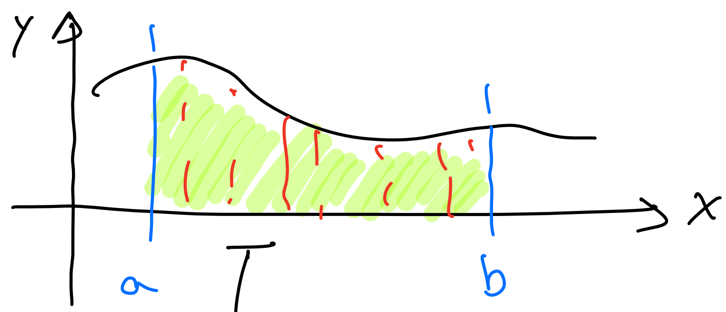
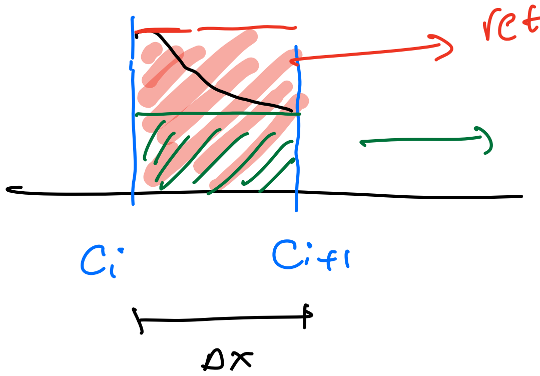
Metodi di Integrazione



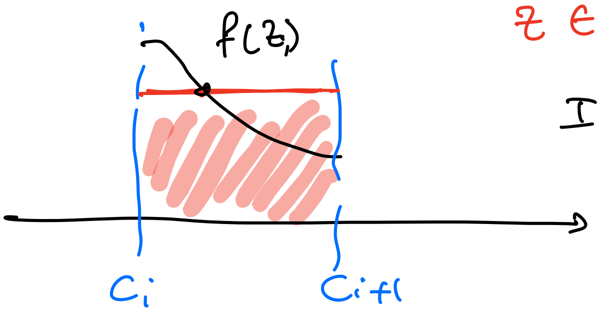
- metodo dei rettangoli
- metodo del trapezio

$$\frac{b-a}{N} = \Delta x = c_{i+1} - c_i$$



rettangoli con il massimo in $[c_i, c_{i+1}]$

rettangoli con il minimo nell'intervallo $[c_i, c_{i+1}]$



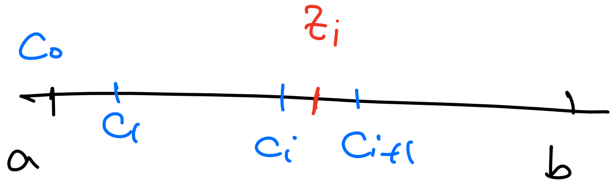
$z \in [c_i, c_{i+1}]$

$$I_i = \text{Area rettangolo: } \Delta x \cdot f(z) = \frac{b-a}{N} \cdot f(z_i)$$

$$I = \int_a^b f(x)\,dx = \sum_i I_i = \sum_i \frac{b-a}{N} f(z_i) = \frac{b-a}{N} \sum_i f(z_i)$$

Metodo del punto di mezzo:

$$z_i = \frac{c_i + c_{i+1}}{2}$$



$c_i =$

$c_0 = a$

$c_1 = a + \Delta x$

b − a diviso in N intervalli

$c_i = a + i\,\Delta x$

$c_{i+1} = c_i + \Delta x = a + i\,\Delta x + \Delta x = a + (i+1)\Delta x$

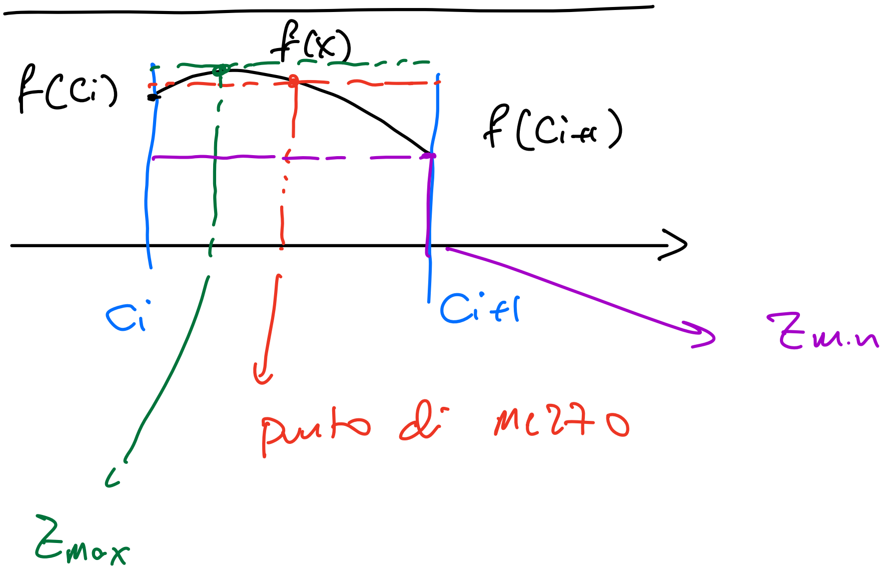
$$z_i = \frac{c_i + c_{i+1}}{2} = \frac{a + i\,\Delta x + a + (i+1)\Delta x}{2} = a + i\Delta x + 0.5\Delta x$$

$= c_i + 0.5\,\Delta x$

$$I = \underbrace{\frac{b-a}{N}}_{\Delta x} \sum_i f(z_i) = \frac{b-a}{N} \sum_{1}^{N} {}_i\, f(a + i\,\Delta x + 0.5\Delta x)$$

```
double dx = (b-a)/N;
double integ = 0;
for(int i=0; i<N; i++){
        integ += f(a+i*dx+0.5*dx)
}
integ *= (b-a)/N;
```

Metodo del trapezio



Polinomi di Legendre: $L_{n-1} = \sum_{1}^{n} {}_i\, f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$

Conosciamo $f(x_i)$ in $\{x_1, \dots, x_n\}$

$n = 1 \Rightarrow L_0 = \sum_{i=1}^{1} f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$

x_1, x_2

$$= f(x_1) \prod_{\substack{j=1 \\ j \neq 1}}^{1} \frac{x - x_j}{x_1 - x_j}$$

$$= f(x_1)$$

$n = 2 \qquad L_1 = \sum_{i=1}^{2} f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$

$$= f(x_1) \prod_{\substack{j=1 \\ j \neq 1}}^{2} \frac{x - x_j}{x_1 - x_j} + f(x_2) \prod_{\substack{j=1 \\ j \neq 2}}^{2} \frac{x - x_j}{x_2 - x_j}$$

$$= f(x_1) \frac{x - x_2}{x_1 - x_2} + f(x_2) \frac{x - x_1}{x_2 - x_1}$$

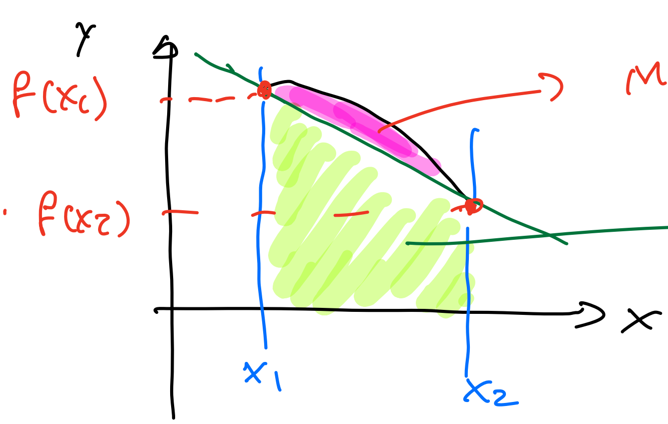
$$= f(x_1) \frac{x - x_2}{x_1 - x_2} - f(x_2) \frac{x - x_1}{x_1 - x_2}$$

retta passante per x_1, x_2

$$L_1(x = x_1) = f(x_1) \frac{\cancel{x_1 - x_2}}{\cancel{x_1 - x_2}} - f(x_2) \frac{\cancel{x_1 - x_1}}{x_1 - x_2}$$

$$= f(x_1)$$

$$L_1(x = x_2) = f(x_2)$$

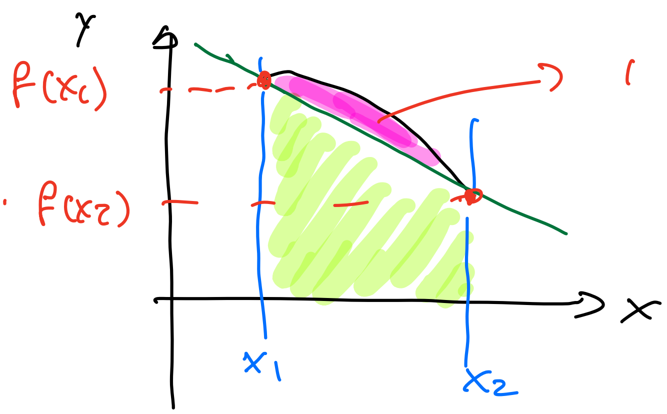


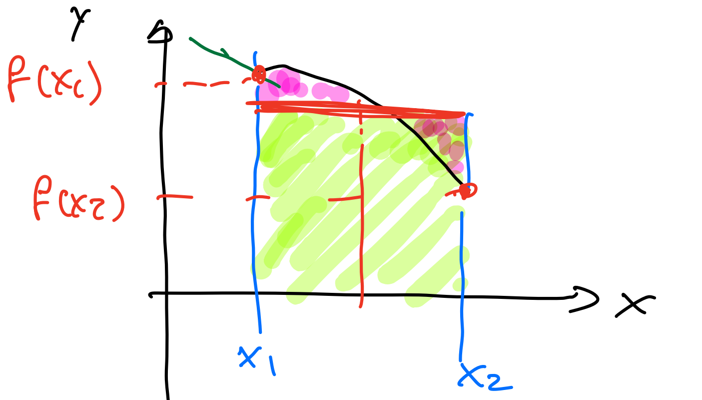
integrale sottostimato

area del trapezio

$$I_i = \frac{(f(x_2) + f(x_1)) \cdot \Delta x}{2}$$

Metodo del trapezio





$$\Delta_n(x) = |f(x) - L_{n-1}(x)| \le \sup_{z\in[a,b]} |f^{(n)}(z)| \frac{\prod_{j=1}^{n}(x-x_j)}{n!}$$

$$I_i = \int_{c_i}^{c_{i+1}} f(x)\,dx \qquad I_i^{(n)} = \int_{c_i}^{c_{i+1}} L_{n-1}(x)\,dx$$

$$\delta_i^{(n)} =: |I_i - I_i^{(n)}| \le \int_{c_i}^{c_{i+1}} |\Delta_n|\,dx =$$

Fattore numerico

$$= \sum_i \delta_i^{(n)} = A_n(f)\,\frac{D_n}{2^{n+1}}\,\frac{(b-a)^{n+1}}{M^n}$$

$$A_n(f) = \left|\sup_{z\in[a,b]} f^{(n)}(z)\right|$$

M: numero di intervalli: $\Delta x = \frac{b-a}{M}$

```
for(int i=0; i<N; i++) {
    for(int j=0; j<N && j!=i; j++) {
        if(j==i) continue;
    }
}
```

double integrale1 =

$$I = \int_a^b f(x)\,dx$$

funzione integrale:

tipo: double

argomenti: a, b, n, funzione

double integral(double a, double b, int n,

```
x = integrale( 0, M_PI, 100, sin );
x = integral( 0, 2.3, 1000, func );
```

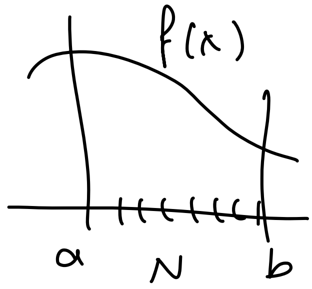
puntatore a funzione:

```
double myf(double x) {
    return  x * sin(x);
}

int main() {


    double tot = integral( 0, 2.5, 1000, myf );

    double (* pf)(double);
```

pf: puntatore a funzione di tipo double, con 1 solo arg. di tipo double pf = &myf;

pf = &sin;

pf = &exp;



funzione → estr. int → estremo sup. → N punti

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

double myf(double x) {
    return x*sin(x)*log(x);
}


double midPoint( double (*)(double), double, double, int);
double trapezio( double (*)(double), double, double, int);

int main() {

    double (*pf)(double);
    pf = &myf;
    //pf = &cos;
    //pf = &sinh;

    double a = 0.001, b = 2.5;

    for(int npt=10; npt<1e8; npt*=10) {
        double mpInt = midPoint( pf, a, b, npt);
        double trapInt = trapezio( pf, a, b, npt);
        printf("npt: %8d \t midpoint: %.6f \t trapezio: %.6f\n", npt, mpInt, trapInt);
    }

}

double midPoint( double (*f)(double), double a, double b, int npt)  {

    double dx = (b-a)/npt;

    double tot = 0.;
    for(int i=0; i<npt; i++) {
        double c1 = a + i*dx;
        double c2 = a + (i+1)*dx;
        double m = a + i*dx + 0.5*dx;
        tot += f(m)*dx;
    }
    return tot;

}


double trapezio( double (*f)(double), double a, double b, int npt)  {

    double dx = (b-a)/npt;

    double tot = 0.;
    for(int i=0; i<npt; i++) {
        double c1 = a + i*dx;
        double c2 = a + (i+1)*dx;
        tot += 0.5*(f(c1)+f(c2))*dx;
    }
    return tot;

}
```

```
Mac:material rahatlou$ gcc -o /tmp/app integrals.c
Mac:material rahatlou$ /tmp/app
npt:       10    midpoint: 1.204937      trapezio: 1.200466
npt:      100    midpoint: 1.203545      trapezio: 1.203493
npt:     1000    midpoint: 1.203528      trapezio: 1.203527
npt:    10000    midpoint: 1.203528      trapezio: 1.203528
npt:   100000    midpoint: 1.203528      trapezio: 1.203528
npt:  1000000    midpoint: 1.203528      trapezio: 1.203528
npt: 10000000    midpoint: 1.203528      trapezio: 1.203528
```