

ALCUNI ALGORITMI PER LA COMPUTER GRAFICA DISEGNARE ENTRO I MARGINI

Nel numero scorso abbiamo trattato il test di appartenenza di un punto ad un poligono, cioè abbiamo trovato un algoritmo per verificare se un punto è interno o esterno rispetto ad un poligono comunque dato.

Questo test è stato applicato ad un programma per l'individuazione della posizione di un menu rettangolare comunque posizionato sulla tavoletta grafica di MCmicrocomputer.

Estendiamo il problema al caso della visualizzazione di dati grafici comunque forniti e comunque dimensionati su uno schermo di uscita comunque dimensionato.

È il caso che si presenta nella trattazione degli archivi di dati grafici che hanno una propria organizzazione e sono individuati in un proprio sistema di riferimento del tutto autonomo dalle modalità di visualizzazione.

Il problema detto in parole semplici è questo: abbiamo un disegno qualsiasi con proprie caratteristiche archiviato nel computer, o, se si tratta di funzioni, elaborato dal computer, e lo vogliamo vedere su un output qualsiasi, ad esempio il monitor grafico, che ha a sua volta proprie caratteristiche.

Per comodità chiameremo l'output scelto finestra.

Esistono tre possibilità. La prima è che tutto il disegno entri perfettamente entro i margini della finestra; la seconda è che non entri per niente; l'ultima è che entri parzialmente.

Ci sono due modalità di approccio al

problema. La prima consiste nel modificare tutti i dati del disegno in modo da renderli compatibili con il formato del sistema output scelto. Questa operazione si chiama SCALING, e richiede la preventiva elaborazione di tutti i dati allo scopo di trovare per essi i parametri necessari alla traduzione, e solo con tali parametri può avvenire la visualizzazione.

Abbiamo già trattato per varie applicazioni questa problematica e rimandiamo i lettori interessati agli articoli sui numeri precedenti di MC.

La seconda modalità di approccio è quella di visualizzare solo quello che si vede attraverso la finestra, ignorando le porzioni del disegno esterne.

Questa problematica è propria dei programmi di output di archivi grafici, dove è l'operatore in pratica che sposta la finestra sull'archivio dati completo.

Facciamo un esempio "cartografico". Supponiamo di avere memorizzato l'intera pianta del comune di Roma e di voler visualizzare sul video la zona di P.za Venezia ad una certa scala di uscita. Sul video apparirà al centro P.za Venezia e intorno un certo numero di strade, piazze, palazzi in funzione della scala scelta. Ci saranno strade, piazze e palazzi "tagliati in due" dal bordo del monitor. L'operatore può spostare la finestra e/o cambiare scala. In ogni caso ci saranno sempre elementi del disegno tagliati in due dai margini della finestra.

Nell'articolo vogliamo trattare questa problematica da un punto di vista analitico, precisando subito che le routine sono

alquanto complicate e rallentano in maniera a volte insostenibile i programmi di output.

I programmi sono scritti in linguaggio Applesoft, Basic che è implementato con buone funzioni grafiche.

Molti microcomputer più recenti dell'Apple II, hanno software grafici più potenti ed hanno funzioni di test dentro/fuori dalla finestra implementate a livello software di base e quindi il tentativo di visualizzare un elemento esterno alla finestra non produce un errore ma produce semplicemente il non disegno dell'elemento.

Ciononostante riteniamo che una trattazione analitica possa mettere a fuoco gli aspetti più interessanti del problema.

Come al solito i programmi sono dimostrativi, ovvero tendono solo ad esemplificare la trattazione teorica e non hanno alcuna utilità pratica. Potranno essere inseriti come subroutine in procedure grafiche complesse laddove occorra risolvere problemi di formati di visualizzazione.

I programmi che presentiamo sono quattro e li abbiamo intitolati Punti, Cerchi, Linee e Segmenti.

Infatti il problema di fare il test dentro/fuori dalla finestra di un disegno, si riduce a verificare che i suoi singoli elementi siano dentro o fuori. E i singoli elementi di un disegno sono punti, segmenti, linee.

Il programma punti verifica che punti generati casualmente cadano dentro o fuori una finestra sul monitor Apple II.

Il programma cerchi prende come finestra l'intero monitor grafico. Vengono ge-

```

100 HGR2 : HCOLOR= 3
110 REM QUADRANTE ESTERNO
120 SX = 10 : DX = 270 : RX = 10 : BX = 180
130 X% = SX : Y% = AX : GOSUB 280 : X% = DX : GOSUB 290 : Y% = BX
140 GOSUB 290 : X% = SX : GOSUB 290 : Y% = AX : GOSUB 290
150 REM QUADRANTE INTERNO
160 SX = 60 : DX = 210 : RX = 50 : BX = 130
170 X% = SX : Y% = AX : GOSUB 280 : X% = DX : GOSUB 290 : Y% = BX
180 GOSUB 290 : X% = SX : GOSUB 290 : Y% = AX : GOSUB 290
190 REM PUNTO RANDOM
200 M% = RND (1) * 270 + 10 : N% = RND (1) * 170 + 10
210 X% = M% : Y% = N% : GOSUB 280
220 REM TEST DENTRO/FUORI
230 IF M% > SX AND M% < DX AND N% > RX AND N% < BX THEN 250
240 GOTO 270
250 X% = M% - 2 : Y% = N% : GOSUB 280 : X% = M% + 2 : GOSUB 290
260 X% = M% : Y% = N% - 2 : GOSUB 280 : Y% = N% + 2 : GOSUB 290
270 GOTO 200
280 HPLLOT X% : Y% : RETURN : REM FUNZIONE PLOT X, Y
290 HPLLOT TO X% : Y% : RETURN : REM FUNZIONE PLOT TO X, Y
    
```

Figura 1 - Programma Punti - Listato. Il punto è generato in maniera casuale dalla funzione RND. Se risulta esterno si vede un puntino, se risulta interno si vede una crocetta.

```

100 HOME : VTAB (20) : PRINT "ATTENDERE"
110 PRINT "CARICAMENTO TABELLA SENI/COSENI"
120 ONERR GOTO 250
130 N = 100 : M = N / 2 : P = 3.141593 : PP = P / M
140 DIM C(N), S(N) : FOR I = 0 TO N
150 C(I) = COS (I * PP) : S(I) = SIN (I * PP) : NEXT I
160 HGR2 : HCOLOR= 3
170 HPLLOT 0, 0 TO 279, 0 TO 279, 191 TO 0, 191 TO 0, 0
180 FOR L = 1 TO 20 : R% = RND (1) * 50 + 50
190 C = RND (1) * 190 + 40 : D = RND (1) * 130 + 30
200 HPLLOT C - 2, D TO C + 2, D : HPLLOT C, D - 2 TO C, D + 2
210 X = C + R% : Y = D : HPLLOT X, Y
220 FOR I = 0 TO N
230 X = C + R% * C(I) : Y = D + R% * S(I)
240 HPLLOT TO X, Y : NEXT I, L : END
250 REM ROUTINE ERRORE
260 IF PEEK (222) < > 53 THEN 320
270 IF X < 0 THEN X = 0
280 IF Y < 0 THEN Y = 0
290 IF X > 279 THEN X = 279
300 IF Y > 191 THEN Y = 191
310 RESUME
320 TEXT : PRINT "ERRORE " : PEEK (222) : END
    
```

Figura 3 - Programma Cerchi - Listato. Il test dentro/fuori è realizzato con la routine di errore. L'errore 53 è l'ILLEGAL QUANTITY che si verifica quando si vuol tracciare un punto esterno al formato del monitor.

nerati cerchi casuali e non viene tracciata la porzione di circonferenza che risulta esterna alla finestra.

Con il programma linee viene esteso il test eseguito per i punti a tutti i punti di una linea. Quindi sul monitor vengono tratteggiate le parti esterne e tracciate a tratto continuo le parti interne del segmento.

Infine il programma segmenti esegue per il segmento casuale un complesso calcolo di intersezioni per individuare se e quale sua parte è interna alla finestra.

Programma Punti

Il primo programma è listato in figura 1, l'output è in figura 2.

Il quadrante esterno, disegnato nelle righe 100/140, rappresenta lo spazio entro il quale può essere generato un punto. Nei programmi per i quali la finestra è rappresentata dal monitor grafico, lo spazio del disegno può essere qualsiasi.

Il quadrante interno, disegnato nelle righe 150/180, rappresenta il margine entro il quale i punti sono considerati entro la finestra.

Nelle righe 190/210 vengono calcolati punti random e tracciati sullo schermo con un pixel. Nelle righe 220/230 viene eseguito il test dentro/fuori, se il punto è esterno non succede nulla e ne viene calcolato un altro, se il punto è interno viene evidenziato con una crocetta.

Avrete notato che nel programma sono isolate le funzioni di plot. Questo viene fatto per rendere il programma facilmente traducibile sia per altri computer, che abbiano differente sintassi degli statement grafici, sia per uscite su plotter, che hanno anche esse ovviamente istruzioni differenti.

Programma Cerchi

Il programma è listato in figura 3 e l'output in figura 4. Anche questo è scritto per l'Apple II ed in particolare usa la sua sintassi di errore.

Cioè il programma calcola in modo random alcune circonferenze e le traccia sul monitor. Se la circonferenza esce dal formato del monitor si genera un errore ILLEGAL QUANTITY, che è l'errore 53. In questo caso con una apposita routine viene corretto il valore del punto da tracciare e il programma prosegue.

Con questo sistema non si deve testare ciascun elemento del disegno, con grande vantaggio sulla velocità di esecuzione. Il programma inizia con il caricamento della tabella di seni e coseni (righe 100/150). Ricordiamo che le funzioni SEN (X) e COS (X) sono lente e quindi quando è possibile è meglio costruire una tabella con un certo numero di loro valori, e usare la tabella. È infatti molto più rapido il tempo di accesso ad una tabella che non il calcolo di una funzione trigonometrica.

Viene poi accesa la pagina grafica e disegnata la cornice esterna che rappresenta la

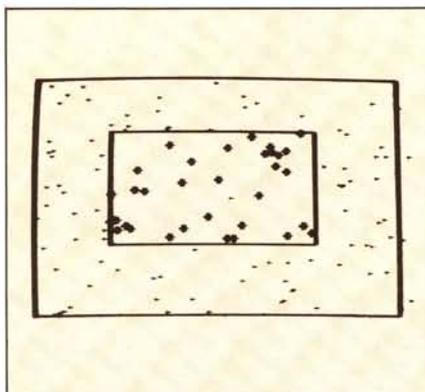


Figura 2 - Programma Punti - Output. La cornice esterna rappresenta lo spazio dei punti possibili. La nostra finestra è rappresentata dalla cornice interna.

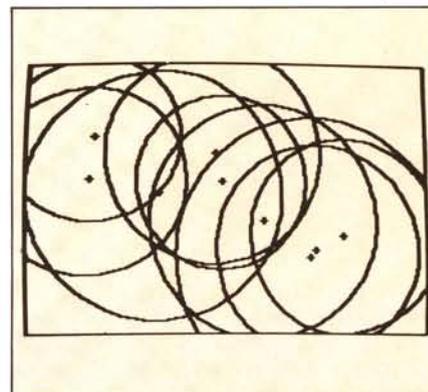


Figura 4 - Programma Cerchi - Output. Il programma ha una piccola attesa iniziale necessaria per intabellare i valori dei seni e dei coseni.

```

100 HGR : HCOLOR= 3: HOME
110 REM QUADRANTE ESTERNO
120 S = 10: D = 270: A = 1: B = 159: T = .5
130 X% = S: Y% = A: GOSUB 390: X% = D: GOSUB 400: Y% = B
140 GOSUB 400: X% = S: GOSUB 400: Y% = A: GOSUB 400
150 REM QUADRANTE INTERNO
160 S = 60: D = 210: A = 50: B = 110
170 X% = S: Y% = A: GOSUB 390: X% = D: GOSUB 400: Y% = B
180 GOSUB 400: X% = S: GOSUB 400: Y% = A: GOSUB 400
190 X1% = RND (1) * 260 + 10: Y1% = RND (1) * 149 + 10
200 X2% = RND (1) * 260 + 10: Y2% = RND (1) * 149 + 10
210 DX = X2% - X1%: DY = Y2% - Y1%: R = DY / DX
220 F = 0: IF ABS (R) > 1 THEN F = 1: R = 1 / R
230 HOME : VTAB (21): PRINT " X1 = "; X1%: TAB (12): "Y1 = "; Y1%
240 PRINT " X2 = "; X2%: TAB (12): "Y2 = "; Y2%
250 PRINT " DX = "; DX: TAB (12): "DY = "; DY: TAB (22): "F = "; F
260 X% = X1%: Y% = Y1%: GOSUB 390: X% = X2%: Y% = Y2%: GOSUB 390
270 IF F THEN 310
280 FOR X = X1% TO X2% STEP SGN (DX)
290 Y = Y1% + (X - X1%) * R: X% = X + T: Y% = Y + T
300 GOSUB 340: NEXT X: GOTO 190
310 FOR Y = Y1% TO Y2% STEP SGN (DY)
320 X = X1% + (Y - Y1%) * R: X% = X + T: Y% = Y + T
330 GOSUB 340: NEXT Y: GOTO 190
340 REM TEST DENTRO FUORI
350 IF X% > S AND X% < D AND Y% > A AND Y% < B THEN GOSUB 390
: GOTO 380
360 IF F AND Y / 3 = INT (Y / 3) THEN GOSUB 390: GOTO 380
370 IF X / 3 = INT (X / 3) THEN GOSUB 390
380 RETURN
390 HPLLOT X%, Y%: RETURN
400 HPLLOT TO X%, Y%: RETURN

```

Figura 5 - Programma Linee - Listato. Il tratteggio è realizzato tracciando un punto ogni tre nel caso che il segmento o la porzione di segmento sia esterno alla finestra.

nostra finestra (righe 160/170). In righe 180/190, vengono calcolati in modo random, centro, il punto di coordinate C e D, e il raggio della circonferenza R. Il centro, calcolato in modo che sia sempre interno alla finestra, viene visualizzato con una crocetta (riga 200) infine c'è il disegno vero e proprio della circonferenza realizzato tramite una spezzata di 100 segmentini (righe 210/240).

La routine di correzione dell'errore vie-

ne richiamata ogni volta che il segmentino da tracciare cade fuori della finestra. Se l'errore non è quello che ci aspettiamo, cioè l'illegal quantity in 240, il programma segnala l'errore e finisce. Se invece è proprio l'errore 53, viene corretto il valore delle coordinate del punto P (X, Y) illegale, in un valore pari alle coordinate dei margini (righe 250/300).

L'istruzione RESUME (di riga 310), ricordiamo, fa riprendere l'esecuzione del

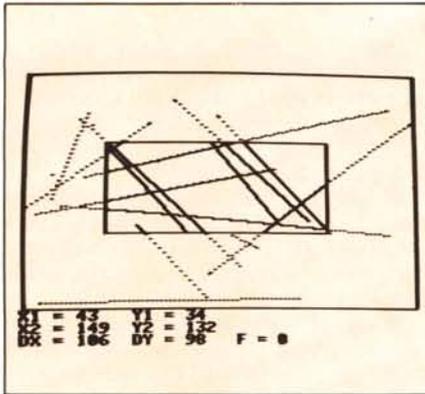


Figura 6 - Programma Linee - Output. Il disegno di una linea tracciata punto per punto è molto lento e si rallenta ancora di più quando ogni punto subisce il test per il tracciato.

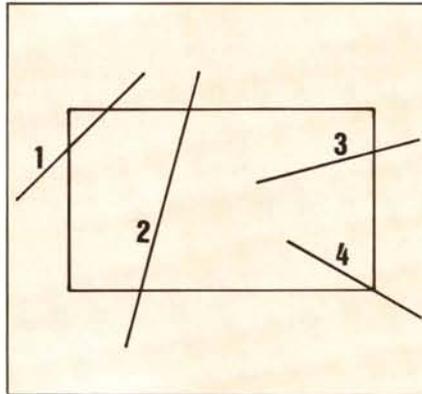


Figura 7 - Programma Segmenti - Casistica. La casistica delle posizioni relative che assume un segmento rispetto ad una finestra è molto complessa e presenta anche casi singolari.

programma a partire dall'istruzione che ha causato l'errore. In questo modo in pratica la porzione esterna della circonferenza viene schiacciata lungo il margine della finestra.

L'unico inconveniente a questo sistema si verifica quando si aumenta troppo il passo del tracciamento. Infatti se un segmento cade a cavallo del margine, non se ne calcola l'intersezione ma si modifica, in un modo un pò brutale, il valore delle coordinate del suo estremo esterno alla finestra e questo ovviamente non è corretto dal punto di vista geometrico.

Programma Linee

Listato in figura 5, output in figura 6.

Mentre non abbiamo avuto difficoltà a verificare se un punto è esterno o interno rispetto ad una finestra, il discorso si complica nel caso di segmenti. Infatti un segmento, e lo vedremo più avanti, può assumere varie posizioni su un piano sul quale sia posizionata una finestra.

Un metodo per risolvere il problema è quello di disegnare il segmento per punti testando ogni singolo punto.

Questo procedimento può essere utilizzato solo su un output di limitata definizione, come i monitor. Non è utilizzabile ad esempio su plotter. Pensate quanto ci vorrebbe a tracciare un segmento lungo 20 cm con un plotter che individua il decimo di millimetro.

Il programma linee quindi può essere usato solo con uscite su monitor.

Vengono, come per il programma punti, visualizzati lo spazio entro il quale tracciamo segmenti e la finestra attraverso la quale vederli (righe 100/180).

Viene individuato un segmento, calcolandone in modo random i due punti estremi (righe 190/220), e poi viene eseguito il calcolo dell'inclinazione del segmento (righe 210/220) per decidere se tracciare i suoi punti eseguendo un loop sulla X (righe 280/300) o sulla Y (righe 310/340).

Alcuni dei valori del calcolo vengono visualizzati sulle quattro righe di testo sulla pag. HGR. Ricordiamo che se il segmento è molto inclinato rispetto all'orizzontale (più di 45°) conviene eseguire il loop sulla Y in quanto altrimenti per piccoli incrementi di X avremmo grandi incrementi di Y e quindi i punti del segmento verrebbero sgranati.

La routine di individuazione geometrica di un punto P(X,Y) che cade sul segmento individuato da due punti è riportata nella riga 290, se è noto X, o riga 320, se è noto Y. Il loop ha come step il SGN (DX) o il SGN (DY) per coprire anche il caso di loop alla rovescio, cioè di punto iniziale con coordinate più grandi del punto finale.

Il test dentro/fuori (righe 340/380) è sostenuto da tutti i punti. Se è superato, il punto viene tracciato, se non è superato, viene tracciato solo un punto su tre in modo da realizzare una linea punteggiata. Vengono disegnati solo quei pixel la cui coordinata X o Y sia divisibile per tre.

```

100 REM INIZIALIZZAZIONI
110 DIM XX(6),YY(6): REM VETTORI PUNTI ORIGINE E INTERSEZ
120 X1X = 1: X2X = 200: Y1Y = 0: Y2Y = 150: REM MARGINI QUADR. ESTERNO
130 SX = 50: DX = 150: AX = 50: BX = 100: REM MARGINI QUADR. INTERNO
140 T = 3: F = 0: F1 = 0: F2 = 0: REM INIZIALIZZ. VARIABILI
150 TEXT : HGR : HCOLOR = 3: GOSUB 660: GOSUB 690
160 REM IMMISSIONE ESTREMI SEGMENTO
170 HOME : VTAB (21)
180 INPUT " X1, Y1 " : XX(1), YY(1): INPUT " X2, Y2 " : XX(2), YY(2)
190 XX = XX(1): YY = YY(1): GOSUB 720
200 XX = XX(2): YY = YY(2): GOSUB 730
210 REM CALCOLO E STAMPA DEI COEFFICIENTI DELLA RETTA
220 IN = XX(2) - XX(1): IF IN = 0 THEN IN = .00001
230 A = (YY(2) - YY(1)) / IN: B = YY(1) - A * XX(1)
240 IF A = 0 THEN A = .00001
250 HOME : VTAB (22): PRINT " COEFF. A " : A
260 PRINT " COEFF. B " : B
270 PRINT : INPUT " RETURN PER CONTINUARE " : S#
280 REM INDIVIDUAZIONE DELLE INTERSEZIONI
290 Y = AX: X = (Y - B) / A: GOSUB 570: GOSUB 630
300 Y = BX: X = (Y - B) / A: GOSUB 570: GOSUB 630
310 X = SX: Y = X + A + B: GOSUB 600: GOSUB 630
320 X = DX: Y = X + A + B: GOSUB 600: GOSUB 630
330 F = 0: HGR : HCOLOR = 3: GOSUB 660: GOSUB 690
340 REM DISEGNO DELLE INTERSEZIONI
350 FOR I = 1 TO T - 1: XX = XX(I) - 3: YY = YY(I): GOSUB 720
360 XX = XX(I) + 3: GOSUB 730
370 XX = XX(I): YY = YY(I) - 3: GOSUB 720
380 YY = YY(I) + 3: GOSUB 730: NEXT I
390 HOME : VTAB (22): INPUT " RETURN PER CONTINUARE " : S#
400 HGR : HCOLOR = 3: GOSUB 660: GOSUB 690
410 REM DISEGNO ESTREMI DEL SEGMENTO SE INTERNI
420 IF XX(1) < DX AND XX(1) > SX AND YY(1) < BY AND YY(1) > AY THEN XX =
XX(1): YY = YY(1): F = 1: GOSUB 720
430 IF XX(2) < DX AND XX(2) > SX AND YY(2) < BY AND YY(2) > AY AND F THEN
XX = XX(2): YY = YY(2): GOSUB 730: GOTO 560
440 IF XX(2) < DX AND XX(2) > SX AND YY(2) < BY AND YY(2) > AY THEN XX =
XX(2): YY = YY(2): F = 1: GOSUB 720
450 REM INDIVIDUAZIONE DI QUALE INTERSEZIONE E' QUELLA GIUSTA
460 IF XX(1) < = XX(3) AND XX(3) < = XX(2) THEN F1 = 1
470 IF XX(1) > = XX(3) AND XX(3) > = XX(2) THEN F1 = 1
480 IF XX(1) < = XX(4) AND XX(4) < = XX(2) THEN F2 = 1
490 IF XX(1) > = XX(4) AND XX(4) > = XX(2) THEN F2 = 1
500 REM UNO DEI DUE ESTREMI E' INTERNO
510 IF F AND F1 THEN XX = XX(3): YY = YY(3): GOSUB 730: GOTO 550
520 IF F AND F2 THEN XX = XX(4): YY = YY(4): GOSUB 730: GOTO 550
530 REM I DUE ESTREMI SONO ESTERNI E INTERSECONO
540 IF F1 AND F2 THEN XX = XX(3): YY = YY(3): GOSUB 720: XX = XX(4): YY =
YY(4): GOSUB 730
550 FOR I = 1 TO 6: XX(I) = 0: YY(I) = 0: NEXT I
560 HOME : VTAB (22): INPUT "RETURN PER CONTINUARE " : S# : GOTO 140
570 REM CONTROLLO SE INTERNO AL MARGINE ORIZZONTALE
580 F = 1: IF X < SX OR X > DX THEN F = 0
590 RETURN
600 REM CONTROLLO SE INTERNO AL MARGINE VERTICALE
610 F = 1: IF Y < AY OR Y > BY THEN F = 0
620 RETURN
630 REM VETTORE CONTENENTE LE INTERSEZIONI
640 IF F THEN XX(T) = X: YY(T) = Y: T = T + 1: RETURN
650 RETURN
660 REM CORNICE ESTERNA
670 XX = X1X: YY = Y1Y: GOSUB 720: XX = X2X: GOSUB 730: YY = Y2Y
680 GOSUB 730: XX = X1X: GOSUB 730: YY = Y1Y: GOSUB 730: RETURN
690 REM CORNICE INTERNA
700 XX = SX: YY = AY: GOSUB 720: XX = DX: GOSUB 730: YY = BY
710 GOSUB 730: XX = SX: GOSUB 730: YY = AY: GOSUB 730: RETURN
720 HPLLOT XX, YY: RETURN : REM FUNZIONE PLOT X, Y
730 HPLLOT TO XX, YY: RETURN : REM FUNZIONE PLOT TO X, Y

```

Figura 8 - Programma Segmenti - Listato. La complessità della casistica comporta una complessità dell'algoritmo che deve essere in grado di funzionare anche per i casi più difficili.

Programma Segmenti

Una trattazione più rigorosa del test dentro/fuori comporta necessariamente l'approfondimento del tema intersezione di un segmento con i margini della finestra. Abbiamo esemplificato in figura 7 parte della casistica. Un segmento può essere interno (non ha intersezioni) esterno (non ha intersezioni) o intersecare i margini (ha una o due intersezioni).

Esistono casi particolari come intersezione coincidente con l'intersezione dei margini (caso 4), oppure segmento che attraversa tutta la finestra (caso 1 e 2). E inoltre il problema è complicato in quanto la retta a cui appartiene il segmento interno o parzialmente interno comunque ha

interno) viene isolato in una subroutine poiché viene eseguito più volte.

I dati relativi al segmento sono chiesti via input (righe 160/180). Non c'è controllo di formato e quindi sarà buona cura dell'operatore immettere dati corretti.

Viene visualizzato il segmento (righe 190/200) rimandando alle solite subroutine di HLOT X, Y (riga 720) e HLOT TO X, Y (riga 740).

Viene ora individuata l'inclinazione IN della retta e i parametri A, B dell'equazione della retta contenente il nostro segmento (righe 210/240). I valori A, B sono visualizzati per controllo (righe 250/270).

A questo punto c'è la sezione relativa alla individuazione dei punti di intersezione tra la retta del segmento e le quattro

Terza fase

Dopo aver pulito la pagina grafica e ridisegnati i margini esterno e interno (riga 400) vengono eseguiti tre test successivi.

In riga 420 viene verificato se il primo punto è interno. Se è interno viene disegnato e viene settato il flag F.

In riga 430 viene verificato se il secondo punto è interno nel caso lo fosse anche il primo (in quanto $F = 1$). Se è interno tutto il segmento è interno, viene disegnato e non occorrono altri controlli.

In riga 440 viene verificato se il secondo punto è interno, nel caso che non lo fosse il primo (in quanto $F = 0$). Se è interno viene disegnato e il flag F viene settato.

Se usciamo da questi tre controlli con $F = 0$ significa che i punti estremi del seg-

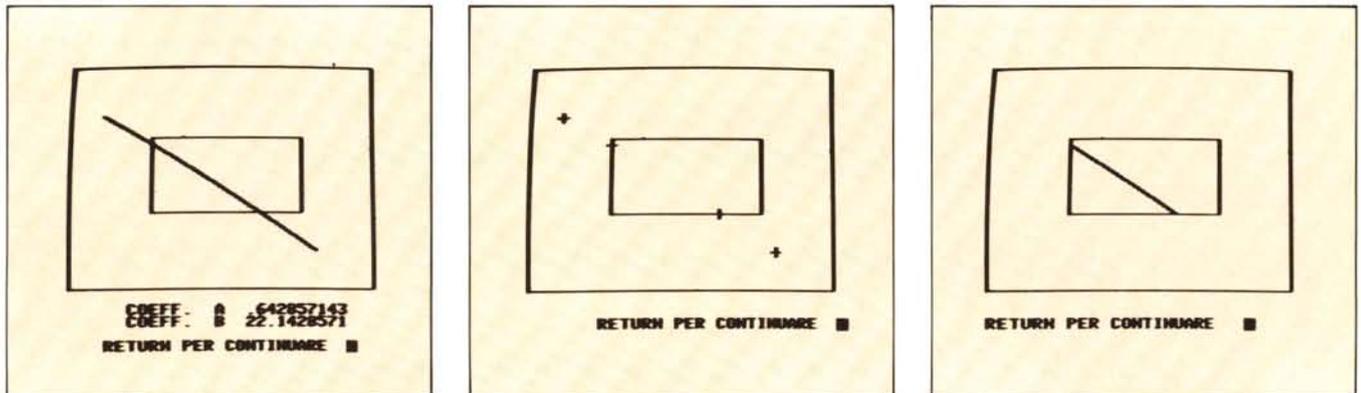


Figura 9 - 10 - 11 - Programma Segmenti - Output. Gli output sono tre in quanto prima viene visualizzato tutto il segmento, poi vengono visualizzati i punti estremi e gli eventuali punti di intersezione con la finestra e infine viene visualizzato il segmento o la porzione di segmento interno alla finestra.

due intersezioni con la finestra (caso 3). Si capisce come il test diventa in realtà molto complesso, in quanto non sapendosi a priori di quale caso si tratta, occorre prevederli tutti. Cercheremo di analizzare il processo logico descrivendo il programma (listato in fig. 8 e output in fig. 9, 10, 11) che lo riproduce fedelmente.

Il programma si svolge in tre fasi.

1ª fase - Comprende il disegno dei margini esterno ed interno, la richiesta via input dei punti identificanti il segmento da tracciare, il calcolo della retta cui appartiene il segmento.

2ª fase - Disegno dei punti esterni del segmento e delle intersezioni della sua retta con i margini del quadrante interno.

3ª fase - Disegno della porzione di segmento o del segmento se interno alla finestra.

Prima fase

Vengono inizializzati i valori dei parametri dei margini esterno e interno del disegno e azzerati i flag F, F1, F2 che sono utilizzati nel programma (righe 100/140). Viene accesa la pagina HGR (quella con quattro righe di testo) e vengono disegnati i margini (riga 150).

Il disegno dei margini (righe 660/680 per quello esterno e righe 690/710 per quello

rette costituenti i margini. Tali quattro rette sono:

- Y = A% per il margine in alto
- Y = B% per il margine in basso
- X = S% per il margine a sinistra
- X = D% per il margine a destra.

Sostituendo questi valori nell'equazione della retta $Y = A * X + B$ (di cui abbiamo già calcolato A e B) troviamo i quattro punti di intersezione tra le rette (righe 290/320).

Individuato il singolo punto va verificato se appartiene ai margini o se è esterno al margine. Questa verifica è eseguita dalla subroutine in riga 570, per verificare se il punto di intersezione con un margine orizzontale è interno od esterno, e dalla subroutine 600 per il margine verticale.

Utilizzando il flag F, vengono caricati i vettori X%(T), Y%(T) con i valori dei punti di intersezione interni trovati (righe 630/650). In tali vettori sono già immessi i valori dei punti estremi del segmento.

Seconda fase

A questo punto vengono disegnati, con una crocetta, i punti estremi del segmento e le varie intersezioni trovate (righe 340/390).

Non è finita qui in quanto per un segmento a cavallo della finestra (caso 3 nella fig. 7) non sappiamo ancora quale delle due intersezioni ci interessa.

mento sono esterni alla finestra, viceversa se $F = 1$ sono uno interno ed uno esterno, in quanto il caso tutti e due interni lo abbiamo già verificato.

Per individuare se il punto intersezione tra la retta del segmento e la retta del margine è quello che ci interessa basta verificare che il valore della coordinata X di tale punto sia compreso tra i valori della coordinata X dei due estremi.

Vengono eseguiti quattro test, due per ogni intersezione.

Il primo verifica che la prima intersezione sia quella che ci interessa (righe 460/470) e il secondo verifica l'altra intersezione (righe 480/490).

Le condizioni di uscita da questi test sono tre:

$F = 1, F1 = 1$: il segmento interseca il margine con un estremo fuori ed uno dentro e la intersezione che ci interessa è la prima che abbiamo trovato (riga 510).

$F = 1, F2 = 1$: è la seconda che abbiamo trovato (riga 520)

$F1 = 1, F2 = 1$: il segmento attraversa completamente la finestra e le intersezioni ci interessano tutte e due (righe 530/540).

A questo punto è disegnato il segmento se è tutto interno, la porzione interna se non lo è del tutto, non è stato disegnato niente se è tutto esterno. Il programma è finito; vengono azzerate le intersezioni e si ricomincia da capo. **MC**

