



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Laboratorio di Calcolo Numerico  
Laboratorio 3: Algoritmi stabili e instabili  
Programma gnuplot

Andrea Franceschini

E-mail: franceschini@dmsa.unipd.it

Dispense:

[http://www.math.unipd.it/~putti/teaching/calcolo\\_ambientale/index.html](http://www.math.unipd.it/~putti/teaching/calcolo_ambientale/index.html)

17 Marzo 2015

## Concetti base nella programmazione

Quando si usa un linguaggio di programmazione come il Fortran serve avere ben chiari alcuni concetti:

**assegnazione:** il programma valuta l'espressione a destra del simbolo  $=$  e assegna il risultato alla variabile a sinistra. Così:

- $x = y = z$  non va bene (due uguali);
- $1 = x$  non va bene (a sinistra c'è una costante);
- $x = 1$  va bene (attenzione ai tipi: interi e reali);
- usare il simbolo di uguale con un **read** / **write** non va bene.

**divisione per interi:** il Fortran interpreta diversamente l'operatore  $/$ :

- divisore e dividendo sono interi: il risultato (numero intero) è la sola parte intera, così  $1/2$  risulta 0;
- divisore o dividendo sono reali: il risultato (numero reale) è la classica divisione, così  $1.d0/2.d0$  risulta  $0.5d0$ .

## Ciclo while

Impariamo l'utilizzo dell'ultimo costrutto base nel Fortran: il costrutto *while*. Insieme ai costrutti *do* e *if* ci consente di implementare qualsiasi algoritmo. La sintassi è:

---

---

```
1: do while (condizione)
2:     ...
3: end do
```

---

---

Una semplice applicazione è l'implementazione del calcolo del fattoriale di un numero:

$$n! = n \cdot (n - 1)!$$

## Creare il programma “Fortran 90”

Aprire un terminale ed inserire i seguenti comandi:

- `[studente@pc ~]$ mkdir laboratorio3`
- `[studente@pc ~]$ cd laboratorio3`
- `[studente@pc laboratorio3]$ ls`
- `[studente@pc laboratorio3]$ gedit fattoriale.f90`

# Calcolo del fattoriale

---

```
1: program fattoriale
2: implicit none
3: integer :: n,i      ! Variabili intere
4: real*8 :: fatt     ! Variabili reali
5: write(* ,*) 'Inserire n'
6: read(* ,*) n
7: i = n
8: fatt = 1.d0
9: do while (i .ge. 1)
10:     fatt = fatt * float(i)    ! Aggiornamento variabile
11:     i = i - 1
12: end do
13: write(* ,*) 'n! = ',fatt
14: end program fattoriale
```

---

## Calcolo integrali - schema instabile

Vogliamo scrivere un programma per il calcolo dei seguenti integrali  $I_n$ :

$$I_n = \frac{1}{e} \int_0^1 x^n e^x dx, \quad n = 0, \dots, 20. \quad (1)$$

Dall'integrazione per parti si ottiene la seguente formula ricorsiva:

$$I_n = 1 - \frac{n}{e} \int_0^1 x^{n-1} e^x dx = 1 - nI_{n-1}, \quad (2)$$

con  $I_0 = 1 - 1/e$ . L'applicazione diretta di questa formula ricorsiva è *instabile*, cioè amplifica gli errori di arrotondamento.

## Calcolo integrali - schema stabile

Dalla formula ricorsiva  $I_n = 1 - nI_{n-1}$  possiamo ottenere il seguente schema all'indietro:

$$I_{n-1} = \frac{(1 - I_n)}{n}, \quad n = 20, 19, \dots, 1. \quad (3)$$

Sapendo che  $\lim_{n \rightarrow \infty} I_n = 0$ , approssimiamo  $I_n = 0$  per un  $n$  sufficientemente grande (e.g.  $I_{20} = 0$ ). L'applicazione di questa formula ricorsiva è *stabile*.

## 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)`



# Implementazione di uno schema

---

```
1: program instabile      ! oppure stabile
2: Dichiarazione delle variabili: gli integrali sono intn e intnm1
3: Inizializzazione delle variabili
4: do while (Condizione)
5:     ! La sequenza di operazioni vale nel caso INSTABILE
6:     n = n + 1 ! Aggiornamento variabile n
7:     intnm1 = intn ! Salvo il valore precedente
8:     intn = 1.d0 - n * intnm1 ! Calcolo
9:     Stampa in formato SU FILE di n e intn
10: end do
11: end program instabile    ! oppure stabile
```

---

## Grafici delle funzioni: gnuplot

È sempre utile visualizzare graficamente le funzioni e i dati che si devono utilizzare. Il programma *gnuplot* permette di visualizzare i grafici delle funzioni direttamente dal terminale.

- `[studente@pc laboratorio3]$ gnuplot`

- `gnuplot>`

**NOTA:** all'interno del programma *gnuplot* non possiamo utilizzare i comandi della shell di Linux.

- Per evitare il bloccaggio dei computer usiamo il terminal X11:

```
gnuplot> set terminal 'X11'
```

- Grafico della funzione  $y = x$ :

```
gnuplot> plot x
```

- Grafico delle funzioni  $y = x^2 - 3x + 2$  e  $y = 0$ :

```
gnuplot> plot x**2-3*x+2, 0
```

# Comandi di gnuplot 1

- Guida ai comandi di gnuplot: *help* seguito dal comando:  
`gnuplot> help plot`  
`gnuplot> help style`
- Cambiare la legenda: *title* seguito dalla legenda tra apici:  
`gnuplot> plot x**2-3*x+2 title 'y=f(x)', 0 title 'y=0'`
- Limiti dell'asse  $x$ , [xmin:xmax]  
`gnuplot> plot [0:3] x**2-3*x+2, 0`
- Limiti dell'asse  $x$  e  $y$ , [xmin:xmax] [ymin:ymax]  
`gnuplot> plot [0:3] [-1:2] x**2-3*x+2, 0`
- Nomi degli assi  $x$  e  $y$ : *set xlabel* seguito dal nome:  
`gnuplot> set xlabel 'asse x'`  
`gnuplot> set ylabel 'asse y'`  
`gnuplot> replot`
- Uscire da gnuplot: *quit*  
`gnuplot> quit`

## Comandi di gnuplot 2

- Salvare un grafico in un file con formato *png*:

```
gnuplot> set terminal 'png'
```

```
gnuplot> set output 'nomefile.png'
```

```
gnuplot> replot
```

**NOTA:** questa sequenza di comandi non visualizza il grafico sullo schermo, ma salva direttamente il grafico nel file *nomefile.png*.

- Per tornare alla visualizzazione dei grafici a schermo:

```
gnuplot> set terminal 'X11'
```

### Esercizio

- Visualizzare le seguenti funzioni in un unico grafico:

$$y = \sqrt{3x - 2}; \quad y = \frac{x^2 + 2}{3}; \quad y = \frac{x^2 - 2}{2x - 3}; \quad y = x.$$

- Salvare il grafico in un file con formato *png*.

## Comandi di gnuplot 3

- Plottare il contenuto di un file, scritto in formato XY (due colonne):

Se vogliamo una linea che collega i punti:

```
gnuplot> plot 'filename' with lines
```

Se vogliamo una linea che collega i punti, con i punti visibili:

```
gnuplot> plot 'filename' with linespoints
```

Se vogliamo soltanto i punti:

```
gnuplot> plot 'filename' with points
```

### Esercizio

- Visualizzare i valori degli integrali  $I_n$  al variare di  $n$  per i due casi (stabile e instabile).
- Salvare i grafici in formato *png*.