

Manuale rapido di Bongo

Destinato agli studenti del primo anno di Astronomia, Università di Bologna

a.a. 2010-2011

**Per far funzionare sotto WINDOWS l'editing delle righe di un file di comandi Bongo (.bon) procedere come segue (provato su Windows XP Tablet):
andare in Windows/system32 e cercare il file CONFIG.NT. Aprire questo file con notepad e aggiungere in fondo la riga
device=%SystemRoot%\system32\ansi.sys
salvare il file CONFIG.NT e poi ritornare in "prompt di comandi" (finestra nera DOS).**

1. Introduzione

Bongo è un programma di elaborazione e graficazione dati. E' stato costruito attorno ad un interprete di comandi, cioè ad un insieme di programmi in grado di leggere ed interpretare una riga di comando letta da un file esterno o immessa da tastiera.

Per questo scopo sono stati costruiti circa 80 comandi che Bongo "vede" come parole riservate al suo uso interno e che si usano unitamente a parametri numerici o stringa.

Due sono i modi di operare con Bongo:

a) Da tastiera (Dos-Windows). Si forniscono i comandi battendoli direttamente sulla tastiera e Bongo li esegue in tempo reale. Un piccolo editor interno permette di correggere gli eventuali comandi sbagliati. I comandi e il grafico "convivono" nella stessa pagina e l'effetto complessivo è di poca chiarezza anche se, con un minimo di esperienza, questo sistema risulta molto utile nel lavoro normale quando non è chiaro che tipo di grafico si vuole. Per ottenere un grafico "pulito" si usa il comando PLAY.

Una volta realizzato il grafico che si desidera, si può salvare l'insieme di comandi in un file su disco, per un uso successivo.

b) da file esterno. Con un editor si scrive un file che contenga tutti i comandi necessari ad ottenere il grafico voluto e lo si salva su disco.

Si esegue poi Bongo e da tastiera si dà il comando READ *nome-del-file che contiene i comandi*. In questo modo l'insieme dei comandi viene messo nella memoria di Bongo. Il comando PLAY (come il punto a) provvede ad eseguire tutti i comandi in sequenza.

Questo è il sistema preferibile all'inizio e alla fine, quando si ha molta familiarità con i comandi.

Bongo lavora su 4 vettori interni. Due vettori (XV,YV) sono le coordinate x e y dei dati mentre gli altri due (EXV e EYV) sono gli errori dei primi due. È possibile eseguire operazioni tra vettori utilizzando le coppie di vettori interni XV-EXV e YV-EYV.

Se Bongo viene eseguito in una finestra DOS all'interno di Windows, dopo aver disegnato il grafico usare il tasto "stampa" per salvare sulla clipboard. Poi lanciare un programma di grafica (gimp, paint, paintshop,..) e usare il comando "paste" per visualizzare, modificare e stampare il grafico.

2. Comandi principali

READ

Formato: READ nome-file

Parameteri: nomefile, il nome del file che contiene i comandi di Bongo, lungo al massimo 40 caratteri.

Scopo: Legge da un file su disco l'insieme dei comandi di Bongo e riempie la memoria. L'estensione di default è .bon

Esempi:

READ crema.xyz ! cerca il file di comandi
crema.xyz

READ crema ! cerca il file di comandi
crema.bon

PLAYBACK

Formato: PLAYBACK

Parametri: nessuno

Scopo: scorre la memoria (buffer) dei comandi, eseguendo in sequenza ognuno di loro, dal primo comando all'ultimo senza salti.

COLOR

Formato: COLOR *col*

Parametri: colore di tracciamento (0-15).

Scopo: fissa il colore di tracciamento al valore *col*. In Bongops, *col* viene cambiato in un livello di grigio tra 0 e 1 tramite la formula $grigio = 1 - col/15$. Da notare che in Postscript grigio=1 è bianco e grigio=0 è nero.

I codici e i colori disponibili in DOS sono:

0	Nero	8	Grigio
1	Blu	9	Blu brillante
2	Verde	10	Verde brillante
3	Ciano	11	Ciano brillante
4	Rosso	12	Rosso brillante
5	Magenta	13	Magenta brillante
6	Giallo	14	Giallo brillante
7	Bianco	15	Bianco brillante

Se si lavora in bianco e nero questi colori sono trasformati in toni di grigio.

PHYSICAL

Formato: PHYSICAL xiniz xfinal yiniz yfinal

Parametri: area di disegno in coordinate schermo.

Scopo: definisce le coordinate dell'area di tracciamento, in coordinate schermo. Corrisponde alla combinazione PBOX+XAXIS+YAXIS.

DIGITS

Formato: DIGITS ndx ndy oppure DIGITS

Parametri: ndx- cifre decimali per l'asse X. ndy- cifre decimali per l'asse Y.

Scopo: Fissa il numero di cifre decimali dei numeri sugli assi. I valori di default sono 2 2. Se ndx e ndy sono entrambi minori di zero, sono fissati i valori di default.

Esempi:

DIGITS !ndx=2, ndy=2

DIGITS 2 2 !ndx=2, ndy=2

DIGITS -3 -5 !ndx=2, ndy=2

DIGITS 0 1 !ndx=0 , ndy=1

LTICKS

Formato: LTICKS nx, ny

Parametri: Numero di intervalli in x e y

Scopo: definisce il numero di intervalli in cui viene diviso l'asse x. In corrispondenza dei limiti di questi intervalli (tacche) vengono scritti i valori numeri sull'asse. Il valore minimo è 1.

Vedere anche: STICKS

STICKS

Formato: STICKS nx ny

Parametri: Numero di intervalli in x e y

Scopo: Definisce il numero di intervalli piccoli in cui viene diviso ogni intervallo grande (LTICKS). Il valore minimo è 1.

Vedere anche: LTICKS

BOX

Formato: BOX p1 p2 p3 p4 oppure BOX p1 p2 oppure BOX

Parametri: p1, p2 sono interruttori (flags): se uno di loro è zero i valori numerici sull'asse corrispondente non sono scritti (p1 si riferisce all'asse X; p2 all'asse Y). Se questi parametri sono omessi, si intende che il loro valore sia 1 1 (o, come si dice, il loro valore di default è 1).

Anche p3, p4 sono flags: si riferiscono alla scrittura del valore, rispettivamente, più a destra e più in alto sugli assi. Se p3=0 non viene scritto l'ultimo valore sull'asse X; se p4=0 non viene scritto l'ultimo valore sull'asse Y. I valori di default sono 1 1.

Scopo: Disegna il rettangolo e gli intervalli sugli assi entro cui verrà realizzato il grafico, utilizzando i parametri del comando LIMITS come estremi. Se il comando LIMITS non è stato dato, sono necessari prima i comandi XCOL e YCOL per calcolare i limiti del grafico.

BOX non cancella lo schermo nè permette l'accesso alla pagina grafica (cioè il grafico non si vede). Usare

il comando MODE n (Es. MODE 18 per entrare nel modo grafico VGA) prima di ogni altro, per visualizzare il grafico. Nell'uso fuori-linea (da file esterno), il comando PLAY(BACK) incorpora la chiamata a MODE.

I valori di p1, p2, p3 e p4 dovrebbero essere zero (tutti o in parte) quando si creano mosaici di grafici uniti (ad esempio 4 grafici con le stesse scale, sullo stesso foglio).

Vedere anche: CLABEL, XLABEL, YLABEL

XLABEL

Formato: XLABEL stringa

Parametri: stringa, un testo di 40 caratteri al massimo.

Scopo: Scrive *stringa* centrata sotto l'asse X del grafico. Come angolo di scrittura viene usato l'ultimo definito in TEXT (se necessario dare un altro comando TEXT con angolo=0)

Vedere anche: BOX, CLABEL, YLABEL

GRID

Formato: GRID aspetto , colore, flag

Parameters: aspetto e colore della griglia, se flag=1 disegna la griglia anche in corrispondenza delle suddivisioni piccole.

Scopo: disegna una griglia, sovrapposta al disegno.

Gli aspetti disponibili sono:

0	Nessuna linea
1	'-----'
2	'-----'
3	'-----'
4	'-----'
5	'.....'
6	'.....'
7	'---.---.---'

Esempio:

GRID 6 8 2	disegna una griglia a punti (6), in grigio (8), in corrispondenza delle divisioni principali (non 1).
GRID 4 4 1	disegna una griglia a tratti (4), in rosso (4), in corrispondenza di tutte le divisioni (1).

Errori: nessuno, ma usare il comando prima di LABEL, così le scritte possono sovrascrivere la griglia e non viceversa.

DATA

Formato: DATA nome-file oppure DATA ?

Parametri: nome-file. La lunghezza massima è di 40 caratteri

Scopo: Rende disponibile il file che contiene i dati da graficare. Il file deve essere organizzato in colonne, separate da spazi o da virgola. fino ad un massimo di 255 caratteri. Se sono presenti caratteri non numerici (nel senso che è presente una colonna che contiene tutti caratteri non numerici), questi vengono trasformati nel valore 9999

Se nome-file è '?', bongo chiede di introdurre il nome da tastiera.

LINES

Formato: LINES L1 L2

Parametri: prima ed ultima linea

Scopo: definisce la prima e l'ultima linea utilizzata nel file di dati attualmente aperto. I vettori interni XV e YV saranno inizializzati con i valori del file di dati a partire dalla riga L1 per finire alla riga L2. **Il file di dati deve contenere una riga bianca come ultima linea.**

Se L1 e L2 sono entrambi zero, Bongo chiede che questi dati vengano immessi da tastiera.

XCOLUMN

Formato: XCOLUMN n

Parametri: numero di una colonna nel file dei dati

Scopo: Assegna la colonna numero *n* del file dei dati al vettore interno di Bongo, XV. Se non è stato dato il comando LIMITS, questo comando calcola anche il massimo e il minimo del vettore.

XOPER

Formato: XOPER codice, costante

Parametri: codice dell'operazione scelta, costante da usare nell'operazione (o 1 o 0, se necessario).

Scopo: Esegue l'operazione, definita da *codice*, sul vettore interno XV.

Codici disponibili:

Operation	Opcode	Operation	Opcode
1) vector+constant	(+)	7) vector/vector(constant)	(7)
2) vector*constant	(*)	8)vector-vector(constant)	(8)
3) vector/constant	(/)	9) 10**(vector*constant)	(10^,10**)
4) LOG10(vector)*constant	(log,LOG)	10) EXP(vector*constant)	(exp,EXP)
5) LN(vector)*constant	(ln,LN)	11) vector/ \sum (vector values)	(11)
6) vector**constant	(^,**)	12) vector to hh.mmss or dd.mmss	(12)

13) $\sum(1:i)\text{vector} / \sum(1:N)\text{vector}$	(13)		
21) $\text{vector1} \pm (\text{vector2} * \text{const})$	(21)	24) $\text{vector1} = \text{vector2}$	(24)
22) $\text{vector1} * (\text{vector2} * \text{const})$	(22)	25) $\text{vector2} = \text{vector1}$	(25)
23) $\text{vector1} / (\text{vector2} * \text{const})$	(23)		

Il vettore interno XV risulta modificato da OGNUNA delle operazioni riportate nella tabella.

Le modifiche restano valide fino all' XOPER successivo.

Il logaritmo di un numero negativo è posto uguale a -50.

Codici 7 e 8: consentono la normalizzazione rispetto ad uno qualsiasi degli elementi del vettore XV (la costante sarà definita come il numero d'ordine di questo elemento).

Codice 11: permette la normalizzazione rispetto alla somma di tutti i valori del vettore XV. Viene creato il file SUM.OUT, contenente la somma del vettore XV.

Codice 12: si assume che i dati siano espressi in termini delle unità più basse (ad esempio in *secondi d'arco* se si richiedono *gradi.minutisecondi*).

Codice 13: calcola la funzione di distribuzione empirica (funzione cumulativa) del vettore XV, precedentemente binnato (comando BIN). Viene creato il file SUM.OUT che contiene la somma dei valori di XV.

Codici maggiori di 20: permettono operazioni tra i vettori XV e EXV (nella tabella sono chiamati V1 e V2). Il risultato dell'operazione viene scritto in XV (cioè in V1).

Vedere anche: EXOPER, EYOPER, YOPER

Esempio:

Viene presentato solo un esempio riferito ai codici 20-25. Si supponga di leggere due file, dai quali debbano essere estratti 3 vettori di dati; un'ascissa comune ad entrambi i vettori dati e due ordinate (o1 ed o2 dai file file1.dat e file2.dat, rispettivamente). Le ordinate debbano essere elaborate per dare un nuovo vettore $o3 = \log(o1) - \log(o2)$. Il file di comandi è allora:

```
LIMITS      x1 x2 y1      ! limiti in coord. utente
            y2

BOX          ! traccia gli assi e le divisioni

DATA        file2.dat  ! nota: per primo si legge file2.dat

LINES       12 200    ! da riga 12 a riga 200

XCOL        1         ! mette l'ascissa in XV

YCOL        2         ! la 2.a ordinata, o2, in YV

YOPER       log 1     ! log (o2)

YOPER       25 1      ! mette log(o2) in EYV (cioè V2=V1)

DATA        file1.dat  ! legge il primo file file1.dat

LINES       12 200    ! da riga 12 a riga 200
```

YCOL 2 ! la 1.a ordinata o1 in YV

YOPER log 1 ! log (o1)

YOPER 21 -1 ! sottrae: $YV=YV+(-1*EYV)$

LTYPE 1 15 ! fissa un riga continua (1), bianca (15).

CONNECT ! congiunge i punti

Errori: nessuno, ma verificare che nell'operazione 23, V2 non sia nullo.

PTYPE

Formato: PTYPE codice, dimensione, colore

Parametri: codice numerico per la forma del simbolo, dimensione (lato o raggio) in pixel, colore

Scopo: Definisce il codice del simbolo, la dimensione e il colore. I simboli sono poi effettivamente disegnati dal comando POINTS.

I codici e i simboli disponibili sono:

0	Nessun Simbolo
1	+
2	x
3	Triangolo
4	Quadrato
5	Rombo
6	Cerchio
7	Punto (1 pixel)
8	Asterisco
9	Stella (dimensione minima 3 pixel)

PTYPE resta valido fino alla successiva chiamata PTYPE.

Vedere anche: POINTS, fig9.bon

POINTS

Formato: POINTS

Parametri: nessuno

Scopo: Disegna un simbolo, definito dall'ultimo comando PTYPE, alle coordinate x,y memorizzate nei vettori interni XV e YV.

Vedere anche: PTYPE

LTYPE

Formato: LTYPE codice, colore

Parametri: codice numerico per la forma della linea, colore della linea.

Scopo: Definisce la forma e il colore della linea che sarà disegnata con il comando CONNECT.

I codici e le forme disponibili sono:

0	Nessuna linea
1	-----
2	-----
3	-----
4	-----
5
6
7	--- . --- . ---

Vedere anche: CONNECT

CONNECT

Formato: CONNECT

Parametri: nessuno

Scopo: Unisce i dati sul grafico con l'ultima linea definita (comando LTYPE).

Vedere anche: LTYPE

RELOCATE

Formato: RELOCATE x y

Parameters: x, y in coordinate utente

Function: posiziona la penna scrivente (il punto corrente o current point) a x,y in coordinate utente (cioè nelle unità dell'utente). Definisce dove inizierà la prossima operazione di scrittura sullo schermo.

END

Formato: END

Parametri: nessuno

Scopo: Esce da Bongo. Questo comando non salva i comandi presenti in memoria (usare WRITE per questo). Durante la fase interattiva di inserimento (INSERT) di nuovi comandi nella memoria di Bongo, END termina l'inserimento. Usato con Bongops (la versione PostScript) END chiude il file di uscita bongo.ps e scrive sul video "Bongo.ps written on disk". Scrive anche il logo 'Bongo-Linux 1.4' nell'angolo in basso a destra del foglio.

In pratica, i comandi di uscita da Bongo sono tre: END, OFF, QUIT.

END esce da Bongo e scrive il logo; OFF non esce da Bongo e non scrive il logo: in Bongops chiude solo il file bongo.ps.; QUIT agisce come END, ma non scrive il logo.

END è per l'uso normale; OFF per '*guarda - correggi - rilancia*' e QUIT quando il grafico va inserito in un testo e il logo disturba.

Vedere anche: OFF, QUIT, WRITE

3. Esempi di file di comandi, di file di dati e dei grafici corrispondenti

In generale un file di comandi Bongo è composto da 3 sezioni:

1	Sezione generale: sono definiti i commenti generali, le dimensioni fisiche del grafico, se il grafico è a colori o no, lo spessore delle righe, il numero di cifre decimali.
2	Sezione relativa al grafico: si definiscono i limiti, il numero di divisioni (grandi e piccole) degli assi, il tipo di squadrettatura, le scritte sugli assi, la lettura dei dati, il tipo di punti e di linee e il loro disegno.
3	Commenti finali: annotazione del file di dati da cui è tratto il grafico, data ed ora.

Come esempio viene ripreso, un pò modificato, il file di comandi relativo al pianerottolo geiger riportato subito dopo.

Sezione 1

```
%legge geiger-np.dat 11.9.97
```

```
rgb 1
```

```
physical 100 400 100 400
```

```
color 15
```

```
wline 0.5
```

```
digits 0 0
```

Sezione 2

```
limits 300 600 0 140
```

```
lticks 3 7
```

```
sticks 5 4
```

box

xlabel Tensione (Volt)

ylabel Conteggi/min x 10

clabel pianerottolo geiger

wline 1

data geiger-np.dat

lines 2 20

xcol 1

ycol 2

yoper / 10

eycol 2 1

eyoper / 10

eycol 2 1

eyoper ^ 0.5

eyoper / 10

fill

ptype 6 4 15

points

nofill

ltype 1 15

connect

Sezione 3

reloc -3 280

text 0

label geiger-np.dat 15.1.97

id fz

%

**1) disegna il pianerottolo del contatore geiger
dati: geiger-np.dat**

comandi Bongo

%legge geiger-np.dat	geiger nuovo - pianerottolo 11.9.97
rgb 1	325 90
physical 100 400 100 400	327 818
color 15	330 1013
wline 0.5	350 1090
digits 0 0	375 1128
limits 300 600 0 140	400 1078
lticks 3 7	400 1078
sticks 5 4	450 1102
box	500 1169
xlabel Tensione (Volt)	550 1135
ylabel Conteggi/min x 10	575 1152
clabel pianerottolo geiger	
wline 1	
data geiger-np.dat	
lines 2 20	
xcol 1	
ycol 2	
yoper / 10	

eycol 2 1

eyoper ^ 0.5 ! radice quadrata

eyoper / 10 ! diviso 10

fill

ptype 6 4 15

points

nofill

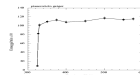
ltype 1 15

connect

id fz

%

Il risultato è:



N.B.: Nel file di comandi precedente viene **prima** fatta la radice quadrata dei conteggi [yoper 6 0.5] e **poi** la divisione per 10 [yoper 3 10]. Questo perchè la propagazione degli errori ci dice

che se $x=1420$ e $\sigma(x)=10$, $x=37.7$, nel dividere x per 10 ottengo un errore pari a $\sigma(x)/10$ e **non** a $\sigma(x/10)$. Una verifica si può avere dal calcolo dell'errore relativo nei due casi.

2) Disegna il fit dei minimi quadrati del geiger

N.B.: usa i conteggi/min non i ln(conteggi)

dati: cg222982.dat

comandi Bongo

%legge cg222982.dat

Geiger CG2 2.2.98

rgb 1 0 1491

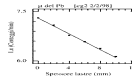
delta 5 5 2 1195

physical 100 299 100 299 4 874

color 15 6 723

```
wline 0.5                8 589
digit 0 1                10 462
limits 0 12 6 7.5
lticks 3 1
sticks 5 5
box
xlabel Spessore lastre (mm)
ylabel Ln (Conteggi\min)
label
clabel \m del Pb [cg2 2/2/98
]
wline 1
data cg222982.dat
lines 2 20
xcol 1
ycol 2
yoper ln 1                (ln dei conteggi)
fill
ptype 4 2 15
points
nofill
ltype 1 15
reloc 0 7.3
draw 10 6.1
%
```

Il risultato è:



3) disegna l'andamento della temperatura nel calorimetro

dati: c312296.dat Comandi Bongo

T(° C) t(s)

18.7	0	COLOR 15
18.7	30	DELTA 5 3
18.6	60	LIMITS 0 150 18 43
18.6	90	DIGITS 0 0
23.0	93	LTICKS 5 5
23.5	96	STICKS 3 5
24.3	99	GRID 6 3 1
25.5	102	BOX
26.3	105	CLABEL Temperatura/tempo (c312296)
27.0	108	XLABEL Tempo (Secondi/10)
27.6	111	YLABEL Temperatura (Gradi)
28.5	114	DATA c312296.dat
29.0	117	LINES 3 73
29.5	120	XCOLUMN 2
30.0	123	YCOLUMN 1
30.5	126	XOPER 3 10
31.2	129	PTYPE 2 2 15
32.5	132	POINTS
32.7	135	LTYPE 1 15

33.0 138 CONNECT

33.3 141

33.8 144

34.3 147

34.6 150

34.6 153

34.7 156

35.0 159

35.3 162

35.5 165

35.7 168

36.0 171

36.3 200

36.0 225

35.7 228

35.6 231

35.5 234

35.5 242

35.4 245

35.4 248

35.4 251

35.3 254

35.2 257

35.2 278

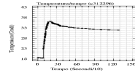
35.1 281

35.0 284
35.0 310
34.9 316
34.8 323
34.6 331
34.4 341
34.3 344
34.3 353
34.2 356
34.1 359
34.0 362
34.0 401
33.9 404
33.9 436
33.8 466
33.5 496
33.5 526
33.4 556
33.2 586
33.0 646
32.9 706
32.8 766
32.6 826
32.5 886
32.3 1006

32.2 1126

32.1 1286

Il risultato è:



4) disegna l'andamento della sensibilità della bilancia con il carico

comandi Bongo

dati: b15296.dat

color 15 0 0.340 0.043

digits 0 2 10 0.290 0.043

limits 0 50 0.2 0.4 20 0.270 0.043

lticks 5 4 50 0.230 0.043

sticks 5 5

delta 5 5

box

clabel "Sensibilita' in funzione del
carico"

xlabel Peso (g)

ylabel Sensibilita' (div./mg)

eycol 3 1

fill

ptype 9 3 15

points

eycol 3 0

data b15296.dat

lines 1 6

xcol 1

ycol 2

eycol 3 1

fill

ptype 7 1 15

points

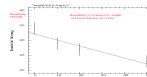
ltype 1 15

reloc x1 y1 !sostituire i valori reali

draw x2 y2 !sostituire i valori reali

in cui x1 e y1 sono il peso iniziale (0 grammi) e la sensibilità **calcolata** per questo peso e x2,y2 il peso finale (50 grammi) e la sensibilità **calcolata** per quest'ultimo peso.

Il risultato è:



5) Un esempio più complesso

[parametri di ammassi globulari](#)

[file di comandi ...](#)

[file dei dati ...](#)

[... e grafico](#)

[Esempi delle prove di laboratorio](#)

Ultimo aggiornamento: 27.10.10