

Cognome	Nome	N. matricola	Postazione

CALCOLO NUMERICO E PROGRAMMAZIONE  
 Ingegneria per l'Ambiente ed il Territorio, Prof. Mario Putti  
 Compitino: 28 aprile 2016

Si vuole trovare lo zero della funzione  $f(x) = 7 - 5.5x - 6x^2 + 5.5x^3 - x^4$  nell'intervallo  $[-1.5, 4]$  usando il metodo di Newton-Raphson.

Si richiede di:

- costruire un programma in fortran90 (chiamato `zeri.f90`) che implementa il metodo di Newton-Raphson per risolvere l'equazione  $f(x) = 0$ . Il programma dovrà seguire le indicazioni riportate nel template `template_zeri.f90` e in particolare, dopo aver dichiarato tutte le variabili, si dovrà chiamare la subroutine `newrap` che implementa le iterazioni del metodo. Nella subroutine verranno usate le functions `real*8 function fun(...)` e `real*8 function der(...)` che implementano rispettivamente la  $f(x)$  e la sua derivata;
- all'interno del ciclo stampare nel file `conv.ris` una tabella di 5 colonne con i valori di:  
`iter, xkp1, scarto, asint1, asint2`,  
 dove le variabili `asint1` e `asint2` contengono rispettivamente il rapporto tra lo scarto all'iterazione corrente e quello all'iterazione precedente e il rapporto tra lo scarto all'iterazione corrente e il quadrato di quello all'iterazione precedente;
- dopo aver completato il programma, compilarlo con l'istruzione: `gfortran zeri.f90` che crea il file eseguibile `a.out`. Correggere gli eventuali errori;
- far girare il programma per i seguenti casi test:
  - TC1:  $x_0 = 0.00$ ,  $toll = 10^{-7}$ ,  $ITMAX = 100$ .
  - TC2:  $x_0 = -0.25$ ,  $toll = 10^{-7}$ ,  $ITMAX = 100$ .

controllare i risultati nei file di output `newrap.ris` e `conv.ris`. Salvare i file di output con i nomi `newrap1.ris` e `conv1.ris` per TC1 e `newrap2.ris` e `conv2.ris` per TC2.

- dalle tabelle contenute in `conv1.ris` e `conv2.ris` verificare gli ordini di convergenza ( $p_1$  per TC1 e  $p_2$  per TC2) dello schema e le costanti asintotiche di convergenza ( $M_1$  per TC1 e  $M_2$  per TC2). Riportare i valori nello spazio sottostante:

$p_1$ (TC1)	$M_1$ (TC1)	$p_2$ (TC2)	$M_2$ (TC2)
2	0.89985E+00	2	0.12908E+01

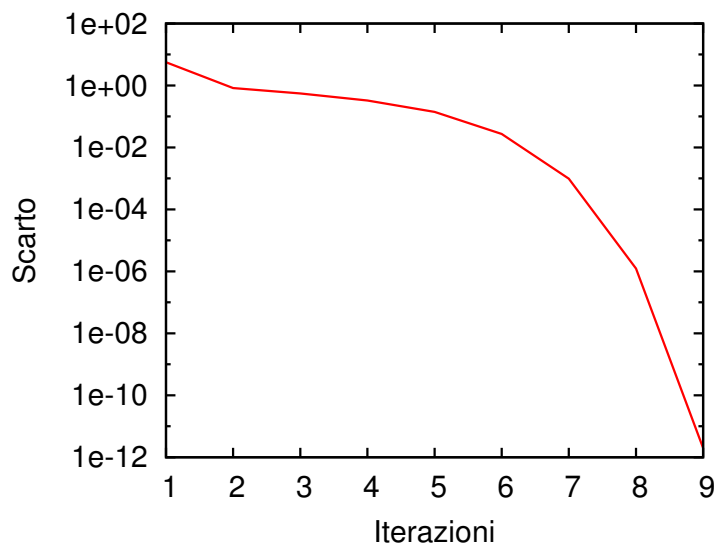
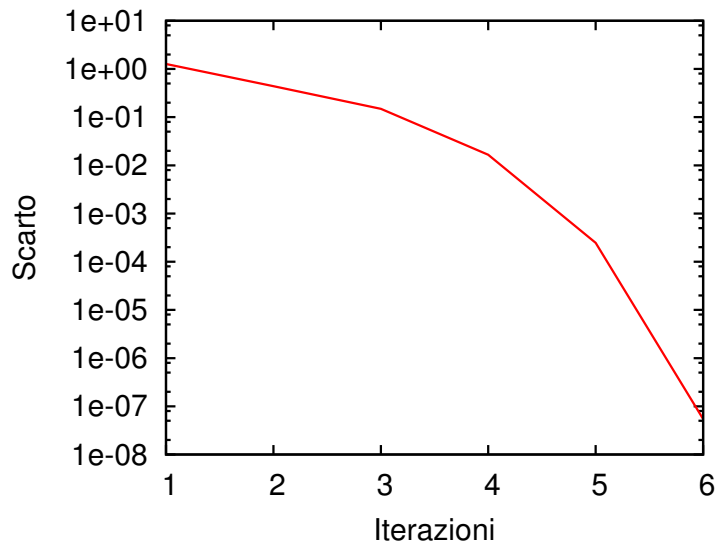
Giustificare le differenze che si notano nei risultati di TC1 e TC2.  
 Partendo da -0.25 si è vicini ad un punto in cui la derivata è piccola, per cui il metodo di Newton, alla prima iterazione, si allontana dal punto iniziale e converge alla soluzione più lontana.

- disegnare il grafici di convergenza nei due casi e salvarli nei file `conv1.jpg` e `conv2.jpg` in formato jpeg. I comandi principali di gnuplot per costruire e salvare l'immagine in formato jpeg sono:  

```
set terminal 'jpeg'
set output 'conv?.jpg'
set logscale y
plot 'conv?.ris' using 1:3
```

NB: lasciare tutti i file nella vostra directory home per la correzione.

Tempo a disposizione: 1 ora 45 min.



```

1 program zeri
2 !
3 ! AA 2015-2016
4 ! Ing. per l'Ambiente e il Territorio
5 ! Calcolo Numerico e Programmazione
6 ! primo compito
7 ! 28 aprile 2016
8 !
9 ! completare inserendo o completando le opportune istruzioni
10 ! al posto dei puntini (che vanno cancellati)
11 !
12 !
13 ! STUDENTE: Mario Putti <-inserire nome e cognome
14 ! n. matr.: 9999999 <-inserire il numero di matricola
15 !
16 !
17 !
18 ! dichiarazione di tutte le variabili
19 !
20 implicit none
21 integer :: itmax, iter
22 real*8 :: x0, x1, toll, scarto, sol
23
24 open(3, file='newrap.dat')
25 open(9, file='newrap.ris')
26
27 !
28 ! leggere dal file collegato all'unita 3 i dati di input:
29 ! x0 = soluzione iniziale (real*8)
30 ! toll = tolleranza (real*8)
31 ! itmax = numero massimo di iterazioni (integer)
32 !
33 read(3,*) x0, toll, itmax
34 write(9,*) ' stampa dei dati di lettura '
35 write(9,*) ' x0 =',x0
36 write(9,*) ' toll =',toll
37 write(9,*) ' itmax =',itmax
38 !
39 ! chiamare la subroutine che implementa il metodo della secante fissa
40 ! inizializzare le variabili opportune
41 !
42 ! inserire nella chiamata alla subroutine sfissa
43 ! le variabili di scambio opportune
44 !
45 call newrap(x0,toll,itmax,iter,sol,scarto)
46 !
47 ! stampare la soluzione finale, lo scarto finale raggiunto, e il numero
48 ! di iterazioni necessarie per arrivare alla convergenza richiesta
49 !
50 if(iter.le.itmax) then
51 write(9,*) ' soluzione raggiunta in ',iter,' iterazioni '
52 write(9,'(a,e13.5)') ' sol =',sol
53 write(9,'(a,e13.5)') ' scarto =',scarto
54 else
55 write(9,*) ' soluzione non raggiunta in ',iter,' iterazioni '
56 write(9,'(a,e13.5)') ' sol =',sol
57 write(9,'(a,e13.5)') ' scarto =',scarto
58 end if
59 !
60 ! chiudere i file di input/output
61 !
62 close(3)
63 close(9)
64 stop
65 end program zeri
66
67 subroutine newrap(x0,toll,itmax,iter,xkp1,scarto)

```

```

68  implicit none
69  integer :: itmax, iter
70  real*8  :: x0, xk, xkp1
71  real*8  :: toll, scarto, scarto_old
72  real*8  :: asint1, asint2
73  real*8  :: fun, der
74  !
75  ! aprire il file conv.dat dove stampare la tabella di convergenza:
76  ! iter, xkp1, scarto, scarto_old, asint1, asint2
77  !
78  open(10, file='conv.ris')
79  !
80  ! inizializzare le variabili opportune
81  !
82  xk=x0
83  scarto=2.0d0*toll
84  iter=0
85  scarto_old=scarto
86  !
87  ! implementare il ciclo while
88  !
89  do while (scarto.gt.toll .and. iter.le.itmax)
90     iter = iter + 1
91     xkp1 = xk - fun(xk)/der(xk)
92     scarto = abs(xkp1 - xk)
93     asint1 = scarto/scarto_old
94     asint2 = scarto/scarto_old**2
95     write(10, '(i5,5e13.5)') iter, xkp1, fun(xkp1), scarto, asint1, asint2
96     write(*,'(i5,5e13.5)') iter, xkp1, fun(xkp1), scarto, asint1, asint2
97     xk = xkp1
98     scarto_old = scarto
99  end do
100
101  !
102  ! chiudere il file della tabella di convergenza
103  !
104  close(10)
105  return
106 end subroutine newrap
107
108 real*8 function fun(x)
109  !
110  ! function che implementa la funzione di cui trovare lo zero
111  !
112  implicit none
113  real*8 :: x
114  fun = 7.0 - 5.5*x - 6.0*x**2 + 5.5*x**3 - x**4
115  return
116 end function fun
117
118 real*8 function der(x)
119  !
120  ! function che implementa la derivata della funzione
121  !
122  implicit none
123  real*8 :: x
124  der = -5.5 - 12.0*x + 16.5*x**2 - 4.0*x**3
125  return
126 end function der

```