

LABORATORIO DI MATEMATICA

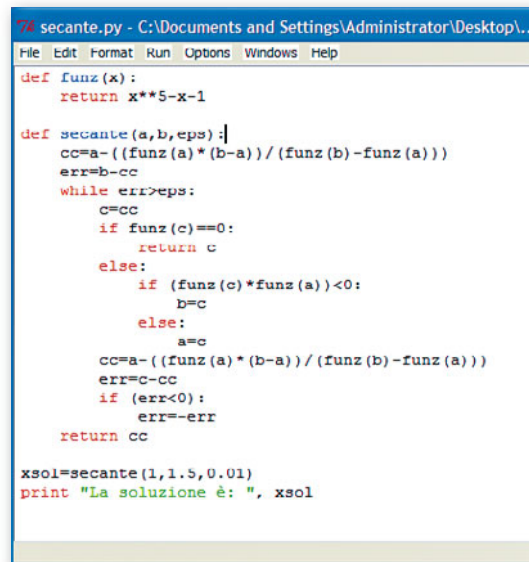
Il metodo delle secanti con Python

Traduciamo in una funzione l'algoritmo del metodo delle secanti; applichiamo la funzione per determinare la radice di $x^5 - x - 1 = 0$ nell'intervallo $\left[1; \frac{3}{2}\right]$, con un'approssimazione di 0,01.

RISOLUZIONE

Usiamo il linguaggio di programmazione Python.

- Definiamo preliminarmente la funzione `funz` che per ogni numero reale x in ingresso restituisce $f(x) = x^5 - x - 1$.
- La funzione `secante` riceve in ingresso gli estremi dell'intervallo e l'approssimazione richiesta, utilizza la funzione `funz` definita precedentemente e restituisce il valore della radice cercata.
- Determiniamo l'intersezione c tra la secante che passa per gli estremi dell'intervallo e l'asse x .
- Valutiamo la funzione nel punto d'intersezione.
- Se $f(c) = 0$, allora abbiamo trovato la radice altrimenti ridefiniamo l'intervallo di ricerca scegliendolo opportunamente tra $[a; c]$ e $[c; b]$ e ripetiamo i passi precedenti.
- Usciamo dal ciclo quando troviamo la radice esatta oppure quando la distanza (`err`) tra due valori approssimati consecutivi è minore dell'approssimazione richiesta.
- Utilizziamo la funzione `secante` inserendo i dati del problema con l'istruzione `secante(1, 1.5, 0.001)`.
- Dopo aver salvato il file, dalla barra dei menu in alto, clicchiamo su *Run* e poi su *Run Module* per avviare l'esecuzione del programma. Sulla Python Shell comparirà il messaggio a fianco.



```

secante.py - C:\Documents and Settings\Administrator\Desktop\...
File Edit Format Run Options Windows Help

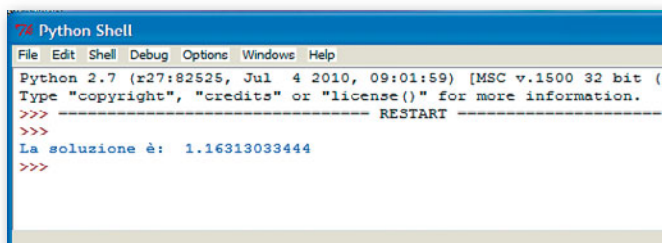
def funz(x):
    return x**5-x-1

def secante(a,b,eps):
    cc=a-((funz(a)*(b-a))/(funz(b)-funz(a)))
    err=b-cc
    while err>eps:
        c=cc
        if funz(c)==0:
            return c
        else:
            if (funz(c)*funz(a))<0:
                b=c
            else:
                a=c
        cc=a-((funz(a)*(b-a))/(funz(b)-funz(a)))
        err=c-cc
        if (err<0):
            err=-err
    return cc

xsol=secante(1,1.5,0.01)
print "La soluzione è: ", xsol

```

figura 1



```

Python Shell
File Edit Shell Debug Options Windows Help

Python 2.7 (r27:82525, Jul 4 2010, 09:01:59) [MSC v.1500 32 bit (I
Type "copyright", "credits" or "license()" for more information.
>>> ----- RESTART -----
>>>
La soluzione è: 1.16313033444
>>>

```

figura 2

ESERCIZI IN PIÙ

- 1 Scrivi, utilizzando il linguaggio di progetto, l'algoritmo di ricerca delle radici mediante il metodo delle secanti; modifica il listato dell'esercitazione guidata in modo che il programma restituisca il numero di iterazioni compiute per fornire il risultato.
- 2 Utilizzando il linguaggio di progetto, scrivi l'algoritmo di ricerca delle radici mediante il metodo di bisezione e traducilo nel linguaggio di programmazione Python.

Prova i programmi degli esercizi precedenti nella ricerca degli zeri delle seguenti funzioni e confronta, a parità di approssimazione richiesta, il numero di iterazioni compiute dai due algoritmi.

3 $y = 3x^4 - 5x + 1$

4 $y = 2x^3 - 3x^2 + x - \frac{1}{2}$

5 $y = x^6 - 2x^5 - 1$