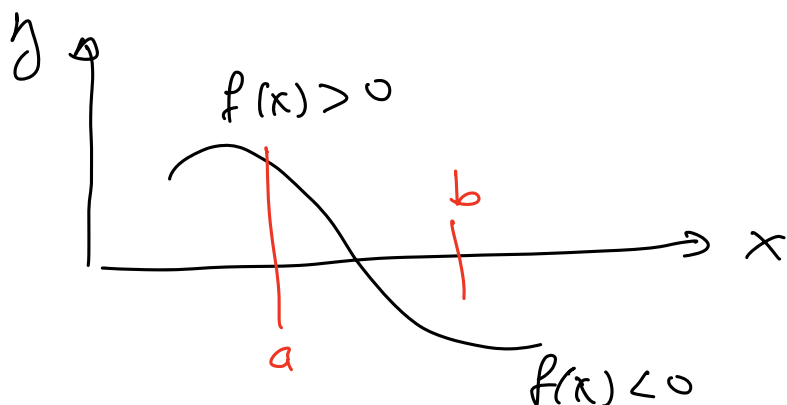


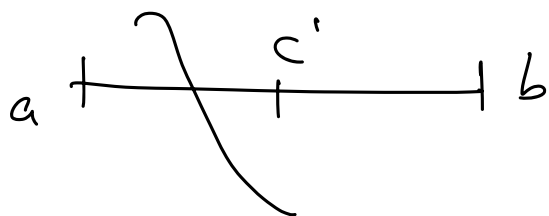
Metodo di bisezione



$$f(a) * f(b) < 0$$

$$x_i: f(x_i) = 0.$$

scelgo $[a, b]$ che contiene x_i :



$$f(a) * f(b) < 0.$$

$$c = \frac{a+b}{2}$$

$$f(a) * f(c) \leq 0$$

$$f(b) * f(c) > 0 \quad x_i \notin [c, b]$$

ε : precisione, risoluzione, tolleranza

$$|x_i - x_{vera}| < \varepsilon$$

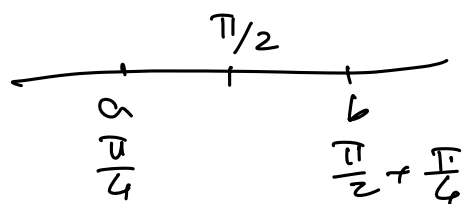
$$x_i: C_n$$

Condizione effettiva

$$|a-b| < \varepsilon$$

$$f(x) = \cos(x)$$

$$x_0 = \frac{\pi}{2}$$



$$a = \frac{\pi}{4}$$

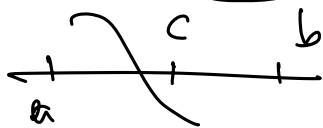
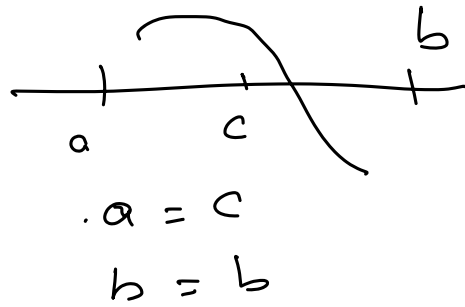
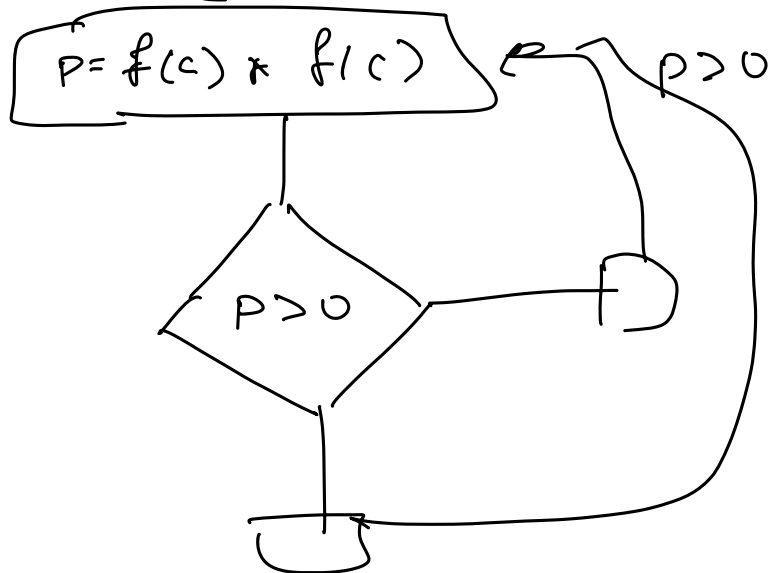
$$b = \frac{\pi}{4} + \frac{\pi}{2}$$

$$\varepsilon = 0.1, 0.001, 10^{-8}$$

determina numero di iteraz.

$$c = \frac{a+b}{2}$$

punto di mezzo.



$$P = f(a) * f(c) < 0$$

#include <math.h> a = a
b = c

double epsilon;
printf("precision della soluzione: ");
scanf("%lf", &epsilon);

// input a
// input b
int n = 0;
double c = -99999;
double delta = fabs(a - b);

~~scanf("%lf", &x);~~
~~scanf("%lf", &x);~~
1 a - b)

while (delta > epsilon) {

~~double c;~~
c = (a + b) / 2;
n++;

double P = f(a) * f(c);

f(x) = cos(x)

if (P > 0) {

a = c;

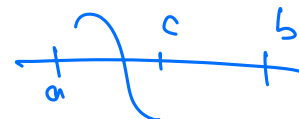
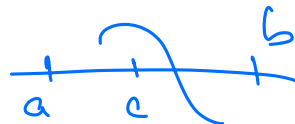
} else if (P < 0) {

b = c;

} else {

a = c;
b = c;

}



$$a = b = c;$$

```

    delta = fabs(a-b);
} // while

```

```

printf("soluzione: %.3g\n", c);
printf("c - pi/2 = %.1f\n", c - M_PI/2.);

```

} undefined symbol c

variabili dichiarate dentro una coppia { }
non disponibili fuori.

scope delle variabili:

```

int main() {
    _____
    _____
    _____
    _____

```

```

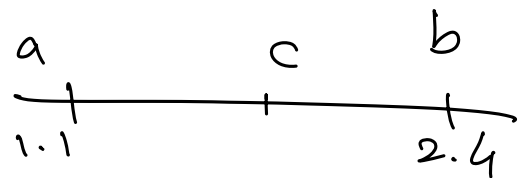
}

```

epsibn = 0.1

a = $\pi/3$ 1.11 1.1

b = $\pi/2 + \pi/4$ 2.4



```

#include <math.h>

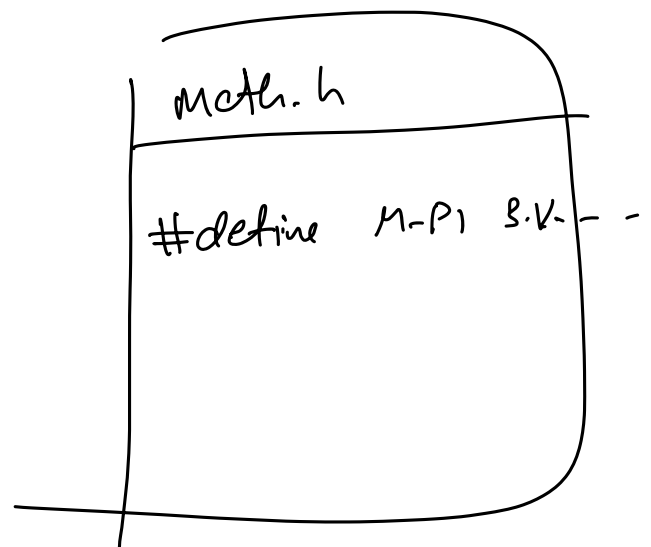
```

```

scanf("%.1f", &x);

```

M-PI/2



Errori Comuni:

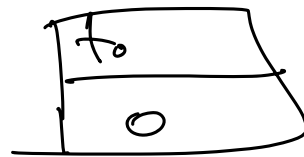
^{assegnazione}
`if (t = 0) {`

`}`

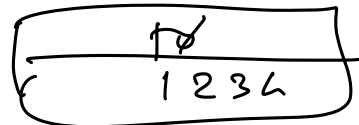
`if (espressione logica)`

~~t~~ = 0
^{confronto}
`if (t == 0) {`

Sempre vera



vera



False.

`if ("t = 0") {`

Sempre vera

`if ("pippo") {`

Sempre vera

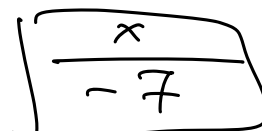
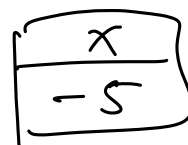
`double x;`
`while (x) {`

Sempre vera

se $x \neq 0.00000$

`}`

`printf("x: ");`
`scanf("%i", &x);`
`if (x < 0) { vera`
 `scanf("%i", &x);`
`}`



```

if (x < 0) {
    printf("in sense: x > 0: ");
    scanf("%i", &x);
}

```

x
-9

```

do {
    // sense:
    printf("x > 0: ");
    scanf("%i", &x);
} while (x < 0);

```

x
-5

-7

$$G(t) = \exp(-0.5 \times \dots)$$

$$(-1/2) + (\sin(x)) \quad -1/2$$

$$-0.5 \times \sin(x)$$

$$(-1/2)$$

$$(t - t_0)^{1/2}$$

$$-\frac{x}{y} \quad \frac{-x}{y} \quad \frac{x}{-y}$$

NON C'E' in C!

$$1/2$$

Seppure Zero

double $G(t)$; (simbolo speciale

$$G = \exp(-0.5 \times (t - t_0) \times (t - t_0) / \text{signe} / \text{signe})$$

$$\frac{(t - t_0)^2}{\sigma^2} \equiv \left(\frac{(t - t_0)}{\sigma} \right)^2 \frac{1}{\sigma}$$

$\text{pow}(f-t^0, 2) / \text{pow}(\text{Sign}, 2)$

C: gware.c
~~gware.99~~

Python

No Compilatore
solo interprete

```
eSempio.py  
-----  
print("Ciao!")  
a = -3.5  
print(a)
```

shell > python3

eSempio.py