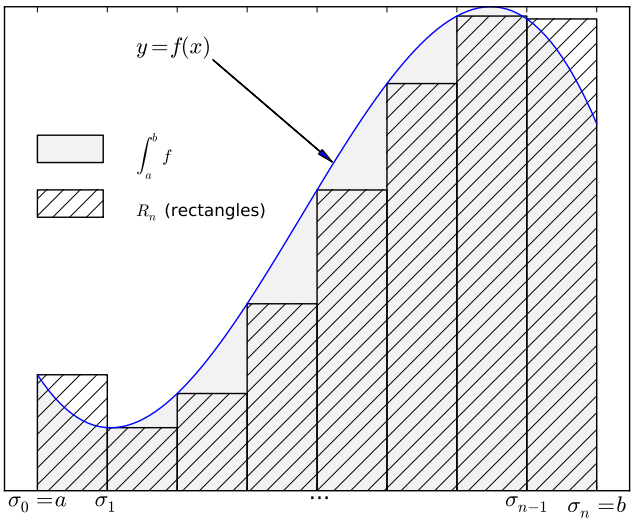
III) Calculs d'intégrales

A) Subdivisions

$$x_k = a + \frac{k(b-a)}{n}$$

`x[k] = a+k*pas` avec `pas = (b-a)/n`

B) Méthode des rectangles à gauche



- Formule :

$$I_n = \frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a+\frac{k(b-a)}{n}\right)$$

- Code

```
def methode_rect(f, a, b, n):
    pas=(b-a)/float(n)
    return pas*sum([f(a+k*pas) for k in range(n)])
```

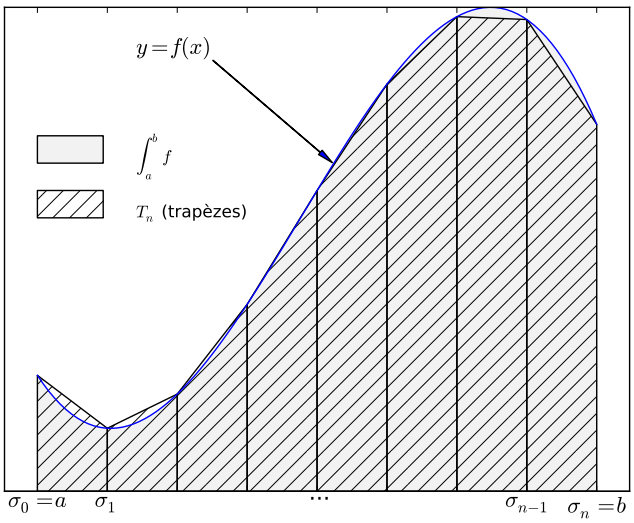
- Erreur :

$$|I - I_n^{\text{rect}}| \leqslant \frac{(b-a)^2}{2n}\sup_{[a,b]}|f'|$$

- Méthode d'ordre 0 (exacte au degré 0 : $f(x)=a_0$).

C) Trapèzes

1) Naïfs



- Formule :

$$I_n = \frac{b-a}{n}\left(\frac{f(a)+f(b)}{2} + \sum_{k=1}^{n-1} f\left(a+\frac{k(b-a)}{n}\right)\right)$$

- Erreur :

$$|I - I_n^{\text{trap}}| \leqslant \frac{(b-a)^3}{12n^2}\sup_{[a,b]}|f''|$$

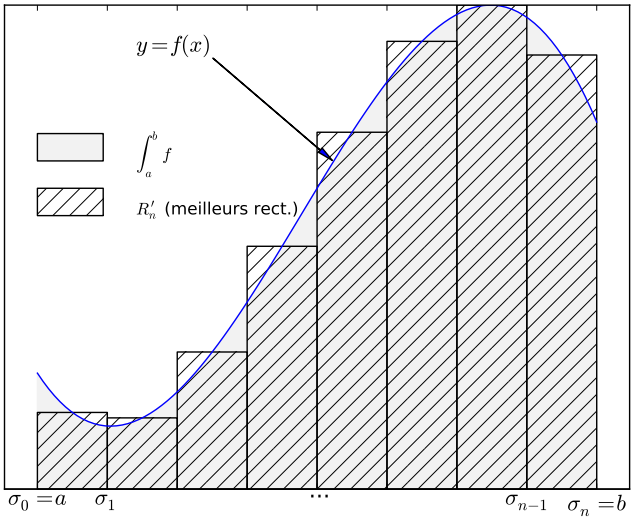
- Méthode d'ordre 1 (exacte au degré 1 : $f(x)=a_1x+a_0$).

- Code

```
def methode_trap(f, a, b, n):
    pas=(b-a)/float(n)
    return pas*((f(a)+f(b))/float(2)+sum([f(a+k*pas) for k in range(1,n)]))
```

2) Raffinement : point milieu

Correspond à un trapèze tangent en $\dfrac{x_k + x_{k+1}}{2}$.



- Formule :

$$I_n = \frac{b-a}{n} \sum_{k=1}^{n-1} f\left(a + \frac{b-a}{2n} + \frac{k(b-a)}{n}\right)$$

- Erreur :

$$|I - I_n^{\text{trap}}| \leqslant \frac{(b-a)^3}{24n^2} \sup_{[a,b]} |f''|$$

- Méthode d'ordre 1 (exacte au degré 1 : $f(x) = a_1 x + a_0$).

- Code

```
def methode_pt_milieu(f, a, b, n):
    pas=(b-a)/float(n)
    return pas*sum([f(a+pas/2+k*pas) for k in range(n)])
```

Code Python des dessins précédents

```
# -*- coding: utf-8 -*-
"""
Created on Thu Aug 15 09:59:09 2013

@author: stephane
"""

import matplotlib.pyplot as pypl
import numpy as np


arp = dict(width=0.2,headwidth=4,shrink=0.01) # arrows properties
a, b = -0.8, 0.9

def f(t):
    return 0.5+t-t*t*t

x = np.linspace(a, b, 100)
y = f(x)

def un_rectangle(g,d,y):
    pypl.fill([g,d,d,g] , [0,0,y,y], fill = False, hatch = '/')

def un_trapeze(g,d):
    pypl.fill([g,d,d,g] , [0,0,f(d),f(g)], fill = False, hatch = '/')

n = 8 # si on change, attention au xticks

gris = '0.95'

sigma = np.linspace(a,b,n+1)

def decorer():
    pypl.plot(x,y)
    pypl.fill([b,a]+list(x),[0,0]+list(y), color=gris)
    pypl.axes().set_xlim(a-.1, b+.1)
    pypl.xticks(sigma,['$\sigma_0=a$','$\sigma_1$','','','...','','',
                    '$\sigma_{n-1}$','$\sigma_n=b$'], fontsize=16)
    pypl.yticks([])
    pypl.annotate('$y=f(x)$', xy=(0.1, f(0.1)), xytext=(-0.5,0.8), arrowprops=arp,
                  fontsize=16)
    pypl.axhline(color='black')
    pypl.fill([-0.8,-0.6,-0.6,-0.8] , [0.65,0.65,0.6,0.6], gris)
    pypl.fill([-0.8,-0.6,-0.6,-0.8] , [0.5,0.5,0.55,0.55], fill=False, hatch = '/')
    pypl.text(-0.5,0.6,'$\int_a^b f$')


#
# Rectangles
#

decorer()
```

```
pypl.text(-0.5,0.5,'$R_n$ (rectangles)')
for i in range(n):
    un_rectangle(sigma[i],sigma[i+1],f(sigma[i]))
pypl.savefig('rectangles-gauches.pdf')
pypl.show()

#
# Trapèzes
#

pypl.clf()
decorer()
pypl.text(-0.5,0.5,u'$T_n$ (trapèzes)')
for i in range(n):
    un_trapeze(sigma[i],sigma[i+1])
pypl.savefig('trapezes.pdf')
pypl.show()

#
# Point milieu
#

decorer()
pypl.text(-0.5,0.5,'$R_n\'$ (meilleurs rect.)')
for i in range(n):
    un_rectangle(sigma[i],sigma[i+1],f((sigma[i]+sigma[i+1])/2.0))
pypl.savefig('rectangles-milieu.pdf')
pypl.show()
```