Mathématiques concrètes

Vers l'infini et au-delà...

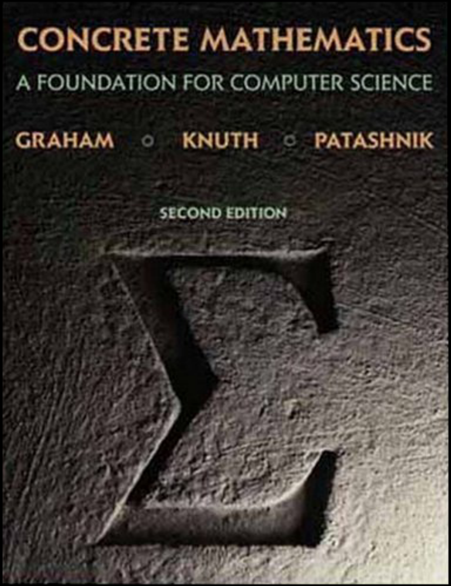
Guillaume CONNAN

IUT de Nantes - Dpt d'informatique

12 décembre 2011

Sommaire

Sommaire


CONCRETE MATHEMATICS
A FOUNDATION FOR COMPUTER SCIENCE
GRAHAM ○ KNUTH ○ PATASHNIK
SECOND EDITION

```python
def divise(liste):
    n = len(liste)
    if n == 0:
        return [],[]
    return liste[: n//2], liste[n//2:]

def fusionner(L1,L2):
    if len(L1) == 0:
        return L2
    if len(L2) == 0:
        return L1
    if L1[0] < L2[0]:
        return ([L1[0]] + fusionner(L1[1:],L2))
    else:
        return ([L2[0]] + fusionner(L2[1:],L1))

def tri_f(L):
    if len(L) == 1:
        return L
    return fusionner(tri_f(divise(L)[0]),tri_f(divise(L)[1]))
```

$$2nc\lfloor\log_2(n)\rfloor \leqslant K(n) \leqslant 4nc\left(\lfloor\log_2(n)\rfloor+1\right)$$

Par exemple, si on prend $c=1$ et qu'on travaille sur une liste de taille 10 000, alors :

$$265\ 600 = 20\ 000 \times 13{,}28 \leqslant K(10\ 000) \leqslant 40000(13{,}28+1) = 571\ 200$$

Comparaison tri par insertion.

$$2nc\lfloor\log_2(n)\rfloor \leqslant K(n) \leqslant 4nc\left(\lfloor\log_2(n)\rfloor + 1\right)$$

Par exemple, si on prend $c = 1$ et qu'on travaille sur une liste de taille 10 000, alors :

$$265\ 600 = 20\ 000 \times 13,28 \leqslant K(10\ 000) \leqslant 40000(13,28+1) = 571\ 200$$

Comparaison tri par insertion.

$$2nc\lfloor \log_2(n)\rfloor \leqslant K(n) \leqslant 4nc\left(\lfloor \log_2(n)\rfloor + 1\right)$$

Par exemple, si on prend $c = 1$ et qu'on travaille sur une liste de taille 10 000, alors :

$$265\ 600 = 20\ 000 \times 13,28 \leqslant K(10\ 000) \leqslant 40000(13,28+1) = 571\ 200$$

Comparaison tri par insertion.

$$2nc\lfloor \log_2(n)\rfloor \leqslant K(n) \leqslant 4nc\left(\lfloor \log_2(n)\rfloor + 1\right)$$

Par exemple, si on prend $c = 1$ et qu'on travaille sur une liste de taille 10 000, alors :

$$265\ 600 = 20\ 000 \times 13,28 \leqslant K(10\ 000) \leqslant 40000(13,28 + 1) = 571\ 200$$

Comparaison tri par insertion.

Sommaire

Brooks's Law [prov.]
« Adding manpower to a late software project makes it later » – a result of the fact that the expected advantage from splitting work among N programmers is $O(N)$, but the complexity and communications cost associated with coordinating and then merging their work is $O(N^2)$

in « The New Hacker's Dictionnary »
http ://outpost9.com/reference/jargon/jargon_17.html#SEC24

Définition 1 (« Grand O »)

Soit f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{R}$. On dit que f est un « grand O » de g et on note $f = O(g)$ ou $f(n) = O(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \leqslant C|g(n)|$ pour tout $n \in \mathbb{N}$.

Pour le tri fusion, on avait obtenu pour $n > 1$:

$$K(n) \leqslant 4nc(\lfloor \log_2(n) \rfloor + 1)$$

Définition 1 (« Grand O »)

Soit f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{R}$. On dit que f est un « grand O » de g et on note $f = O(g)$ ou $f(n) = O(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \leqslant C|g(n)|$ pour tout $n \in \mathbb{N}$.

Pour le tri fusion, on avait obtenu pour $n > 1$:

$$K(n) \leqslant 4nc\left(\lfloor \log_2(n) \rfloor + 1\right)$$

Définition 1 (« Grand O »)

Soit f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{R}$. On dit que f est un « grand O » de g et on note $f = O(g)$ ou $f(n) = O(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \leqslant C|g(n)|$ pour tout $n \in \mathbb{N}$.

Pour le tri fusion, on avait obtenu pour $n > 1$:

$$K(n) \leqslant 4nc\left(\lfloor \log_2(n) \rfloor + 1\right)$$

Définition 2 (« Grand Oméga »)

Soit f et g deux fonctions de $\mathbb{R}$ dans lui-même. On dit que f est un « grand Oméga » de g et on note $f = \Omega(g)$ ou $f(n) = \Omega(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \geqslant C|g(n)|$ pour tout $n \in \mathbb{N}^*$.

Tri fusion : $2cn\lfloor \log_2(n) \rfloor \leqslant K(n)$.

Définition 2 (« Grand Oméga »)

Soit f et g deux fonctions de $\mathbb{R}$ dans lui-même. On dit que f est un « grand Oméga » de g et on note $f = \Omega(g)$ ou $f(n) = \Omega(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \geqslant C|g(n)|$ pour tout $n \in \mathbb{N}^*$.

Tri fusion : $2cn\lfloor \log_2(n) \rfloor \leqslant K(n)$.

Définition 2 (« Grand Oméga »)

Soit f et g deux fonctions de $\mathbb{R}$ dans lui-même. On dit que f est un « grand Oméga » de g et on note $f = \Omega(g)$ ou $f(n) = \Omega(g(n))$ si, et seulement si, il existe une constante positive C telle que $|f(n)| \geqslant C|g(n)|$ pour tout $n \in \mathbb{N}^*$.

Tri fusion : $2cn\lfloor \log_2(n) \rfloor \leqslant K(n)$.

Définition 3 (« Grand Théta »)

$$f = \Theta(g) \iff \begin{cases} f = O(g) \\ f = \Omega(g) \end{cases}$$

coût \ n	100	1 000	10^6	10^9
$\log_2(n)$	≈ 7	≈ 10	≈ 20	≈ 30
$n\log_2(n)$	≈ 665	$\approx 10\ 000$	$\approx 2\cdot 10^7$	$\approx 3\cdot 10^{10}$
n^2	10^4	10^6	10^{12}	10^{18}
n^3	10^6	10^9	10^{18}	10^{27}
2^n	$\approx 10^{30}$	$> 10^{300}$	$> 10^{10^5}$	$> 10^{10^8}$

L'âge de l'Univers est environ de 10^{18} secondes...

coût \ n	100	1 000	10^6	10^9
$\log_2(n)$	≈ 7	≈ 10	≈ 20	≈ 30
$n\log_2(n)$	≈ 665	$\approx 10\ 000$	$\approx 2 \cdot 10^7$	$\approx 3 \cdot 10^{10}$
n^2	10^4	10^6	10^{12}	10^{18}
n^3	10^6	10^9	10^{18}	10^{27}
2^n	$\approx 10^{30}$	$> 10^{300}$	$> 10^{10^5}$	$> 10^{10^8}$

L'âge de l'Univers est environ de 10^{18} secondes...

Définition 4 (« Petit o »)

$f = o(g)$ si, et seulement si, pour toute constante positive ε, il existe un entier n_0 tel que, pour tout $n \geqslant n_0$, $\left|f(n)\right| \leqslant \varepsilon \left|g(n)\right|$

Définition 5 (Fonctions équivalentes)

$$f(n) \sim g(n) \Longleftrightarrow f(n) = g(n) + o(g(n))$$

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f + g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ si k est une constante ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f+g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ si k est une constante ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f + g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ si k est une constante ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f+g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ si k est une constante ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f+g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ si k est une constante ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f + g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ *si k est une constante* ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Théorème 6 (Manipulation des O)

- $O(f) + O(g) = O(f+g)$;
- $f = O(f)$;
- $k \cdot O(f) = O(f)$ *si* k *est une constante* ;
- $O(O(f)) = O(f)$;
- $O(f) \cdot O(g) = O(f \cdot g)$;
- $O(f \cdot g) = f \cdot O(g)$.

Approximations asymptotiques quand $n \to +\infty$ ou quand $x \to 0$

$$n! = \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)\left(1 + \frac{1}{12n} + \frac{1}{288n^2} + O\frac{1}{n^3}\right)$$

$$\mathrm{e}^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + O(x^5)$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + O(x^5)$$

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + O(x^5)$$

$$(1+x)^\alpha = 1 + \alpha x + \binom{\alpha}{2}x^2 + \binom{\alpha}{3}x^3 + \binom{\alpha}{4}x^4 + O(x^5)$$

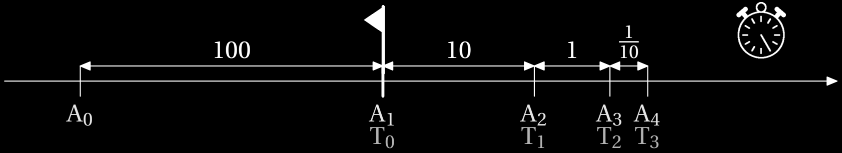
Sommaire

Sommaire

Sommaire

$$(1+x)^n > 1+nx$$

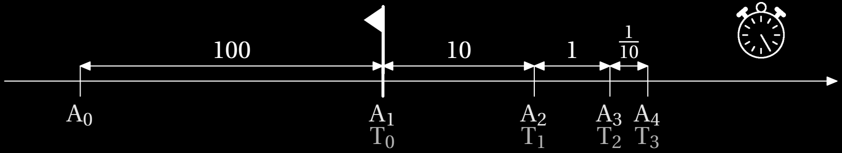
Sommaire



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

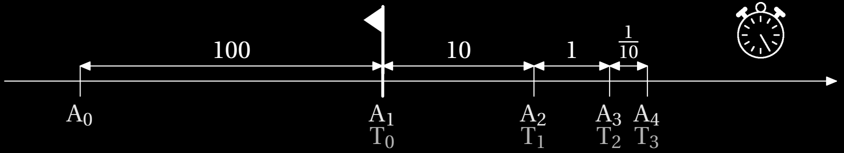
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}\left(T_{n-1} - A_{n-1}\right) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

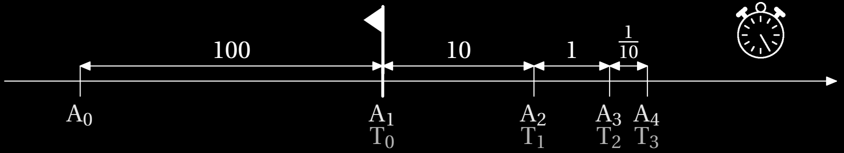
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}\left(T_{n-1} - A_{n-1}\right) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

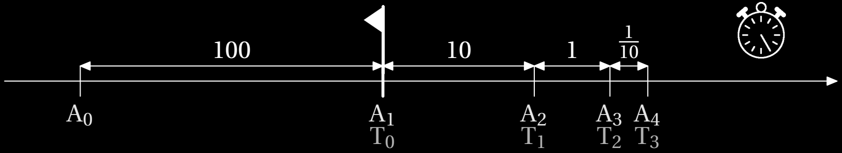
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}\left(T_{n-1} - A_{n-1}\right) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

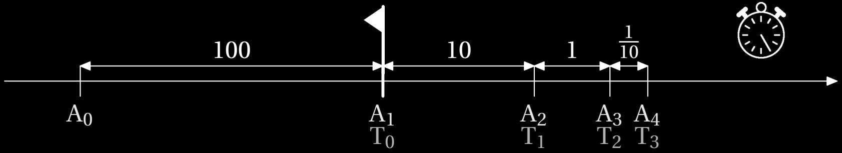
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

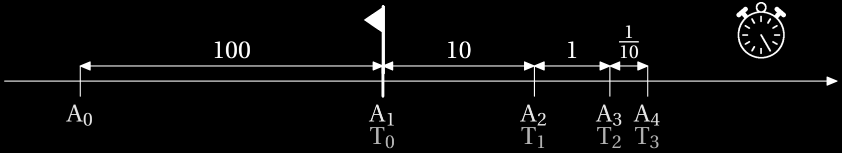
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

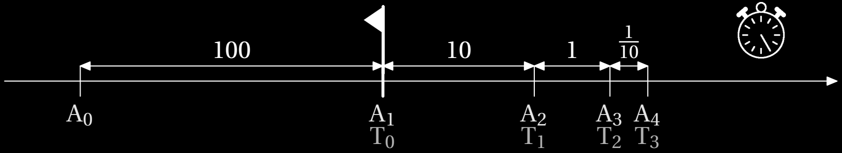
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

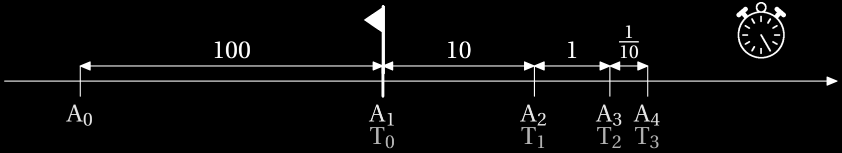
$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$



$$d_n = T_n - A_n = T_n - T_{n-1} = \frac{1}{10}(T_{n-1} - A_{n-1}) = \frac{1}{10} d_{n-1}$$

$$t_1 = \frac{d_1}{v} = \frac{100}{v}$$

$$t_2 = \frac{d_2}{v} = \frac{1}{10}\frac{d_1}{v} = \frac{1}{10} t_1$$

Théorème 7 (Série géométrique)

La suite de terme général

$$S_n = \sum_{k=0}^{n} q^n$$

converge si, et seulement si, $|q| < 1$. *Dans ce cas :*

$$S = \lim_{n \to +\infty} S_n = \frac{1}{1-q}$$

Sommaire

Sommaire

Définition 8 (Dérivabilité)

Soit f une fonction numérique définie sur un intervalle I de $\mathbb{R}$.
Soit $a \in I$. On dit que f est dérivable en a si, et seulement si, le rapport :

$$\tau_a(h) = \frac{f(a+h) - f(a)}{h}$$

admet une limite finie lorsque a tend vers 0.
Dans ce cas on note $f'(a)$ cette limite.

Théorème 9

Si f est dérivable en a avec $f'(a) \neq 0$, alors

$$f(x) - f(a) \underset{a}{\sim} f'(a) \times (x - a)$$

Définition 10 (Dérivée $k^{\text{Ème}}$)

Si f' est dérivable sur I on note f'' ou $f^{(2)}$ la dérivée de f' et on l'appelle la dérivée seconde de f. De même si f'' est dérivable sur I, f''' ou $f^{(3)}$, la dérivée de f'', est la dérivée troisième de f. Plus généralement on note $f^{(k)}$ la dérivée $k^{i\text{Ème}}$ de f avec la convention $f^{(0)} = f$, $f^{(k+1)} = \left(f^{(k)}\right)'$.

Définition 11 (Fonctions de classe $\mathcal{C}^n$, de classe $\mathcal{C}^\infty$)

On dit que f est de classe $\mathcal{C}^n$, $n \in \mathbb{N}$, sur I ssi f est n fois dérivable et ses n dérivées sont continues sur I. On remarquera que si $f^{(n+1)}$ existe sur I alors f est de classe $\mathcal{C}^n$ sur I qui s'écrit $f \in \mathcal{C}^n(I)$. On dit que f est de classe $\mathcal{C}^\infty$ sur I pour exprimer que f est indéfiniment dérivable et, a fortiori, que toutes ses dérivées sont continues.

```
def diff(f,h):
    def fp(x):
        return (f(x+h) - f(x))/h
    return fp
```

```
>>> def g(x):
    return x**2

>>> diff(g,1e-10)(1)
2.000000165480742
```

```
def diff(f,h):
    def fp(x):
        return (f(x+h) - f(x))/h
    return fp
```

```
>>> def g(x):
    return x**2

>>> diff(g,1e-10)(1)
2.000000165480742
```

```python
def diffn(f,h,n):
    if n == 0:
        return f
    else:
        return diff(diffn(f,h,n-1),h)
```

```
>>> from math import exp
>>> diff(exp,1e-5)(0)
1.000005000006965
>>> diffn(exp,1e-5,2)(0)
1.000011849706173
>>> diffn(exp,1e-5,3)(0)
0.8881784197001251
>>> diffn(exp,1e-5,4)(0)
22204.46049250314
>>> diffn(exp,1e-5,5)(0)
-6661338147.75094
```

Sommaire

Sommaire

Théorème 12

Si f est de classe $\mathcal{C}^n$ sur $[a,b]$ et si $f^{(n+1)}$ existe sur $]a,b[$ alors il existe au moins un réel $c \in]a,b[$ tel que :

$$f(b) = \sum_{k=0}^{n} \frac{(b-a)^k}{k!} f^{(k)}(a) + \frac{(b-a)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

On dit que l'on a écrit la formule de Taylor Lagrange à l'ordre n.

$$f(a+h) = \sum_{k=0}^{n} \frac{h^k}{k!} f^{(k)}(a) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(a+\theta h)$$

Théorème 12

Si f est de classe $\mathcal{C}^n$ sur $[a,b]$ et si $f^{(n+1)}$ existe sur $]a,b[$ alors il existe au moins un réel $c \in]a,b[$ tel que :

$$f(b) = \sum_{k=0}^{n} \frac{(b-a)^k}{k!} f^{(k)}(a) + \frac{(b-a)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

On dit que l'on a écrit la formule de Taylor Lagrange à l'ordre n.

$$f(a+h) = \sum_{k=0}^{n} \frac{h^k}{k!} f^{(k)}(a) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(a+\theta h)$$

Théorème 12

Si f est de classe $\mathcal{C}^n$ sur $[a,b]$ et si $f^{(n+1)}$ existe sur $]a,b[$ alors il existe au moins un réel $c \in]a,b[$ tel que :

$$f(b) = \sum_{k=0}^{n} \frac{(b-a)^k}{k!} f^{(k)}(a) + \frac{(b-a)^{n+1}}{(n+1)!} f^{(n+1)}(c)$$

On dit que l'on a écrit la formule de Taylor Lagrange à l'ordre n.

$$f(a+h) = \sum_{k=0}^{n} \frac{h^k}{k!} f^{(k)}(a) + \frac{h^{n+1}}{(n+1)!} f^{(n+1)}(a+\theta h)$$

Sommaire

Théorème 13

$$f(x) = \sum_{k=0}^{n} \frac{x^k}{k!} f^{(k)}(0) + \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\theta x)$$

Sommaire

Théorème 14

$$\forall x \in I, f(x) = \sum_{k=0}^{n} \frac{(x-a)^k}{k!} f^{(k)}(a) + \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)dt$$

Sommaire

Théorème 15

$$\forall x \in I, f(x) = \sum_{k=0}^{n} \frac{(x-a)^k}{k!} f^{(k)}(a) + o\left((x-a)^n\right)$$

Sommaire

Sommaire

Sommaire

Définition 16

On dit que f, définie sur $\mathcal{V}_a$, admet un développement limité d'ordre $n \in \mathbb{N}$ au voisinage de a (on écrit en abrégé : « f admet un $DL_n V(a)$ ») s'il existe un polynôme $P_n \in \mathbb{R}_n[X]$ (ensemble des polynômes de degré inférieur ou égal à n) et une fonction ε vérifiant :

$$\begin{cases} \forall x \in \mathcal{V}_a, f(x) = P_n(x-a) + (x-a)^n \varepsilon(x) \\ \varepsilon(a) = 0 \\ \lim\limits_{x \to a} \varepsilon(x) = 0 \end{cases}$$

Sommaire

Fonction `DL(f(x),n:` *une expression, un entier* `)` : *une expression polynomiale*

Début

- `Der` $\leftarrow$ `f (sous forme de fonction !)`
- `Pol` $\leftarrow$ `0`
- **Pour** `k` **variantDe** `0` **à** `n` **Faire**
  - `Pol` $\leftarrow$ `Pol+Der(0)` $\times \frac{x^k}{k!}$
  - `Der` $\leftarrow$ `fonction dérivée de Der`
- **FinPour**
- **Retourner** `Pol`

Fin

```
def DL(F,n):
    var('x')
    f(x) = F
    Der = f
    Pol = 0
    for k in range(n+1):
        Pol += Der(0)*x**k/factorial(k)
        Der = Der.derivative()
    return Pol
```

```
sage: DL(exp(x),5)
1/120*x^5 + 1/24*x^4 + 1/6*x^3 + 1/2*x^2 + x + 1
sage: DL(log(1+x),5)
1/5*x^5 - 1/4*x^4 + 1/3*x^3 - 1/2*x^2 + x
```

```
sage: taylor(log(1+x),x,0,5)
1/5*x^5 - 1/4*x^4 + 1/3*x^3 - 1/2*x^2 + x
```

```
sage: DL(exp(x),5)
1/120*x^5 + 1/24*x^4 + 1/6*x^3 + 1/2*x^2 + x + 1
sage: DL(log(1+x),5)
1/5*x^5 - 1/4*x^4 + 1/3*x^3 - 1/2*x^2 + x
```

```
sage: taylor(log(1+x),x,0,5)
1/5*x^5 - 1/4*x^4 + 1/3*x^3 - 1/2*x^2 + x
```

```
def anim_DL(f,p,xm,xM,ym,yM,d):
    P = plot(f,rgbcolor='red',xmin=xm,xmax=xM,ymin=ym,ymax=yM,
        thickness=3, linestyle='--')
    a = animate([[P,DL(f,k)] for k in range(1,p)],xmin=xm,xmax=xM,
        ymin=ym,ymax=yM)
    a.show(delay=d) # délai en 100e de seconde entre deux images.
```

```
sage: anim_DL(log(1+x),20,-1,2,-2,2,50)
sage: anim_DL(cos(x),30,-4*pi,4*pi,-1.1,1.1,50)
```

```
def anim_DL(f,p,xm,xM,ym,yM,d):
    P = plot(f,rgbcolor='red',xmin=xm,xmax=xM,ymin=ym,ymax=yM,
        thickness=3, linestyle='--')
    a = animate([[P,DL(f,k)] for k in range(1,p)],xmin=xm,xmax=xM,
        ymin=ym,ymax=yM)
    a.show(delay=d) # délai en 100e de seconde entre deux images.
```

```
sage: anim_DL(log(1+x),20,-1,2,-2,2,50)
sage: anim_DL(cos(x),30,-4*pi,4*pi,-1.1,1.1,50)
```