Chapitre de révision 2

Recherche de racines

2.1 Recherche par dichotomie

2.1.1 Recherche d'une racine par dichotomie

Théorème 1 *Soit f une application réelle continue sur l'intervalle $[a,b]$, et telle que :*

$$f(a).f(b) \leqslant 0$$

Alors f admet une racine sur l'intervalle $[a,b]$
(c.à.d. $\exists\, x \in [a,b]$, $f(x) = 0$).

• Méthode de recherche d'une racine par dichotomie (à e près) :

```
TANT QUE b-a > e:
    m = (a+b) / 2
    SI f(a) * f(m) <= 0 ALORS :
        a, b = a, m    # Dichotomie à gauche
    SINON :
        a, b = m, b    # Dichotomie à droite
FIN TANT QUE
RETOURNE (a+b)/2
```

on peut aussi retourner une valeur par défaut : `a`.
on peut aussi retourner une valeur par excès : `b`.

2.1.2 Recherche d'une racine par dichotomie

• Code `python` :

```
def dichotomie(f,a,b,e):
    assert f(a) * f(b) <= 0
    while b-a > e:
        m = (a+b)/2
        if f(a)*f(m) <= 0:
            a, b = a, m    # Dichotomie à gauche
        else:
            a, b = m, b    # Dichotomie à droite
    return (a+b)/2
```

• Exemple :

```
>>> f = lambda x : x**2-2
>>> dichotomie(f,0,3,0.001)
1.4139404296875
```

2.1.3 Recherche d'une racine par dichotomie

• Quand utiliser une dichotomie pour la recherche d'une racine ?

1. Sous une hypothèse de régularité faible de la fonction f : continuité,
2. Recherche sur l'intervalle $[a,b]$ lorsque $f(a)$ et $f(b)$ sont de signes opposés.
3. Lorsqu'on ne dispose que du tableau des valeurs de f sur une subdivision de l'intervalle.

• L'algorithme est de complexité du même ordre que le nombre de passage dans la boucle `while` : $\approx \log_2\left(\dfrac{b-a}{e}\right)$.

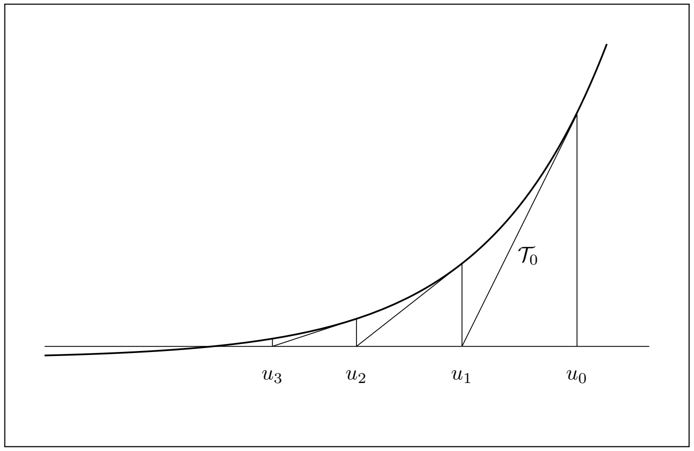
2.2 Méthode de Newton

2.2.1 Recherche d'une racine : méthode de Newton

• La méthode de Newton (ou de Newton-Raphson) est une méthode célèbre pour la résolution approchée d'une équation $f(x) = 0$.
• Elle considère la suite définie par son premier terme u_0 et par :

$$\boxed{u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}}$$

Obtenue géométriquement du graphe de f à partir des tangentes et de leurs intersections avec $[O,x)$:



qui sous certaines hypothèses converge vers une racine de la fonction f.

2.2.2 Recherche d'une racine : méthode de Newton

Théorème 2 (Méthode de Newton) *Soit $f : [a,b] \longrightarrow \mathbb{R}$ une application de classe C^1 et x_0 une solution de l'équation*

$$(E) : \; f(x) = 0$$

en laquelle $f'(x_0) \neq 0$.
• *Alors il existe $r > 0$ tel que dans l'intervalle $I =]x_0 - r; x_0 + r[$ l'équation (E) ait pour solution unique x_0, et de plus la suite $(u_n)_{n\in\mathbb{N}}$ définie par :*

$$u_0 \in I \quad \forall n \in \mathbb{N}, \quad u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}$$

est convergente et a pour limite cette racine x_0.

• *Si de plus f est de classe C^2 (ou mieux), alors la convergence est quadratique, c.à.d., $\exists\, k > 0$, lorsque n est suffisamment grand, en notant $r_n = |x_0 - u_n|$ le reste au rang n :*

$$r_{n+1} \leqslant k.r_n^2$$

2.2.3 Recherche d'une racine : méthode de Newton

On prend comme critère d'arrêt un nombre n d'itérations (calcul de u_n) ou bien $u_{n+1} - u_n < 10^{-N}$.

Il faut savoir calculer la fonction dérivée, et avoir démontré que pour une valeur u_0 la suite (u_n) est bien définie (pas de division par zéro) et converge.

```
def newton(f, g, u0, e):
    u = u0
    v = u0 - f(u0)/g(u0)
    while abs(v-u) > e:
        u = v
        v = v - f(v)/g(v)
    return v
```

• Lorsque f est <u>strictement monotone</u> on peut retourner une valeur approchée à e près, en prenant comme critère d'arrêt :

```
while f(v-e)*f(v+e) > 0:
```

(à l'arrêt f change de signe sur $[u_n - e; u_n + e]$, et par continuité y admet donc une racine $\Longrightarrow u_n$ est une valeur approchée d'une racine à e près.)

2.2.4 Recherche d'une racine de fonction réelle

On dispose de deux méthodes efficaces, à savoir programmer :

1. **Par dichotomie**
   (a) Avantages : hypothèse de régularité de la fonction faible : continuité.
   Peut s'appliquer à un tableau des valeurs de f.
   Peut retourner à e près une valeur approchée, par défaut, par excès.
   (b) Désavantages : $f(a)$ et $f(b)$ doivent être de signes opposés.
   Assez lent.
2. **Méthode de Newton**
   (a) Lorsqu'elle converge, extrêmement rapide si la fonction est de classe C^2.
   (b) Désavantage : hypothèse de régularité fortes C^1 ou C^2.
   <u>Convergence non certaine</u> : seulement lorsque le point initial est suffisamment proche de la racine.
   Il faut connaitre la fonction ET sa dérivée. Ne s'applique pas directement à un tableau.