

Il grande entusiasmo causato dalla recente scoperta dei codici macchina per la programmazione del microprocessore LH5801 ha provocato, in questi ultimi mesi, la monopolizzazione della rubrica da parte di programmi e routine caratterizzati da listati esadecimali poco facilmente interpretabili da lettori meno esperti. Non dobbiamo comunque dimenticare che il PC-1500 è stato progettato principalmente come un potente strumento di calcolo portatile, e che la sua utilizzazione più specifica è quella della programmazione scientifica in Basic.

È per questo che nella rubrica di questo mese pubblichiamo due programmi prettamente scientifici che possono fornire un valido supporto di calcolo a tutti coloro i quali svolgono le loro attività nei settori della matematica, fisica ed elettronica.

### Calcolo dei limiti di una F(X)

Mauro Regio - Civitavecchia (RM)

Questo programma, data una qualsiasi funzione di una variabile tipo  $y=f(x)$ , calcola, se esiste, il valore del limite per  $x$  che tende a un valore finito o per  $x$  che tende ad infinito. La prima parte del programma (righe 5 ÷ 90) gestisce l'input della funzione e la scelta del tipo di limite; naturalmente si può far tendere la  $x$  al valore voluto da valori più grandi o da valori più piccoli ( $x \rightarrow x_0 +$  oppure  $x \rightarrow x_0 -$ ), come pure si può scegliere che  $x$  tenda a più o meno infinito.

Questa parte del programma è molto semplice e basta un'occhiata al listato per comprenderne il funzionamento.

Da questo punto in poi il programma procede in modo diverso a seconda del valore assunto dal flag T ( $T=0 \ x \rightarrow x_0$ ,  $T=1 \ x \rightarrow \infty$ ).

Si hanno così due diverse inizializzazioni (righe 105 ÷ 130); i valori dati in questa sede alle variabili E, D(I), F(I) e H(I), X1 ed X2 sono stati accuratamente tarati con molteplici prove, in modo da garantire una precisione ed una velocità notevoli.

Le righe 150 ÷ 185 gestiscono differenziatamente la variabile  $x$  nei due casi suddetti.

Per controllare se la funzione converge ad un valore finito oppure diverge, vengono fatti due test separati, righe 250 ÷ 260 e righe 265 ÷ 300. Questi due test sfruttano il criterio di convergenza di Cauchy, e la definizione di limite per  $x \rightarrow \infty$ ; si capirà meglio ora perché vengono calcolate e confrontate coppie di valori assunti dalla funzione.

Il tutto viene controllato dal programma tramite il valore assunto da S, righe 200 ÷ 240. I due test suddetti vengono effettuati con due diversi gradi di precisione in momenti diversi.

Inizialmente, se  $S=1$ , essi vengono effettuati entrambi in modo da stabilire la "tendenza" della funzione; se uno dei due test ha esito positivo viene cambiato il valore di S ( $S=2$  convergenza,  $S=3$  divergenza), in maniera da eseguire, da quel momento in poi, solo il test che ha dato esito positivo, per il quale viene inoltre aumentato il grado di precisione. Questo permette di eseguire un numero di iterazioni minore, con una velocità molto elevata

per calcoli di questo tipo nell'ordine dei cinque/quindici secondi.

Come è noto a tutti non è detto che il limite esista: questo caso non è contemplato dal programma che infatti continua a girare senza avere esito positivo nella sua ricerca. D'altra parte questo programma serve per controllare il valore del limite previamente calcolato con le regole dell'analisi; anche se vi accorgete ben presto della sua potenza!

Quindi nel caso in cui il programma non desse risultati immediati, un test manuale sui valori di K e K1 (coppie di valori della  $f(x)$ ) e su X e C danno già un'idea dell'andamento della funzione.

Una volta calcolato il valore del limite, se esso è un valore finito, viene opportunamente arrotondato nelle righe 300 ÷ 310, quindi si passa alla stampa definitiva ove vengono sfruttate anche le buone caratteristiche grafiche del display.

Solo poche parole di precauzione dovute soprattutto ad alcuni limiti di calcolo della macchina: attenzione a  $\cos x$  per  $x \rightarrow 0$  (diventa 1 troppo presto!), le funzioni esponenziali per  $x \rightarrow \infty$  assumono valori troppo elevati con pochi passaggi. Per evitare questo basta fare delle previsioni particolari nella subroutine di calcolo della  $f(x)$ , usando l'istruzione "on error goto".

### Calcolo dell'impedenza di un circuito

di Marco V. Arbolino - Roma

Il circuito elettrico viene inserito nel pro-

```

1:REM *** LIMIT
1 ***
5: CLEAR :F$="Lim
ite = "
10: INPUT "Funz. I
ns. (Y/N)?":A$
15: IF A$="N"PRINT
"Insertare in 5
00":STOP
20: INPUT "X --> X
o (Y/N)?":A$
30: DIM D(1),F(1),
H(1): IF A$="N"
LET T=1:GOTO 6
0
35: INPUT "Xo= ?":
M:PAUSE "Xo+ o
r Xo- (+/-)?":
40: C$=INKEY$: IF
C$="+"LET I=0:
GOTO 55
45: IF C$="-"GOTO 4
0
50: I=1
55: GOTO 90
60: PAUSE "+ or -
Inf. (+/-)?":C
$=INKEY$
65: IF C$="+"LET J
=1:GOTO 90
70: IF C$="-"GOTO 6
0
75: J=-1
90: CLS
105: IF T=1GOTO 120
110: E=1E-4:D(0)=1E
-5:D(1)=-D(0):
F(0)=-1E-6:F(1
)=F(0):X2=100
:H(0)=1E4
115: GOTO 130
120: E=.1:X1=1E4:X2
=X1:X1=J*X1:H(
1)=1E9
130: S=1
150: IF T=0GOTO 160
155: A=X1:X=A:GOTO
165
160: A=M+D(1):X=A
165: GOSUB 500:K=Y
170: IF T=1GOTO 180
175: D(1)=D(1)+F(1)
:F(1)=(M-X)/I0
:B=M+D(1):GOTO
185
180: B=A+X
185: X=B:GOSUB 500:
K1=Y
200: ON S GOTO 205,2
10,215
205: GOSUB 250:
GOSUB 265:GOTO
230
210: GOSUB 250:GOTO
230
215: GOSUB 265
230: IF T=1GOTO 240
235: K=K1:GOTO 245
240: A=X: IF S=3LET
A=K1
245: GOTO 170
250: C=ABS (K1-K):
IF C>ERETURN
255: IF S=2GOTO 300
260: E=1E-6:S=2: IF
T=1LET E=1E-3
262: RETURN
265: IF ABS K1<X2
RETURN
270: IF S=3GOTO 300
275: X1=J*X1E0:X2=H(
T):S=3:RETURN
300: IF S=3GOTO 320
305: O=1E3:R=(INT (
K1*O)/O): IF K1
-R<=5E-4LET K1
=R:GOTO 315
310: K1=R+1E-3
315: BEEP 1:PRINT F
$:K1:GOTO 335
320: IF K1>0LET Z=1
325: B$=CHR$(45-Z*
2):C$="IC2222
21400":D$="142
222221C":WAIT
3:PRINT F$:B$:
:WAIT 300
330: BEEP 2:GPRINT
C$:D$:
335: CLS :END
500: Y=(X)^(1/X)
505: RETURN
1070 Bytes

```

Figura 1 - Listing del programma: Calcolo dei limiti di una F(X).

Limite	Risultato	Valore teorico	Tempo impiegato
$\lim_{x \rightarrow 0} \frac{\sin x}{x} =$	1	1	2 sec.
$\lim_{x \rightarrow 0} \frac{\tan x}{x} =$	1	1	2 sec.
$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x} =$	0	0	2 sec.
$\lim_{x \rightarrow 0} \frac{\log(1+x)}{x} =$	1	1	2 ÷ 3 sec.
$\lim_{x \rightarrow 0} \frac{(1+x)^2 - 1}{x} =$	$\alpha$	$\alpha$	2 ÷ 3 sec.
$\lim_{x \rightarrow 0} \frac{a^x - 1}{x} =$	log a valore approssimato	log a	3 sec.
$\lim_{x \rightarrow \pm \infty} \frac{x^2 + 1}{nx} =$	$\pm \infty$	$\pm \infty$	5 sec.
$\lim_{x \rightarrow +\infty} x^{1/x} =$	1	1	3 sec.
$\lim_{x \rightarrow 0^+} x \log x =$	0	0	7 sec.
$\lim_{x \rightarrow 0^+} x^x =$	1	1	8 sec.

Figura 2 - Risultati e tempi di esecuzione del programma di calcolo dei limiti.



gramma come una stringa, A\$(0), che può contenere fino ad 80 caratteri. La stringa deve iniziare con una parentesi aperta e terminare con una parentesi chiusa; la serie di due o più elementi o gruppi di elementi viene indicata facendo precedere all'elenco degli elementi stessi il carattere "+", mentre per il parallelo si usa il carattere "=".

Gruppi di elementi vengono distinti racchiudendoli tra parentesi, mentre i singoli elementi vengono indicati dalle lettere maiuscole in ordine alfabetico. Ad esempio, al circuito indicato in figura 4 corrisponde la stringa:

A\$(0) = "(+A(=(+BC)(+D(=EF)))G)"

In primo luogo il programma scorre la stringa A\$(0) contando il numero delle parentesi. La variabile B\$ contiene di volta in volta un diverso carattere di A\$(0), la variabile A viene incrementata di 1 ogni volta che B\$ = "(" e diminuita di 1 ogni volta che B\$ = ")". Nella variabile K è registrato il massimo valore raggiunto da A e quindi il massimo numero di parentesi nidificate (linee 80-110).

A questo punto si cercano le parentesi più interne utilizzando le variabili B\$ da A come prima. Quando A = K si è arrivati ad una di queste parentesi ed il programma passa ad una subroutine dove viene eseguito il calcolo vero e proprio e viene modificata di conseguenza la stringa A\$(0). Quando questa stringa è stata scandita completamente, K viene diminuito di 1 e si ricomincia con la ricerca ed il calcolo delle parentesi immediatamente più esterne, fi-

no a quando K = 0 (linee 120-210). Quando il programma salta alla subroutine di calcolo, la variabile B\$ è puntata sulla parentesi aperta. Si procede quindi ad esaminare i caratteri successivi (saltando il primo che è sempre un carattere "+" o "-"), che contengono i nomi degli elementi. Sfruttando l'istruzione ASC, il valore dell'impedenza dell'elemento (parte reale e parte immaginaria) viene letto dai vettori R(i) e I(i), che contengono i valori di tutti gli elementi, e posto nei vettori W(i) e V(i) che contengono i valori delle impedenze usati nel calcolo della parentesi. A questo punto si controlla, utilizzando il carattere saltato in precedenza, se si tratta di una serie o di un parallelo. Il calcolo viene effettuato, in base alle formule date

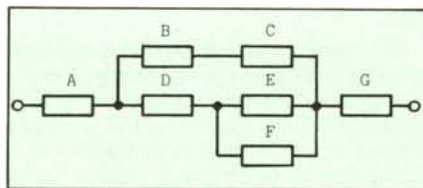


Figura 4 - Circuito dell'esempio citato nel testo.

in precedenza, alle linee 400-410 per il parallelo ed alla riga 500 per la serie.


Il risultato del calcolo viene posto nei vettori R(i) ed I(i) immediatamente di seguito alle posizioni occupate dai valori impostati inizialmente o calcolati in precedenza.

I vettori R(i) ed I(i) ed i vettori W(i) e

V(i) comprendono 26 elementi ciascuno. Data la lunghezza del programma è possibile definire vettori di queste dimensioni solamente se si dispone di un'espansione di memoria; in caso contrario il numero di elementi per ogni vettore è sensibilmente minore. Con la capacità di memoria standard (1850 byte) e cancellando le prime righe che servono unicamente come intestazione, restano 446 byte che vanno divisi tra la stringa A\$(0) ed i quattro vettori. Riducendo la lunghezza di A\$(0) a 40 caratteri restano 406 byte, sufficienti a circa 50 variabili numeriche. Una suddivisione possibile si ottiene dimensionando i vettori W(i) e V(i) con 5 elementi ciascuno ed i vettori R(i) e I(i) con 20 elementi ciascuno.

Il programma parte con i tasti DEF A ed inizia chiedendo il numero di elementi del circuito, la stringa A\$(0) che definisce il circuito, e la frequenza della corrente che percorre il circuito. Se gli elementi sono solo resistenze e la corrente è continua, la frequenza è irrilevante e le si può attribuire un valore qualunque (es. 0).

Quindi il programma richiede l'inserimento del valore degli elementi in ordine alfabetico, specificando per ciascun elemento la sua natura (capacità, induttanza o resistenza). Per ogni elemento è sufficiente inserire il valore della resistenza in ohm, della capacità in farad e dell'induttanza in henry.

Al termine del calcolo il programma fornisce la parte reale, la parte immaginaria, il modulo e la fase dell'impedenza. 

```

2: REM *****
3: REM Calcolo
4: REM impedenze
5: REM di
6: REM Marco
7: REM Arbolino
8: REM *****
10: "A": CLEAR: DIM
  A$(0)*80, R(25)
  , I(25), W(25), U
  (25): INPUT "Nu
  mero degli ele
  menti?"; N
20: INPUT "Circuit
  o?"; A$(0):
  INPUT "Frequen
  za?"; 0
30: PAUSE "Inserim
  ento dati":
  PAUSE "In ordi
  ne alfabetico":
  FOR I=0 TO N-1
40: INPUT "Tipo el
  emento? (R/C/L
  ?)"; A$: IF A$(0)
  "R" AND A$(0) "C"
  AND A$(0) "L"
  THEN 40
50: INPUT "Valore?"
  ; A$: IF A$ = "R"
  LET R(I) = A: I(I
  ) = 0: NEXT I
60: IF A$ = "C" LET R
  (I) = 0: I(I) = -1 /
  (A * 0): NEXT I
70: IF A$ = "L" LET R
  (I) = 0: I(I) = A * 0
  : NEXT I
80: CLS: A = 0: FOR I
  = 1 TO LEN A$(0)
  : B$ = MID$(A$(0)
  ), I, 1)
90: IF B$ = "(" LET A
  = A + 1: IF A > K LET
  K = A
100: IF B$ = ")" LET A
  = A - 1
110: NEXT I
120: I = 1: A = 0
130: B$ = MID$(A$(0)
  ), I, 1)
140: IF B$ = "(" LET A
  = A + 1: IF A = K
  GOSUB "CAL":
  GOTO 120
150: IF B$ = ")" LET A
  = A - 1
160: IF I = LEN A$(0)
  THEN 200
170: I = I + 1: GOTO 130
200: IF K = 0 GOTO "RE
  S"
210: K = K - 1: GOTO 120
300: "CAL": J = I + 2: Q =
  0: W = 0: U = 0
305: B$ = MID$(A$(0)
  ), J, 1)
310: IF B$ = "(" THEN
  "POS"
320: G = J - 1 - 2: H = ASC
  B$ - 65: W(G) = R(H)
  : U(G) = I(H): Q =
  Q + 1
330: J = J + 1: GOTO 305
350: "POS": B$ = MID$(
  A$(0), I + 1, 1)
360: IF B$ = "=" GOSUB
  "PAR": RETURN
370: IF B$ = "+" GOSUB
  "SER": RETURN
400: "PAR": FOR J = 0
  TO Q - 1: M = W(J) ^
  2 + U(J) ^ 2: W(J) =
  W(J) / M: U(J) = -U
  (J) / M: W = W + W(J)
  : U = U + U(J)
410: NEXT J: M = W ^ 2 + U
  ^ 2: W = W / M: U = -U /
  M: GOTO "CHA"
500: "SER": FOR J = 0
  TO Q - 1: W = W + W(J)
  : U = U + U(J):
  NEXT J: GOTO "C
  HA"
600: "CHA": R(N) = W: I
  (N) = U: N = N + 1
610: C$ = CHR$(64 + N)
  : A$(0) = LEFT$(
  A$(0), I - 1) + C$ +
  RIGHT$(A$(0),
  (LEN A$(0) - (I +
  2 + Q))) : RETURN
700: "RES": PAUSE "I
  mpedenza total
  e": PRINT "Par
  te re. ="; R(N - 1)
  )
710: PRINT "Parte i
  mm. ="; I(N - 1):
  PRINT "Modulo
  ="; J(R(N - 1) ^ 2 +
  I(N - 1) ^ 2)
713: IF R(N - 1) = 0 AND
  I(N - 1) > 0 LET F =
  90: GOTO 720
715: IF I(N - 1) < 0 AND
  R(N - 1) = 0 LET F =
  -90: GOTO 720
717: F = ATN(I(N - 1) /
  R(N - 1))
720: PRINT "Fase ="
  ; F: PAUSE "Per
  farne ancora p
  remi 0": PAUSE
  "Se no un tast
  o qualunque"
730: F$ = INKEY$: IF
  F$ = "" THEN 730
740: IF F$(0) = "0" END
750: GOTO "A"

```

Figura 3 - Listing del programma "Calcolo dell'impedenza di un circuito".





Hai già un personal computer? Vuoi acquistarne uno? Accademia ti propone un corso teorico-pratico di facile comprensione anche per chi si avvicina per la prima volta all'informatica, che ti mette fin dal primo giorno in contatto diretto col computer.

Imparerai prima a digitare e poi a programmare sul VIC 20, sul COMMODORE 64 o sull'ATARI 400 che, se lo desideri, potremo

fornirti a condizioni particolarmente interessanti o in comode rate.

Imparare a esprimersi in BASIC è indispensabile a chi vuole saper utilizzare correttamente il computer per dare una nuova dimensione al proprio lavoro, ai propri studi o all'amministrazione familiare, o per acquisire conoscenze che potranno rivelarsi indispensabili subito o in un futuro molto prossimo.

Con il corso Accademia il linguaggio BASIC in poche settimane non avrà più segreti per te, e il personal diventerà il tuo collaboratore più affidabile, sempre disponibile per la rapida soluzione di ogni tipo di problema.

A fine corso un certificato di studio testimonierà la tua preparazione.

Vuoi saperne di più? Spedisci oggi stesso il tagliando.

# studio, lavoro, hobby, tempo libero



*corsi*  
**ACCADEMIA**  
*per imparare a casa*

**se vuoi guadagnare tempo**

Detta alla nostra segreteria, **chiama Roma** funzionante 24 ore su 24, **06/62.30.341** nome, cognome, indirizzo e corso che ti interessa. Riceverai immediatamente le informazioni.



ACCADEMIA - Via Diomede Marvasi 12/ R 00163 Roma

Desidero ricevere informazioni sul vostro corso di programmazione BASIC.

**R530**

Cognome \_\_\_\_\_ Nome \_\_\_\_\_

Via \_\_\_\_\_ N. \_\_\_\_\_

Città \_\_\_\_\_ C. A. P. \_\_\_\_\_ Prov \_\_\_\_\_ Età \_\_\_\_\_

Motivo della richiesta  studio  lavoro  hobby. Desidero informazioni sul  solo corso  corso + personal computer

Se risiedi in Germania, in Svizzera, scrivi direttamente a:

R.L. Conc. Accademia - 7140 LUDWIGSBURG - MARBACHER STR 37 - tel. 07141/57039  
IBERCULTURA GmbH, Baselstrasse, 67 - 6003 LUZERN - tel. 041/226.617

# IL BITTEGONE di FELICE PAGNANI

*è un SUSY...  
non fatevi imbarazzare  
dalla scelta*

SUSY 2-48K	L. 732.000
SUSY 2-48KS	L. 780.000
SUSY 2-48K/SS	L. 884.000
SUSY 2-64K/S	L. 940.000
SUSY 2-64K/Z80	L. 1.140.000
SUSY 2-64K/Z80S	L. 1.228.000
SUSY 2-64K/TC	L. 1.340.000
SUSY 2-PC1	L. 1.372.000
SUSY 2-PC2	L. 1.440.000
SUSY 2-PC3	L. 1.528.000
SUSY 5-PCIBM	L. 6.000.000
SUSY 5-PCIBMDD	L. 6.500.000

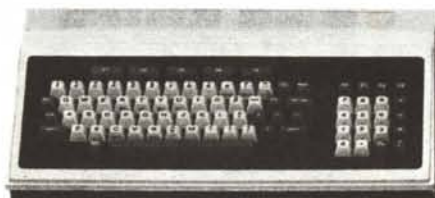
PREZZI I.V.A. INCLUSA



DRIVES: MINIFLOPPY, WINCHESTER, SLIM 8"  
SOFTWARE PER SUSY, CP/M E PC IBM  
OLTRE 50 SCHEDE DI INTERFACCIA



TASTIERE SEPARATE



STAMPANTI A IMPATTO (GRAFICHE COLORE E PER W.P.) PER SUSY E PC IBM  
MONITOR COLORE E B/N, VERDE E AMBRA, 12, 14, 15, 20 E 26 POLLICI

Felice Pagnani - Via U. Comandini, 49 - 00173 ROMA - Tel. 06/6133025-7970559

**Tutti i prezzi nella GUIDA MERCATO**