# Alcuni problemi di Computer Grafica Bidimensionale

*La Computer Grafica Bidimensionale è un argomento sicuramente meno affascinante e spettacolare rispetto alla Computer Grafica Tridimensionale, ma é altrettanto interessante per le sue molteplici applicazioni e sicuramente più semplice sia da comprendere che da realizzare.*

*Tutti infatti, anche i non addetti ai lavori, sono abituati a disegnare o scarabocchiare su un foglio di carta, magari per passare il tempo, oppure per rendere più efficace l'esposizione di un concetto, o addirittura per esprimere qualcosa di interiore non altrimenti esprimibile. Fare lo stesso con un microcomputer non è tanto più complicato....*

*In questo numero ci occuperemo di due problemi di Computer Grafica Bidimensionale.*

*Il primo è noto come il "problema del bersaglio" e comprende l'esposizione e un esempio d'uso delle formule matematiche necessarie per realizzare un programma in cui c'entri in qualche modo il tiro al bersaglio, ovvero lo studio della traiettoria di un proiettile lanciato contro un certo bersaglio.*

*Tutti i programmi di giochi del tipo "Affondate la nave", "Colpite l'aereo", "Freccette", "Allunaggio", ecc. si basano sull'uso di queste formule.*

*Il secondo argomento, sicuramente meno applicabile ai giochi, è il problema del formato di uscita dei grafici bidimensionali, cioè come bisogna modificare i dati da visualizzare in modo che siano posizionati correttamente sull'output disponibile. Vedremo esempi su monitor, su stampante alfanumerica e su plotter.*

## Il problema del bersaglio

Uno degli argomenti più classici e sicuramente uno dei più divertenti nei testi di fisica, capitolo meccanica, dei licei è quello definito "problema del bersaglio", che si può formulare così:
lo studio del moto di un proiettile, dotato di una sua velocità iniziale VØ, lanciato da una posizione XØ, YØ, lungo una direzione che forma un angolo A (detto alzo) con l'orizzontale, verso un bersaglio individuato in una posizione XB, YB.

Il problema semplificato ha il bersaglio in posizione fissa ed ha come incognita l'alzo che bisogna dare al cannone che spara il proiettile per raggiungere il bersaglio stesso.

È in questo caso un problema bidimensionale.

Poiché nella formula risolutiva interviene l'accelerazione di gravità, occorrerà risolvere un sistema di secondo grado che, come al solito, presenta tre alternative a seconda che il discriminante △ della formula risolutiva sia > = < di zero:

1° due soluzioni reali; sono quindi due gli alzi che il proiettile può avere per raggiungere l'obiettivo.
2° due soluzioni reali e coincidenti; un solo alzo.
3° due soluzioni non reali; il proiettile non è in grado, con nessun alzo, di raggiungere l'obiettivo, poiché non ha una VØ sufficiente.

In fig. 1 vediamo lo sviluppo della soluzione e le formule risolutive da utilizzare nei programmi.

I casi più complessi sono tutti riconducibili al caso semplice ora esposto, solo che alcune costanti indicate potranno diventare anche esse variabili. Ad esempio il bersaglio sarà esso stesso in moto oppure il proiettile avrà un suo sistema di propulsione (non sarà semplicemente lanciato), il problema potrà diventare tridimensionale.

**Il problema del bersaglio**

| | |
|---|---|
| XØ,YØ | posizione iniziale del proiettile |
| VØ | velocità iniziale del proiettile |
| A | alzo - angolo rispetto all'orizzontale al momento del lancio |
| G | forza di gravità |

| | |
|---|---|
| VX = VØ*SIN(A) | componente della velocità lungo X |
| VY = VØ*COS(A) - G*T | componente della velocità lungo Y |
| SX = VX*T+XØ | spostamenti lungo X |
| VX = VØ*SIN(A) | componente della velocità lungo X |
| VY = VØ*COS(A) - G*T | componente della velocità lungo Y |
| SX = VX*T+XØ | spostamenti lungo X |
| SY = VY*T+YØ - G*(T^2)/2 | spostamenti lungo Y |

risolvendo rispetto a T, avremo che la formula che individua la traiettoria del proiettile è una parabola, ovvero una curva di 2° grado, del tipo
SY = C1*SX^2+ C2*SX+C3
dove C1, C2, C3 sono i termini noti, desunti dai dati iniziali.

*Figura 1 - Sviluppo analitico del problema del bersaglio*

Il problema del bersaglio ha avuto, come detto, applicazione in numerosi giochi, perché, aldilà della sua soluzione analitica, che come abbiamo visto è relativamente semplice, è un problema di immediata comprensione per tutti, o perlomeno per chiunque abbia mai lanciato qualcosa verso un bersaglio.

Inoltre la semplicità della formula risolutiva e la sua immediata applicabilità alle istruzioni di output del computer rendono il problema adattissimo ad un trattamento computerizzato, in special modo se con la tastiera o le paddles si simulano i comandi del cannone di lancio e del movimento del bersaglio e eventualmente del proiettile.

## Il programma "tiro al bersaglio"

Il programma è stato realizzato per illustrare tutto quanto detto precedentemente.

Bisogna indicare la posizione del bersaglio, costituito da un quadratino, mentre la posizione iniziale (cioè la posizione di lancio del proiettile) è l'origine del sistema di riferimento.

Il programma richiede dunque, con istruzione INPUT, velocità di lancio e alzo. Con questi dati elabora e visualizza la traiettoria, controllando via via la condizione di colpito, che si verifica quando la posizione del proiettile e quella del bersaglio coincidono. Se il bersaglio non viene colpito si può riprovare e così si possono confrontare le varie traiettorie in funzione del cambiamento effettuato sulla velocità di lancio e sull'alzo.

Il programma è stato realizzato con l'APPLE II, le cui caratteristiche grafiche evidenziano con buon dettaglio tutte le caratteristiche della traiettoria. Va da sé che il problema "analitico" del bersaglio è assolutamente indipendente dal computer usato.

Il computer quindi, una volta note la velocità iniziale e l'alzo del proiettile, calcola le due componenti lungo gli assi X e Y. Mentre la componente VX rimane fissa invece sulla VY interviene la forza di gravità; e quando questa prevale sulla velocità iniziale il segno della VY si inverte e quindi il proiettile comincia a cadere.

Ma a noi interessa seguire la posizione del proiettile e quindi realizziamo un LOOP in cui la variabile sia il tempo T e per ogni unità di tempo elementare, calcoliamo la posizione del proiettile XS, YS e la visualizziamo direttamente.

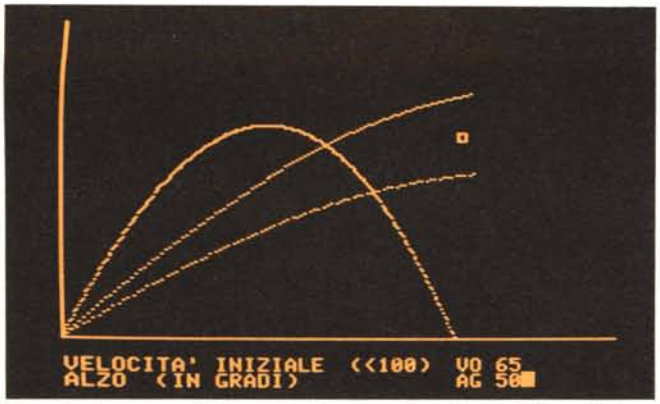


*Figura 2 - Programma del bersaglio + Output - aggiustando via via il tiro, alla fine ci riescono tutti.*

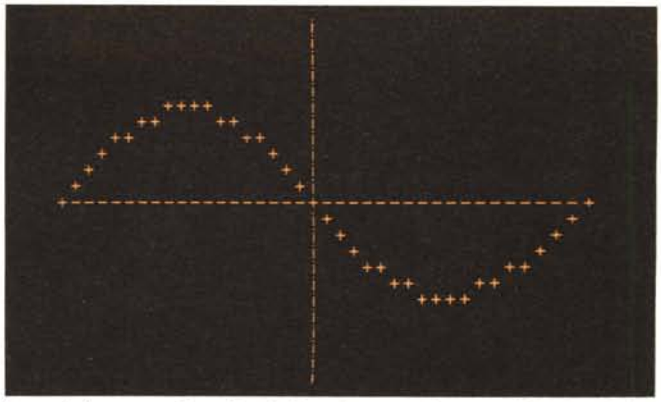

*Figura 4 - Programma Demo Y = Y(X) + Output - anche con una definizione minima si possono ottenere risultati gradevoli.*

Altre considerazioni spicciole. È stata introdotta una velocità iniziale massima in quanto tanto più questa è elevata tanto meno si sente l'influenza dell'accelerazione di gravità e quindi "viene male" la parabola.

Occorre poi fare i conti con il formato di uscita del microcomputer. Ad esempio per l'APPLE II occorre dimensionare opportunamente le grandezze e soprattutto occorre invertire tutti i valori di Y calcolati, sia del bersaglio sia del proiettile, in quanto la Y nel sistema di riferimento dell'APPLE va verso il basso mentre il nostro va verso l'alto.

## Il formato di uscita dei programmi grafici

Analizzeremo il secondo argomento, ovvero il problema del formato di uscita dei programmi di Grafica Bidimensionale, limitandoci al caso di rappresentazione di funzioni matematiche, in quanto questa problematica è già abbastanza vasta da "riempire" un articolo e le difficoltà che si incontrano affrontando i programmi sono le stesse in tutte le applicazioni.

L'analisi delle funzioni matematiche è una delle utilizzazioni più diffuse nei microcomputer, a tal punto che nel linguaggio BASIC è presente una apposita istruzione per definire la funzione DEF FNY (X), cui è correlata la istruzione di calcolo vera e propria Y = FNY (X). Ovvero, definita la funzione, ogni volta che nel corso del programma occorre calcolare il valore della Y per un determinato valore della X, bisognerà scrivere l'istruzione Y1 = FNY(X1).

Anche limitando l'argomento di Computer Grafica Bidimensionale all'analisi di funzioni univoche Y = Y(X), un programma generalizzato che permetta un output grafico su tutti i tipi di output possibili sarebbe estremamente complesso.

Abbiamo quindi realizzato tre programmi separati, il primo con uscita sul monitor alfanumerico, il secondo su stampante alfanumerica, il terzo su monitor grafico, e di quest'ultimo facciamo vedere anche una versione "plotterizzata". Toccheremo così tutte le problematiche connesse con l'argomento Computer Grafica Bidimensionale.

Per quanto riguarda il plotter, che è lo strumento più affascinante e potente in computer grafica, ma ancora poco diffuso a causa dei costi ancora alti, tra i possessori di microcomputer, ne tratteremo a lungo nel prossimo numero dedicando un intero articolo al suo uso.

## L'analisi della funzione Y = Y(X)

Supponiamo di avere una funzione univoca Y = Y(X), ovvero per ogni valore della X esiste un sol valore della Y.

L'esame della funzione dal punto di vista matematico ha caratteristiche differen-

```
100  HGR : HCOLOR= 3: HPLOT 0,0 TO 0,159 TO 279,159
110  HOME : VTAB (21)
120  INPUT "POSIZIONE BERSAGLIO       XB,YB ";XB,YB
130  VTAB (22): INPUT "VELOCITA' INIZIALE  (<100)  VO ";VO
140  VTAB (23): INPUT "ALZO  (IN GRADI)             AG ";AG
200  REM  INIZIALIZZAZIONI
210 P = 3.14159: REM  PI GRECO
220 G = 9.8: REM  ACCELLERAZIONE DI GRAVITA'
230 AR = AG * P / 180: REM   ANGOLO IN RADIANTI
240 VX = VO *  COS (AR):VY = VO *  SIN (AR)
300  REM  DISEGNO DEL BERSAGLIO
310 BX = XB - 2:BY = 157 - YB
320  HPLOT BX,BY TO BX + 4,BY TO BX + 4,BY + 4
330  HPLOT  TO BX,BY + 4 TO BX,BY
400  REM  LOOP PRINCIPALE
430  FOR T = 0 TO 10 STEP .04
440  GOSUB 500: GOSUB 600
450 SX = XS:SY = 159 - YS
460  HPLOT SX,SY
470  IF XS > XB + 5 THEN 800
480  IF SY > 159 THEN 800
490  NEXT T
500  REM  CALCOLO POSIZIONE AL TEMPO T
510 XS = VX * T
520 YS = VY * T - G * (T ^ 2) / 2
530  RETURN
600  REM  CONTROLLO SE COLPITO
610  IF XS > XB - 2 AND XS < XB + 2 AND YS > YB - 2 AND YS <
     YB + 2 THEN 700
620  RETURN
700  REM  COLPITO !!!!
710  FOR I = 1 TO 5: PRINT  CHR$ (7): NEXT I
720  HOME : FLASH : VTAB (22): PRINT "  COLPITO !!!  "
730  NORMAL : FOR I = 1 TO 999: NEXT I
740  END
800  REM  MANCATO
810  HOME : FLASH : VTAB (22): PRINT "  MANCATO !!!  "
820  NORMAL : FOR I = 1 TO 999: NEXT I
830  HOME : VTAB (22): PRINT " VUOI RIPROVARE  Y/N  ";
840  GET A$: IF A$ = "Y" THEN 860
850  END
860  PRINT : HOME : GOTO 130
```

*Figura 3 - Programma del bersaglio + listing - una implementazione molto semplice consiste nel far muovere anche il bersaglio, legandolo al valore T del loop principale.*

```
100  REM  FUNZIONE
110  DEF  FN Y(X) =  SIN (X)
120 P = 3.14156: REM   PIGRECO
130 ST = P / 20: REM  PASSO CALCOLO
140 S = 6: REM   SCALA
150 TX = 20: REM  TRASLAZIONE X
160 TY = 12: REM  TRASLAZIONE Y
170  GOSUB 400
200  REM  CALCOLO
210  FOR X1 =  - P TO P STEP ST
220 Y1 =  FN Y(X1)
250 X2 = X1 * S + TX
260 Y2 = Y1 * S + TY
270  REM  ARROTONDAMENTO
280 X2 =  INT (X2 + .5)
290 Y2 =  INT (Y2 + .5)
300  REM  VISUALIZZAZIONE
310  HTAB (X2): VTAB (Y2)
320  PRINT "+"
330  NEXT : FOR I = 0 TO 2999: NEXT
340  VTAB (23): END
400  REM  TRACCIAMENTO ASSI
410  HOME
420  FOR I = 1 TO 39: VTAB (12): HTAB (I)
430  PRINT "-";: NEXT
440  FOR I = 1 TO 23: HTAB (20): VTAB (I)
450  PRINT "!";: NEXT
460  RETURN
```

*Figura 5 - Programma Demo Y = Y(X) + Listing - le funzioni di trasformazione descritte nel testo sono nelle righe 250 e 260.*

```
DEF VERT PRINTER   = 40
DEF ORIZZ PRINTER  = 54
YMIN   = -1
YMAX   = 3
Y2-Y1  = 4
STEP Y = .1
XMIN   = -.1477
XMAX   = .8
X2-X1  = .9477
SCALA X =   ACCENDI LA STAMPANTE
PREMI SPACE PER CONTINUARE
```

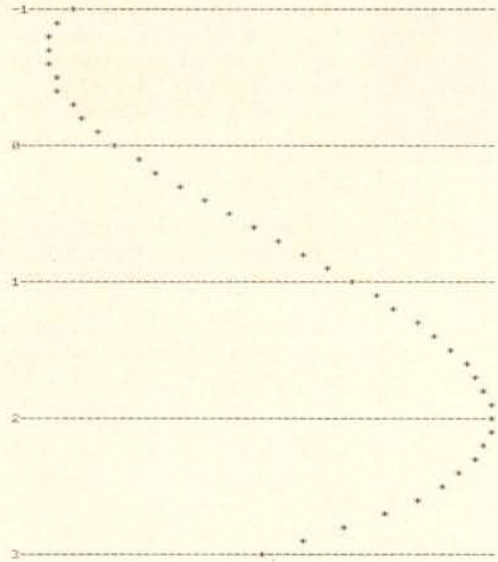


*Figura 6 - Programma Printer + Output - in alto sono indicati i valori (che appaiono sul monitor) della "finestra" entro la quale vediamo il grafico.*

```
10  HOME
20  REM  INIZIALIZZAZIONI
30  INPUT "  DEF VERT PRINTER   = ";D1
40  INPUT "  DEF ORIZZ PRINTER  = ";D2
100  PRINT : PRINT
110  REM  DEFINIZIONE FUNZIONE
120  DEF  FN X(Y) =  - .1 * Y ^ 3 + .2 * Y ^ 2 + .4 * Y
140  INPUT "  YMIN   = ";Y1
150  INPUT "  YMAX   = ";Y2
160 DY = Y2 - Y1:SY = DY / D1
170  PRINT : PRINT "  Y2-Y1  = ";DY
180  PRINT : PRINT "  STEP Y = ";SY
190  PRINT
200  REM    DEFINIZIONE INTERVALLO XMIN,XMAX
210 X1 = 999:X2 =  - 999
220  FOR Y = Y1 TO Y2 STEP SY
230 X =  FN X(Y)
240  IF X < X1 THEN X1 = X
250  IF X > X2 THEN X2 = X
270  NEXT Y
280  PRINT "  XMIN   = ";X1
290  PRINT "  XMAX   = ";X2
300 DX = X2 - X1
310  PRINT : PRINT "  X2-X1  = ";DX
400  REM   SCELTA DELLA SCALA DI TRASFORMAZIONE
410 S = D2 / DX
420  PRINT : PRINT " SCALA X = ";S
500  REM  ACCENSIONE PRINTER
510  PRINT : PRINT
520  PRINT "  ACCENDI LA STAMPANTE  "
530  PRINT "  PREMI SPACE PER CONTINUARE ";: GET A$
540  PRINT : PRINT : HOME
600  REM  CALCOLO FINALE
610  FOR Y3 = Y1 TO Y2 STEP SY * 10
620  FOR Y4 = 0 TO 9
630 Y = Y3 + Y4 * SY
635  IF Y > Y2 THEN  END
640 X =  FN X(Y)
650 XF =  INT ((X - X1) * S + 1.5)
660  IF Y4 = 0 THEN  GOSUB 700
670  IF Y4 = 0 THEN  GOTO 690
680  PRINT  TAB( XF + 4);"*"
690  NEXT Y4,Y3: END
700  REM  TRACCIAMENTO LINEE ORIZZONTALI
710 Y$ =  LEFT$ ( STR$ (Y) + "----",4)
720  PRINT Y$;
730  FOR J = 1 TO XF - 1: PRINT "-";: NEXT
740  PRINT "*";
750  IF XF + 1 >  = D2 THEN 770
760  FOR J = XF TO D2: PRINT "-";: NEXT
770  PRINT : RETURN
```

*Figura 7 - Programma Printer + Listing - il programma "gira" con qualsiasi funzione univoca X = X(Y), nel caso in esame elaboriamo una parabola del terzo ordine.*

ti dall'esame visivo, ovvero tramite OUTPUT grafico, della funzione stessa.

Nel primo caso, tramite i procedimenti suggeriti dall'analisi matematica (vedi libri del liceo), si ottengono facilmente tutte le caratteristiche della curva (punti di zero, punti singolari ovvero punti di massimo e di minimo, flessi, ecc.) e i risultati che si ottengono sono le curve coppie di valori che assumono X e Y in corrispondenza di questi punti. Si può inoltre facilmente seguire passo passo l'andamento della funzione calcolando la Y per dati incrementi della X. Ad esempio:

```
10  DEF  FN Y(X) =  SIN (X)
20  FOR X = X1 TO X2:Y =  FN Y(X)
30  PRINT X,Y: NEXT
```

Queste semplici istruzioni permettono di individuare e in questo caso di stampare le varie coppie di valori che definiscono la funzione nell'intervallo stesso scelto.

Nel secondo caso, ovvero nel caso di esame tramite OUTPUT grafico della funzione Y=Y(X), il procedimento di analisi è sostanzialmente simile, solo che le coppie di valori via via calcolate vanno trasformate in modo che siano visualizzabili sull'OUTPUT prescelto. Occorre quindi conoscere esattamente, nel momento in cui si predispone il programma, le caratteristiche del supporto OUTPUT che si vuole usare.

## L'OUTPUT della funzione Y=Y(X)

Per comprendere concretamente i problemi che si incontrano nei programmi che visualizzano funzioni o disegni bidimensionali faremo riferimento a tre tipi di output differenti:

- output su STAMPANTE alfanumerica, 132 caratteri a 10 caratt./pollice in orizzontale, 66 caratteri a 6 caratt./pollice in verticale, ovvero 8712 caratteri su una pagina di 907,5 cmq, (9,6 car./cmq).
- output su monitor APPLE II 280 punti in orizzontale, 192 punti in verticale, ovvero 53760 punti su una superficie di circa 400 cmq (monitor da 12") pari a 135 punti/cmq.
- output su plotter WATANABE DIGIPLOT (3600 punti in orizzontale 2400 punti in verticale, ovvero 8,64 milioni di punti su una superficie di 864 cmq (10000 punti/cmq.).

Già da questi dati puramente numerici si comprende come il programma di visualizzazione di una funzione, come qualunque programma grafico sia pesantemente condizionato dal tipo di output prescelto.

Il problema di uniformare il formato di uscita a quello reale di calcolo può essere facilmente risolto trasformando i dati di calcolo in dati visualizzabili tramite un fattore di scala da determinare con una opportuna subroutine.

Altri elementi che vanno considerati nella stesura del programma sono la posizione e l'orientamento del sistema di riferimento, che devono essere concordi rispetto a quelli reali. In pratica anche questo problema si risolve facilmente traslando e cambiando (eventualmente) segno alle coordinate calcolate nel caso reale.

Altro problema è quello di rappresentare in uscita un sistema di riferimento corretto, ovvero che permetta una precisa valutazione dei dati interessanti, come i punti singolari della curva.

Vi sono poi i vincoli di programmazione rappresentati dalle caratteristiche tecniche dei vari tipi di unità output. Ad esempio la stampante permette solo avanzamenti lungo la X e lungo la Y della testa scrivente, non permette cioè di tornare indietro.

Per superare questa difficoltà ci sono due soluzioni. La prima consiste nel ruotare il grafico di 90 gradi in modo che il loop crescente della X si trasferisca sulla Y, e quindi diventi concorde con il verso di trascinamento della carta.

La seconda, che non fa uso di questi 'mezzucci', consiste nel fare in modo di sapere per ogni riga che si stampa quanti valori Y ci sono e in quali posizioni X si trovino. Questo si può ottenere caricando una matrice rettangolare con tutti i valori X e Y precalcolati e successivamente di analizzarla riga per riga e di stamparla.

Una ultima considerazione prima di esaminare qualche esempio pratico.

È inutile, quando si vuol realizzzare un

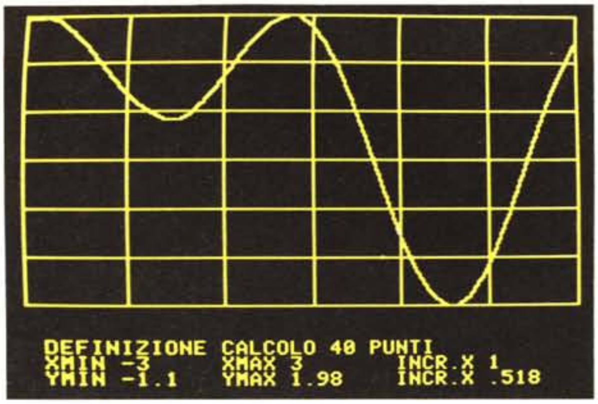


*Figura 8 - Programma Monitor + Output - la quadrettatura è 6 per 6, la sua taratura è indicata sotto al grafico. Non è difficile comunque inserirla, in forma grafica, nel disegno.*

*Figura 9 - Programma Monitor + Listing - una parte più consistente del programma serve per il calcolo e la scelta della scala di trasformazione.*

grafico, spingere la definizione del calcolo oltre la definizione permessa dall'output.

In particolare converrà usare sempre per le coppie X,Y valori interi, non essendo possibile individuare, neanche sul plotter, valori frazionari di X e Y.

## Le funzioni di trasformazione

In definitiva, per trasformare i valori reali desunti dal calcolo in valori output dei punti da visualizzare occorre utilizzare formule del tipo:

X2 = X1 * S + TX
Y2 = Y1 * S + TY

in caso di cambio scala e traslazione semplice,

X2 = (X1 * COS(A) + Y1 * SIN(A) * S + TX)
Y2 = (-X1 * SIN(A) + Y1 * COS(A)) * S + TY

quando, per meglio visualizzare la funzione occorresse anche ruotare i valori output rispetto ai valori reali di un angolo A.

Trascurando questo ultimo caso, vediamo ora come calcolare i valori S, TX, TY, che ci permettono di realizzare la trasformazione, utilizzando pienamente il formato del supporto output.

Per cercare analiticamente tali valori, vanno calcolate dapprima tutte le coppie di valori reali e tra questi le due coppie XMAX e XMIN e YMAX e YMIN e poi le due scale ammissibili SX e SY come rapporto tra l'intervallo massimo calcolato nella funzione tra le X e le Y e l'intervallo massimo disponibile in output. Ad

Figura 9

esempio se XMIN è Ø, XMAX è 48 e l'output ci permette una definizione sulla X pari a 132, la SX sarà 132/48.

Tra i due valori determinati di scala si sceglie il più basso, in quanto evidentemente la scala di trasformazione deve essere unica nelle due direzioni X e Y.

I valori di traslazione vanno invece calcolati in modo tale che i puni reali individuati dalla coppia di valori X, YMIN X, YMIN e XMIN, Y corrispondono rispettivamente al minimo valore della Y e al minimo valore della X in output.

## Un programma esemplificativo

Vogliamo visualizzare sul monitor dell'APPLE II, utilizzato questa volta in modo alfanumerico quindi non grafico, la funzione Y=SIN(X), nell'intervallo che va da — PIGRECO a + PIGRECO.

La definizione del monitor dell'APPLE è di 24 righe per 40 colonne, e noi scegliamo gli assi in modo che l'asse X corrisponda alla 12°-ma riga e l'asse Y alla 20-ma colonna.

Il listing del programma e l'output relativo sono rispettivamente in fig. 4 e in fig. 5. I dati di scala S, di traslazione TX e TY sono fissati nel programma. Con i programmi successivi illustreremo la subroutine per il calcolo diretto di tali valori.

## L'uscita su printer

Il programma listato in fig. 7 e i cui output sono in fig. 6 rappresenta il caso più noto di uso della printer per tracciare grafici.

Fissata la funzione da esaminare (riga 120) occorre impostare la definizione da dare al grafico in uscita (righe 30 e 40), sia in verticale che in orizzontale e poi occorre impostare l'intervallo Y1, Y2 entro il quale esaminare la funzione (righe 140 e 150). Il programma calcola lo step del loop (riga 160) ed eseguendolo una prima volta calcola tutte le X corrispondenti. Per determinare XMIN e XMAX (righe 220-300). A questo punto calcola la scala sulla X, ovvero il valore per il quale moltiplicare tutti i valori della X perché occupino il range definito per la X (righe 400-420). A questo punto non occorre altro che stampare.

Il loop principale viene diviso in due in modo da tracciare ogni 10 righe anche delle linee orizzontali quotate di riferimento (righe 700-770). Il calcolo della posizione del singolo punto del grafico è in riga 650.

Per chi non ha la stampante niente paura. Il programma, se vi accontentate della definizione del monitor, funziona anche sul video.

## L'uscita su monitor

Il programma listato in fig. 9, chiede innanzitutto su quanti punti eseguire il calcolo per graficizzare la funzione e l'intervallo X1-X2 di calcolo (riga 20). Dopo di che calcola tutti i valori di X ed esegue il calcolo delle scale SX e SY, scegliendo poi quella più bassa, ovvero quella con la quale la dimensione maggiore viene comunque ridotta al formato dello schermo (righe 350-380) (righe 200-310). Poi calcola i valori di traslazione per centrare il tutto. Infine (righe 350-380) carica una matrice di interi con i valori delle coppie di coordinate dei punti da unire per tracciare il grafico (righe 500-540).

In più c'è il tracciamento di una quadrettatura (righe 420-480) le cui caratteristiche sono specificate in fondo al grafico (righe 600-650). Il programma è continuo, ovvero alla fine chiede nuovi dati iniziali per ricominciare.

*Francesco Petroni*

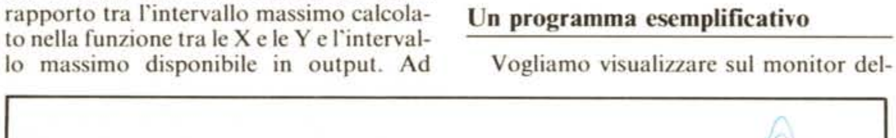

*Figura 10 - Programma Plotter - dello stesso programma di prima, inserendo i parametri in un loop, realizziamo una uscita su plotter.*

Insomma, tra clienti e fornitori, registri e adempimenti di legge, finiva che non avevo neanche più il tempo di rispondere al telefono o di battere una relazione in santa pace.

Così sono andata dal capo e gli ho messo un aut-aut: "O mi prendete un'aiuto, oppure è uno sfascio," ho detto.

E dopo un po' di giorni viene qui il Concessionario Harden Commodore e mi dice: "Mi parli dei suoi problemi." Finalmente: lui e il capo hanno confabulato un po', poi è arrivato questo gioiello, il Sistema Commodore PET Serie 3001.

Mi ha insegnato ad usarlo, ha fatto i programmi e mi ha detto: "Qualunque cosa abbia bisogno, un colpo di telefono e siamo lì in un lampo."

In una settimana siamo partiti.

HARDEN
commodore
n° 1 in Microcomputer.

# "Certo, anche adesso devo fare tutto io: primanota, pagamenti, banche, bilanci, e in più bolle di consegna, carico e scarico del magazzino, fatture. Ma da quando abbiamo "lui", faccio in un lampo."


*Configurazione base*
*COMMODORE PET 3032+*
*Floppy disc 3040+Stamp. L/20*

Job Line

**HARDEN S.p.A. direzione commerciale 26048 SOSPIRO (CR) Tel. 0372/63136 Telex 320588 I**

PIEMONTE E VAL D'AOSTA: Tel. 011/389328 332065 • LOMBARDIA: Tel. 02/4695467 • VENETO: Tel. 0444/563864 • FRIULI V. GIULIA: Tel. 040/793211 • UDINE: Tel. 0432/291466 • TRENTINO A.A.: Tel. 0471/24156 • LIGURIA: Tel. 0185/301032 • EMILIA ROMAGNA: Tel. 0544/30258 30081 • TOSCANA: Tel. 055/663696 • MARCHE: Tel. 071/9170564 • UMBRIA: Tel. 0761/224688 • LAZIO: Tel. 06/5916438 • ABRUZZI: Tel. 085/50883 • CAMPANIA: Tel. 0824/24168 21680 • PUGLIE E BASILICATA: Tel. 0881/76111 080/481327 • CALABRIA: Tel. 0984/71392 • SICILIA: Tel. 090/2928269 • SARDEGNA: Tel. 070/663746