

# PROBLEMI DI GRAFICA BIDIMENSIONALE

Abbandoniamo per un po' la grafica tridimensionale per dedicarci all'esame di qualche problemino elementare di grafica bidimensionale.

Come nostra abitudine, affronteremo i problemi che via via ci poniamo nel modo più semplice possibile, realizzando programmi dimostrativi che potranno servire da spunto per i vostri programmi più consistenti o direttamente da subroutine in procedure più complesse.

Ad esempio un programmino che unisce con una linea tratteggiata due punti potrà servire come subroutine in un programma di grafica tridimensionale che individui e tratteggi quindi le linee nascoste.

Gli argomenti che tratteremo sono le linee tratteggiate, la individuazione dell'arcotangente, i raccordi parabolici.

# Le linee tratteggiate

Tutti i computer dotati di capacità grafi-

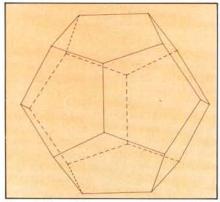


Figura 1 - Output del programma DODECAEDRO. Le linee visibili sono in tratto continuo, le linee nascoste sono in tratteggiato.

che hanno istruzioni di tracciamento di segmenti tra due dati punti. Una delle prime cose che viene in mente è quella di differenziare, nell'ambito di uno stesso disegno, le linee tra di loro.

Come esempio di applicazione c'è quello, ricordato in precedenza e sviluppato in seguito, di ausilio in grafica tridimensionale (es. fig. 1), oppure il tracciamento di curve che rappresentino differenti fenomeni o misurazioni.

Presentiamo dunque un programma che permette il fratteggio di una linea tra due punti.

Il programma è realizzato per il computer APPLE II che ha, come noto, buone capacità grafiche ed utilizza la istruzione HPLOT X,Y che traccia sullo schermo un punto di coordinate X,Y nel riferimento schermo, e la istruzione HPLOT TO X,Y che traccia un segmento dall'ultimo punto plottato al nuovo punto X,Y. Per trasportare il programma su altre macchine bisognerà, è ovvio, usare le istruzioni corrispondenti.

Utilizzando software grafici più potenti è possibile trovare istruzioni che direttamente eseguano un tratteggio; è quindi ovvio che la nostra routine serve solo a chi ha istruzioni grafiche simili a quelle descritte.

Il programma realizzato permette la scelta tra due possibilità:

2 - il tracciamento della linea tratteggiata dato il numero dei tratti.

La routine per il calcolo dei punti intermedi tra i due punti estremi è molto semplice, una volta noto l'angolo individuato dalla retta passante per i due punti e gli assi di riferimento.

Questo angolo si trova con la funzione ATN(Q), dove Q è il rapporto tra DY e DX, differenze di ordinate e di ascisse tra i due punti P1 e P2 da congiungere.

La funzione BASIC ATN(Q) è come noto affetta da una indeterminatezza di quadrante e quindi non possiamo utilizzare

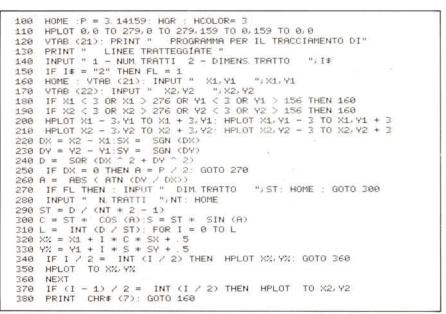


Figura 2 - Listato del programma TRATTEGGIO. Il listato è commentato nel testo. La riga 370 serve per completare anche l'ultimo tratto nel caso in cui la dimensione del tratto non sia sottomultiplo della distanza tra i due punti.

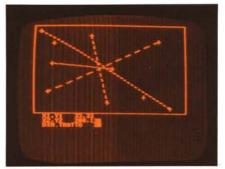


Figura 3 - Output del programma TRATTEGGIO. Le opzioni sono due. Tratteggio dato il numero dei tratti, tratteggio data la dimensione del tratto.

 1 - il tracciamento della linea tratteggiata data la lunghezza del singolo tratto; direttamente il valore numerico così calcolato, ma siamo costretti a individuare l'esatto valore dell'angolo con un altro sistema. Della funzione ATN(Q) parleremo dopo.

Passiamo invece ad esaminare il listato del programma di tratteggio, in figura 2 mentre in figura 3 vediamo l'output su video.

Lavoriamo con la pagina HGR in modo da utilizzare le quattro righe di testo disponibili per l'input dei dati.

Nelle righe 100-150, tracciamo la cornice del disegno e presentiamo l'opzione 1, per il tratteggio dato il numero dei tratti, e l'opzione 2, per il tratteggio data la dimensione del tratto. Se si sceglie l'opzione 2 il Flag FL viene settato a 1.

A questo punto vengono richiesti i punti estremi da unire con il tratteggio (righe 160-170); su questi punti viene eseguito il solito test di compatibilità con il formato schermo (righe 180-190) e se è superato vengono visualizzati i due punti, mediante due crocette sul video (righe 200-210).

A questo punto vengono calcolati i DX.DY e i rispettivi segni-SX,SY e la distanza D tra i due punti (righe 220 ÷ 240).

Prima di calcolare il valore dell'ATN(DY/DX) viene escluso il caso DX = 0 che dà come risultato P-greco/2 ma che comporterebbe un errore DIVISIONE PER ZERO.

Se si è scelta l'opzione 2 viene richiesta la dimensione ST del tratto, altrimenti viene chiesto il numero dei tratti NT, dal quale viene ricavato immediatamente il valore ST (righe 270 ÷ 290).

Vengono ora calcolate le costanti

C = ST\*COS(A), S = ST\*SIN(A) che rappresentano le proiezioni del passo ST lungo la X e la Y (riga 300). Si trova poi L, che è il numero dei passi intermedi su cui eseguire il loop (riga 310).

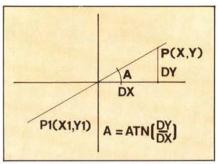


Figura 5 - Schizzo funzione ATN. Il valore DY/DX è uguale sia per il punto P che per il punto P1. L'angolo A invece differisce di 180 gradi.

A questo punto nel loop vengono calcolati gli X%,Y% intermedi (righe 320-330) che se sono nella posizione pari individuano l'inizio tratto (riga 340) altrimenti la fine tratto.

Va notato che nel calcolo di X%,Y% punti intermedi, SX e SY hanno la funzione di dare il giusto segno agli incrementi del passo. Ovvero il tratteggio se X1 > X2 o se Ŷ1>Y2 deve poter andare anche all'in-

Infine la riga 370 permette di tracciare anche l'ultimo segmento, di misura inferiore al passo, nel caso che il passo scelto non sia un sottomultiplo della D, distanza tra i due punti.

### Il dodecaedro

Ci sembra opportuno pubblicare anche il listato relativo alla fig. 1. Il programma

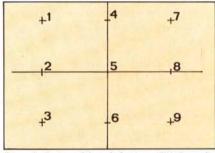


Figura 6 - Casi possibili per la funzione ATN. Generalizzando le combinazioni possibili per P(X,Y) sono nove, in quanto sia X che Y possono assumere valori <=>0.

che genera tale figura, listato in fig. 4, è una implementazione di quello pubblicato sul numero scorso e relativo al disegno dei solidi regolari.

L'implementazione riguarda una routine "manuale" di individuazione delle linee nascoste, tramite la quale la linea nascosta viene tratteggiata, a differenza delle altre linee. Per capire come funziona il programma rimandiamo al numero scorso, per quanto riguarda la routine principale relativa alla vista tridimensionale. Esaminiamo solo come vengono individuate e disegnate le linee nascoste e le linee visibili.

Nei vettori A(N), B(N) immettiamo i dati angolari di ciascun punto del dodecaedro. Nei vettori F%(M), T%(M) mettiamo le altre informazioni che ci servono per il disegno.

F%(M) è la sequenza con la quale vanno uniti tramite linee spezzate i vertici. Il valore zero interrompe la spezzata. Ad esempio i valori 1, 2, 3, 4, 0, 5, 6, 7 ecc. significano che i segmenti da tracciare sono 1/2, 2/3, 3/4, 5/6, 6/7, ecc.

Ad ogni valore di F%(I) corrisponde quindi la indicazione di un segmento da tracciare. A questa informazione associamo la T%(I) che indica se quel segmento è visibile (valore 0) o non è visibile (valore 1).

Se il valore è 1 viene richiamata la nostra routine di tratteggio. Il valore 2 non ha nessun significato in quanto l'abbiamo messo in corrispondenza del salto spezzata



Figura 4 - Listato del programma DODECAEDRO. Il programma è una implementazione del programma SOLIDI REGOLARI pubblicato sul numero scorso.

!	SX	SY	Z !	SUB
	1	1	9	430
!	1	0	! 8 !	420
!	1	-1	! 7 !	410
	0	1	! 6 !	400
!	0	. 0	! 5 !	390
!	0 ,	-1	! 4 !	380
!	-1	1	! 3 !	370
	-1	0	! 2 !	360
!	-1	-1	1 1	350
1 -			1	

Figura 7 - Tabella ATN. Sono riportate in tabella le nove differenti possibili situazioni.

e quindi non essendo individuato nessun segmento, non occorre sapere se è nascosto o meno. Anche questo programma può uscire su plotter mettendo PL=∅ in riga 130.

# La funzione ATN

La funzione arcotangente è quella che permette di individuare in un riferimento cartesiano l'angolo A formato dalla retta passante per un punto P(X,Y) e per l'origine  $O(\emptyset,\emptyset)$ .

Nelle macchine di tipo più scientifico (vedi ad esempio HP serie 80) le istruzioni sono due A = ATN(Q) e A = ATN2(X,Y).

Con la prima viene fornito l'angolo la cui tangente ha un valore Q. Ma poiché (vedi fig. 5) il valore Q è fornito dal rapporto DY/DX, tale funzione è affetta da indeterminazione nei segni e da errori nel caso che, al solito, il denominatore sia uguale a zero.

Ad esempio in figura 5 i punti P e P1 forniscono uno stesso valore Q anche se è evidente che non possono fornire uno stesso valore A.

Invece con la istruzione A = ATN2(X,Y) questi limiti non ci sono in quanto non viene fornito il rapporto tra X e Y, ma singolarmente X e Y, ed è quindi il computer che individua il quadrante esatto.

Non avendo la istruzione ATN2, il calcolo dell'angolo necessita di una routine di correzione che individui tutti i sottocasi e calcoli quindi esattamente i quadranti. Nella figura 6 sono raffigurati tutti i casi possibili, che sono anche listati in figura 7.

Il programma ARCOTANGENTE, listato in figura 8, output su monitor in figura 9, che abbiamo realizzato è un po' pedante, in quanto analizza tutti i sottocasi senza fare alcuna semplificazione che pur sarebbe possibile.

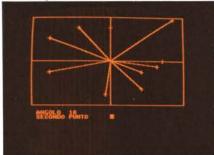


Figura 9 - Output del programma ATN. I valori dell'angolo sono calcolati in radianti e tradotti in gradi per la visualizzazione.

Il programma si divide in tre parti: righe 100-110: individuazione dei valori tipici degli angoli riferiti a P-greco;

righe 120-250: il programma, in questo caso grafico, che utilizza la subroutine;

righe 260-430: la routine che ci interessa è quella che calcola l'angolo A correttamente nel riferimento scelto.

Il nostro programma utilizza la pagina HGR dall'Apple II, per il disegno e quindi le quattro righe di testo per la sezione IN-PUT dati.

Viene immesso il primo punto (riga 160) che viene giudicato compatibile con il formato schermo (riga 170) e che genera il tracciamento di due assi cartesiani (riga 180)

Viene poi immesso il secondo punto (riga 190) che, subito il controllo (riga 200) genera un segmento che l'unisce all'origi-

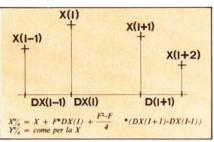


Figura 10 - Schema del programma RACCORDI. Il disegno illustra il procedimento di calcolo e fornisce le formule per i punti intermedi tra P(I) e P(I+1).

ne. A questo punto viene chiamata la routine principale (riga 230). Al ritorno l'angolo viene tradotto in gradi e visualizzato.

La routine di calcolo dell'angolo (riga 260) comporta il calcolo di DX, DY il calcolo dei segni di tali differenze.

Ricordiamo che la funzione SGN(X) dà come risultato -1, 0, +1 a seconda che X sia minore, uguale, maggiore di zero.

Con una semplice formuletta matematica (riga 280) viene calcolato, partendo da SX, SY, un valore Z, che varia tra 1 e 9 e che quindi permette il rinvio, mediante la istruzione di salto ON Z GOSUB... alle 9 subroutine, una per ogni caso considerato.

Viene calcolato l'arcotangente riferito al primo quadrante e richiamata la subroutine di correzione relativa al singolo quadrante.

# Il problema dei raccordi

Il problema di unire con una linea una serie di punti, dati secondo un certo ordine, si risolve in varie maniere.

La maniera più semplice consiste nel tracciare una spezzata.

Vi sono poi dei metodi statistici, che permettono di associare alla serie ordinata di punti una retta o una curva, ottenuta con metodi di interpolazione, che però non è detto che passi esattamente per i punti dati

Figura 8 - Listato programma ATN. Le combinazioni dei valori di SX, SY creano nove possibilità. Viene calcolato un valore Z che richiama, con la istruzione di riga 310, le 9 subroutine.

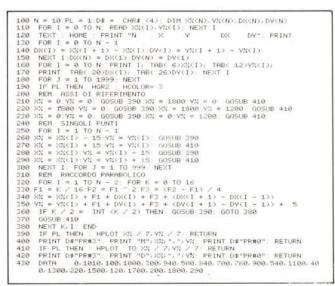


Figura 11 - Listato del programma RACCORDI APERTI. Il procedimento di calcolo viene facilmente tradotto in un programma. La routine principale è la 310-380.

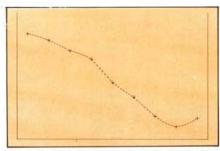


Figura 12 - Output del programma RACCORDI APER-TI. Per valutare l'andamento dei tratti di una curva occorre una definizione di output molto spinta.

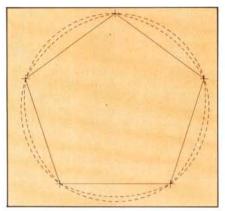


Figura 14 - Output del programma RACCORDI CHIU-SI. Il pentagono rappresenta la spezzata tra i punti, la linea tratteggiata esterna la circonferenza, la linea interna il raccordo parabolico.

Presentiamo un terzo sistema che permette di unire tra di loro i vari punti non con segmenti di retta, ma con linee paraboliche raccordate tra di loro.

Una delle utilizzazioni di tali metodi è nello studio delle traiettorie. Se si ha infatti una serie di rilevazioni che individuano i punti di passaggio di un proiettile, con metodi di questo tipo è possibile valutarne la traiettoria.

Nella figura 10 è riportato lo schema di calcolo del singolo punto P(XF,YF) sulla parabola che raccorda il punto P(I) con il punto P(I+1).

La F che compare nella formula varia tra 0 e 1 all'interno del raccordo a seconda dello step di calcolo voluto. Cioè se tra P(I) e P(I+1) vogliamo calcolare 10 punti intermedi F "loopperà" tra 0 e 1 con passo 0.1.

Si può osservare che ovviamente per  $F = \emptyset$  la XF coincide con X(I) e per F = 1 la XF coincide con X(I) + DX(I) cioè con X(I+1).

Si può infine notare che poiché la formula utilizza i DX precedenti e successivi al tratto interessato, non è possibile disegnare il primo e l'ultimo tratto.

A corredo di questo metodo abbiamo realizzato due programmi, il primo, RAC-CORDO APERTO, traccia la linea di raccordo per una serie aperta di punti, il secondo, RACCORDO CHIUSO, lavora su una linea chiusa.

Ambedue i programmi lavorano sia su plotter che su video. È infatti evidente che il problema dei raccordi richiede una defi-

Figura 13 - Listato del programma RACCORDI CHIUSI. Il procedimento è identico al precedente. Occorre solo utilizzare due punti in più per "chiudere" la curva.

nizione di calcolo e di disegno notevoli, per poter, ad esempio, apprezzare le differenze tra un tratto di parabola e un arco di circonferenza.

## Raccordo aperto

Il programma è listato in figura 11 e l'output su plotter è in figura 12. Nella riga 100 è fissato N numero dei punti in esame. PL è il flag Plotter. Per l'uscita su plotter va posto uguale a zero. Vengono poi caricati (righe 110-140) i dati dei punti X%(I), Y%(I) e calcolati i valori DX(I), DY(I). Il tutto viene visualizzato per un po', allo scopo di permetterne il controllo (150-180). A questo punto c'è la visualizzazione che avviene in tre fasi:

assi di riferimento (righe 200-230), tracciati semplicemente senza operazioni di scaling, che lasciamo volentieri a voi;

singoli punti (righe 240-300), visualizzati al solito con una serie di crocette;

linea di raccordo (righe 310-380). Ci sono due loop. Quello esterno dal punto 1 al punto N-2, in quanto partendo dal punto 1 si raggiunge il punto 2 e così via, rimangono esclusi come detto, le linee tra i punti 0 e 1 e tra i punti N-1 e N.

All'interno di ciascuna coppia di punti P(I), P(I+1) viene creato un loop sul K, nel nostro caso 16 passi, che genera il valore F, che abbiamo già descritto. Vengono così calcolati i 16 punti intermedi P(X%, Y%) tra P(I) e P(I+1). Tali punti intermedi vengono uniti a coppie formando un tratteggio. Il punto pari è quello di inizio

tratto (riga 360) il punto dispari è quello di fine tratto (370).

Dalla riga 390 alla riga 420 sono le due routine di HPLOT X%,Y% e di HPLOT TO X%,Y% richiamate dalle varie parti del programma. Se il flag PL è uguale a zero vengono eseguite le analoghe routine di plotter.

È appena il caso di notare che nei programmi con uscita alternativa monitor/plotter è consigliabile raggruppare in semplici subroutine tutte le istruzioni di plottaggio.

Abbiamo utilizzato come al solito il Plotter Watanabe, che ha un software di base particolarmente semplice e che abbiamo più volte descritto nel corso di questa rubrica.

### Raccordo chiuso

L'ultimo programma presentato è l'estensione del precedente al caso di una linea chiusa. (listato in figura 13, output in figura 14). Per rendere più evidente il risultato abbiamo utilizzato i cinque vertici di un pentagono tra i quali abbiamo tracciato il pentagono stesso, la circonferenza passante per essi e la nostra curva di raccordo.

Il procedimento, e quindi la routine principale del programma, è identico a quello descritto per il programma precedente.

L'unica cosa che va notata è che per "chiudere" la linea di raccordo occorre dare due punti in più. Ovvero il nostro pentagono avrà 7 vertici dei quali 4 coincidenti a due a due tra di loro.

# IL PIU'AVANZATO E' IL PIU' PREPARATO

Il leader della distribuzione europea oggi si muove nell'area produttiva italiana con la più alta preparazione tecnica per fornire componenti, sistemi e terminali

per computers tra i più avanzati.

Celdis vuol dire una costante garanzia per quelle Aziende che hanno bisogno di un servizio rapido, di un grande supporto tecnico applicativo e di prezzi competitivi.

Celdis è l'unico distributore italiano che dispone di uno staff di tecnici che si dedica esclusivamente ai terminali DIGITAL.

Distributore digital Autorizzato TERMINALI



Distributore steamwall Autorit Late

TELEFONA A CELDIS, RISOLVI PRIMA I TUOI PROBLEMI.



Celdis Italiana S.p.A.

Via F,lli Gracchi, 36 - 20092 Cinisello B. (Mi) Tel. (02) 612.00.41 Filiali:

10136 Torino - Via Mombarcaro, 96 Tel. (011) 35,93,12/35,93,69 35100 Padova - Via Savelli, 15 Tel. (049) 77.20,99/77.21.35 40138 Bologna - Via Massarenti, 219/4 Tel. (051) 53.33.36 00162 Roma - Via G. Pitrè, 11 int. 1 Tel. (06) 42,38.55/427.15.50