



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Dipartimento di Matematica

Laboratorio di Calcolo Numerico

Laboratorio 4: Grafici di convergenza e functions

Damiano Pasetto

E-mail: pasetto@math.unipd.it

Dispense: http://dispense.dmsa.unipd.it/putti/calcolo_ambientale/index.html

24 Marzo 2014

Grafici di convergenza

Obiettivi del laboratorio

In riferimento all'equazione non lineare descritta nel Laboratorio 3 si vuole:

- Disegnare il grafico di convergenza dello scarto delle iterazioni di Picard.
- Implementare l'algoritmo di Picard implementando le funzioni g_1 , g_2 e g_3 utilizzando le *function*.

Esempio

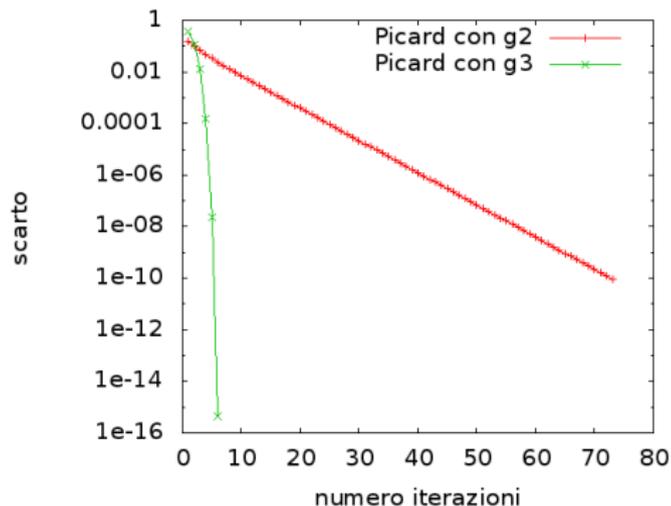


Grafico di convergenza per $g_2(x) = \sqrt{3x - 2}$ e $g_3(x) = (x^2 - 2)/(2x - 3)$.

Stampa di dati in un file di OUTPUT

- Comando fortran per l'apertura del file `nomefile`:

```
open(ounit, file='nomefile')
```

- Comando fortran per la scrittura delle variabili `var1`, `var2`, `var3`, ... dall'unità `ounit` (`ounit` è un numero intero):

```
write(ounit,*) var1, var2, var3,...
```

- Al posto dell'asterisco si può inserire un `formato`, per indicare che tipo di variabili si stanno stampando (intero, reale, ...) e quanti caratteri usare per la stampa. Per esempio il comando

```
write(ounit,'(i6,e15.7,f12.5)') var1, var2, var3
```

stampa: la variabile intera `var1` utilizzando 6 caratteri (`i6`), la variabile reale `var2` in formato esponenziale con 15 caratteri di cui 7 occupati dalla mantissa (`e15.7`), la variabile reale `var3` in formato a virgola fissa con 12 caratteri di cui 5 sono usati per le cifre decimali (`f12.5`).

- Comando fortran per la chiusura del file `nomefile`:

```
close(ounit)
```

Esempio: grafico di una funzione

Programma per valutare una funzione $y = f(x)$ in $n + 1$ punti equispaziati nell'intervallo $I = [x_{min}, x_{max}]$. n , x_{min} e x_{max} sono letti da terminale. I valori x_i , y_i sono stampati sul file *risult.dat*.

```
1  _____program outputfile
2  _____implicit none
3  _____(dichiarazione variabili)
4
5  _____open(1, file = 'risult.dat')
6
7  _____write(*,*)'Inserire xmin e xmax'
8  _____read(5,*) xmin, xmax
9  _____write(*,*)'Inserire n'
10 _____read(5,*) n
11
12 _____h = (passo tra due punti)
13 _____x = (punto iniziale)
14
```

```
15 _____do while (x .le. xmas)
16 _____    y = funzf(x)
17 _____    write(1,'(2f12.6)') x, y
18 _____    x = (aggiornamento)
19 _____end do
20 _____close(1)
21 _____stop
22 _____end
23
24 _____real*8 function funzf(x)
25 _____real*8 x
26 _____funzf(x) = x**2 - 3*x + 2
27 _____return
28 _____end
```

Grafici dei dati in un file

Salvare il file sorgente *outputfile.f* nella cartella *laboratorio4*. Compilare e successivamente eseguire il programma per vari valori dei dati in input (e.g. $I = [-2, 2]$, $n=2$, oppure $n=5$, oppure $n=20$).

Il file *risult.dat* contiene 2 colonne di numeri con i valori di x e $y = f(x)$.

- Per visualizzare tali dati si può utilizzare *gnuplot*:

```
[studente@pc laboratorio4]$ gnuplot
```

```
gnuplot> plot 'risult.dat'
```

- confrontare i dati calcolati con il grafico di $f(x)$:

```
gnuplot> plot 'risult.dat', x**2-3*x+2
```

Esercizio

Modificare il programma per fare il grafico delle seguenti funzioni:

- $f(x) = \cos(x)$ in $I = [0, 3.14]$
- $f(x) = \sqrt{e^{-x^2}}$ in $I = [0.5, 1]$
- $f(x) = e^{x/4}$ in $I = [0, 10]$

Grafici di convergenza

Consegna

- Modificare il programma *picard* implementando le funzioni g_1 , g_2 e g_3 con tre *functions*, chiamate *funzg1*, *funzg2* e *funzg3*.
- Ad ogni iterazione dell'algoritmo di Picard, stampare il numero di iterazione e lo scarto in un file di output (chiamato, per esempio *scarti.dat*). Il comando per fare un grafico in scala logaritmica lungo l'asse y è:
`gnuplot> set logscale y`
- Disegnare e i grafici di convergenza delle funzioni per cui l'algoritmo di Picard converge.

```
1  _____program picard
2  _____implicit none
3  (dichiarazione variabili)
4  _____open(1, file = 'input.dat')
5  _____open(2, file = 'scarti.dat')
6  (leggere itmax, x0, toll)
7  (inizializzare xk, iter e scarto)
8  _____do while(condizione
9  _____           di terminazione)
10 _____   iter = iter +1
11 _____   xkp1 = funzg1(xk)
12 _____   scarto = |xkp1-xk|
13 _____   xk = ...
14 _____   write(2, '(i6,e15.7)')iter,scarto
15 _____end do
16 _____close(1)
17 _____close(2)
18 _____stop
19 _____end
```

```
19 C funzione funzg1
19 _____real*8 function funzg1(x)
20 _____real*8 ...
21 _____funzg1=...
22 _____return
23 _____end
24
25 (funzione funzg2)
26
27 (funzione funzg3)
```