux

Lorsqu'il existe un point d'inflexion entre $a$ et $\alpha$, la suite risque d'être à croissance alternée (possibilité d'encadrement de $\alpha$) mais la convergence vers $\alpha$ n'est pas assurée.
Par exemple si on prend $f(x) = (x-1)^3 - 1$ alors on a $\alpha = 2$ mais la suite définie par $u_0 = 1 - \dfrac{1}{2^{\frac{1}{3}}} \approx 0.21$ et $u_{n+1} = g(u_n)$ ne sera plus définie pour $n \geq 2$, comme le montre la figure, car la tangente en B1 à la courbe est parallèle à l'axe des abscisses.




## c) Le problème du calcul de la dérivée

Si on ne dispose pas d'un logiciel de calcul formel l'obtention de l'expression de $f'(x)$ sera impossible. Deux possibilités s'offriront à nous :

- Adapter l'algorithme à chaque exemple à traiter.
- Utiliser une valeur approchée de $f'(x)$.

# 4) Un algorithme adapté : calcul de $\sqrt{a}$

Il s'agit de résoudre $f(x) = 0$ avec $f(x) = x^2 - a$ et $f'(x) = 2x$

On obtient $u_{n+1} = u_n - \dfrac{(u_n)^2 - a}{2u_n} = \dfrac{1}{2}\left(u_n + \dfrac{a}{u_n}\right)$

En posant $v_n = \dfrac{a}{u_n}$ et $u_{n+1} = \dfrac{1}{2}(u_n + v_n)$ on reconnait la méthode de Héron

Pour plus de détails voir :
http://www.irem.univ-mrs.fr/IMG/pdf/BabyloneV2-2.pdf
http://fr.wikipedia.org/wiki/Méthode_de_héron#Motivation_géométrique

## 5) Calcul d'une valeur approchée de $f'(u_n)$

### a) Une première idée

La première idée est de prendre $f'(u_n) \approx \dfrac{f(u_n+h)-f(u_n)}{h}$ avec $h$ petit.
Mais il faut adapter la valeur de $h$ choisie à chaque cas (pente de la fonction, précision souhaitée).

**Algorithme**

- La suite étant monotone, on ne peut pas obtenir d'encadrement de la racine $\alpha$, il faut donc utiliser un test d'arrêt de la forme $|u_{n+1}-u_n|<\varepsilon$ pour $\varepsilon$ donné.
- $f,\ a,\ h$ et la précision $\varepsilon$ sont donnés
- $L$ désigne la liste des termes de la suite
- $n$ désigne le nombre d'itérations nécessaires pour atteindre la précision $\varepsilon$.
- L'initialisation avec $c=a+1$ n'a pour seul but que de faire démarrer la boucle While.
- L'évaluation répétée de $f(a)$ pouvant ralentir les calculs, cette valeur est stockée dans la variable $t$ et ce calcul n'est effectué qu'une seule fois.

$L \longleftarrow \{a\}$
$n \longleftarrow 0$
$c \longleftarrow a+1$
**Tant que** $abs(c-a)>\varepsilon$ **faire**
  $c \longleftarrow a$
  $y \longleftarrow f(a)$
  $a \longleftarrow a-\dfrac{h\times y}{f(a+h)-y}$
  $L \longleftarrow L$ suivie de $a$
  $n \longleftarrow n+1$
Afficher $n$
Afficher $L$

**Programme sur TI 82 Stats**

La fonction $f$ a d'abord été saisie dans la variable $\mathtt{Y_1}$ du menu $\boxed{f(x)}$ de la calculatrice.

```
PROGRAM : NEWTON
:Input "U0=",A
:Input "H=",H
:Input "PRECISION=",E
:ClrList L₁
:{B}→L₁
:0→N
:A+1→C
:While Abs(C-A)>E
:C→A
:Y₁(A)→Y
:A-H*Y/(Y₁(A+H)-Y)→B
:Chaîne(L₁,{B})→L₁
:N+1→N
:End
:Disp "N=",N
:Disp L₁
```

**Remarque**

La rapidité de convergence de la méthode de Newton est de type quadratique et on a $|u_{n+1}-\alpha| \sim k\ |u_n-\alpha|^2$.

Pour plus de détails voir : http://fr.wikipedia.org/wiki/Méthode_de_newton#Convergence et le document intitulé Evaluation expérimentale de la rapidité de convergence d'une suite
Auteur : Henri ROLAND Groupe algorithmique de l'IREM d'Aix-Marseille

### b) Une autre approximation de la dérivée

Si la suite converge vers $\alpha$, alors pour $n$ assez grand, $u_n$ est voisin de $u_{n-1}$ et on peut écrire : $f'(u_n) \approx \dfrac{f(u_n) - f(u_{n-1})}{u_n - u_{n-1}}$
La formule de récursion s'écrit alors :

- $u_{n+1} = u_n - \dfrac{(u_n - u_{n-1})\, f(u_n)}{f(u_n) - f(u_{n-1})} = \dfrac{u_{n-1} f(u_n) - u_n f(u_{n-1})}{f(u_n) - f(u_{n-1})}$

On retrouve la formule de récurrence habituellement associée à la méthode de la sécante
Pour plus de détails voir : http://fr.wikipedia.org/wiki/Méthode_de_la_sécante

**Algorithme**

- $u_0 = a$ et $u_1 = b$ sont donnés.
- L'évaluation répétée de $f(a)$ et de $f(b)$ pouvant ralentir les calculs, ces valeurs sont stockées dans les variable $t$ et $z$ et ce calcul n'est effectué qu'une seule fois à chaque itération.

$n \longleftarrow 0$
$t \longleftarrow f(a)$
$z \longleftarrow f(b)$
$L \longleftarrow \{a\}$
**Tant que** $abs(b-a) > \varepsilon$ **faire**
  $c \longleftarrow a - \dfrac{(a-b) * t}{t - z}$
  $L \longleftarrow L$ suivie de $c$
  $b \longleftarrow a$
  $a \longleftarrow c$
  $z \longleftarrow t$
  $t \longleftarrow f(a)$
  $n \longleftarrow n+1$
Afficher $n$
Afficher $L$

**Programme sur TI 82 Stats**

```
PROGRAM : NEWTON2
:Input "U0=",A
:Input "U1=",B
:Input "PRECISION=",E
:0→N
:Y₁(A)→T:Y₁(B)→Z
:ClrList L₁
:{A}→L₁
:While Abs(B-A)>E
:A-(A-B)*T)/(T-Z)→C
:Chaîne(L₁,{C})→L₁
:A→B:C→A
:T→Z:Y1(A)→T
:N+1→N
:End
:Disp "N=",N
:Disp L₁
```

**Remarque**
La rapidité de convergence est alors un peu moins bonne et on a $|u_{n+1} - \alpha| \sim k\ |u_n - \alpha|^{\Phi}$ avec $\Phi = \dfrac{\sqrt{5}+1}{2} \simeq 1.618$ mais meilleure que la méthode de la corde pour laquelle on a $|u_{n+1} - \alpha| \sim k\ |u_n - \alpha|$
Pour plus de détails voir le document intitulé Méthode de la corde

Auteur : Henri ROLAND Groupe algorithmique de l'IREM d'Aix-Marseille