

Rotazione attorno ad un Asse

Questa puntata della rubrica di *Computer Grafica* è di tipo «pratico».

L'argomento è uno dei più facili della *Computer Grafica Tridimensionale* e riguarda la Rotazione di una figura piana attorno ad un asse, fino a formare un solido (detto appunto solido di rotazione). Per realizzare tali solidi si parte da una figura piana e la si fa ruotare, assieme al piano cui appartiene, attorno ad un asse.

In pratica ogni singolo punto della figura piana originaria traccia una circonferenza nello spazio attorno all'asse, per cui, date le coordinate X , Z , di ciascun punto è relativamente facile estrarle in tre coordinate spaziali.

Le varianti sul tema che ci permetteranno di implementare lo studio riguardano il collegamento tra le varie posizioni successive assunte durante la rotazione, la possibilità che l'asse di rotazione non sia compreso nel piano della figura base e infine che la stessa figura base, durante il movimento, subisca un movimento sul piano in cui giace.

Lavoreremo in *BASICA IBM* ma utilizzando comandi elementari in modo che sia possibile tradurre i programmi in qualsiasi altro linguaggio grafico. Rispettando le direttive della redazione, i listati non saranno più lunghi di una videata (24/25 righe).

Data la somiglianza tra i vari programmi presentati ne renderemo comuni due routine di tipo generale che listeremo solo nel programma numero 1 ma che vanno caricate anche negli altri. Tali routine partono da riga 900 e comprendono la inizializzazione dello schermo di lavoro e la routine di trasformazione da coordinate tridimensionali a coordinate video.

L'Ambiente di Lavoro

In figura 1 abbiamo realizzato una visualizzazione grafica dell'ambiente di lavoro. Il piano su cui giace la figura base è l' XZ , mentre l'asse di rotazione è l'asse Z . Quindi la figura base sarà espressa in coordinate X , Z . Nel corso della rotazione la coordinata Z rimarrà costante, mentre la coordinata X rappresenta la posizione iniziale del punto che ruotando percorre una circonferenza di raggio X , e che assume quindi anche un valore Y , variabile,

via via, tra $-X$ e X .

Da un punto di vista pratico aver assunto l'asse Z come asse di rotazione non rappresenta una limitazione in quanto l'assumere un altro asse, sempre giacente su XZ , è equivalente a far subire alla figura base una rotazione sul suo piano.

Premessa ai Programmi

Trattandosi di uno studio su una problematica comune, è evidente che i programmi presentati si somiglino tutti. Abbiamo cercato di sottolineare tale somiglianza in modo di rendere evidenti le differenze tra ciascun programma e il successivo; è proprio sulle differenze che va posta l'attenzione.

Tutti i programmi sono parametrizzati al massimo, e quindi tutte le costanti sono definite nelle prime righe, e le formule usano solo valori parametrizzati. In tal modo è facile verificare l'effetto del cambiamento dei parametri sul risultato finale.

Si tratta di programmi di grafica tridimensionale. Ma l'oggetto dei programmi è la problematica di traduzione delle coordinate da bidimensionali a tridimensionali, trattata decine di volte in questa rubrica.

Ripetiamo che le righe da 900 in poi del primo listato vanno riportate an-

che negli altri quattro in quanto comprendono routine comuni.

Programma Base

Listato fig. 2 e Output fig. 3

Le variabili inizializzate nelle prime righe riguardano dimensioni dello schermo di lavoro e dell'asse Z tracciato come riferimento (V , W e Q), riguardano gli elementi che definiscono la vista tridimensionale e cioè distanza e angoli di visuale (B , G , H). Riguardano infine i dati necessari alla rotazione (P , PI e SP).

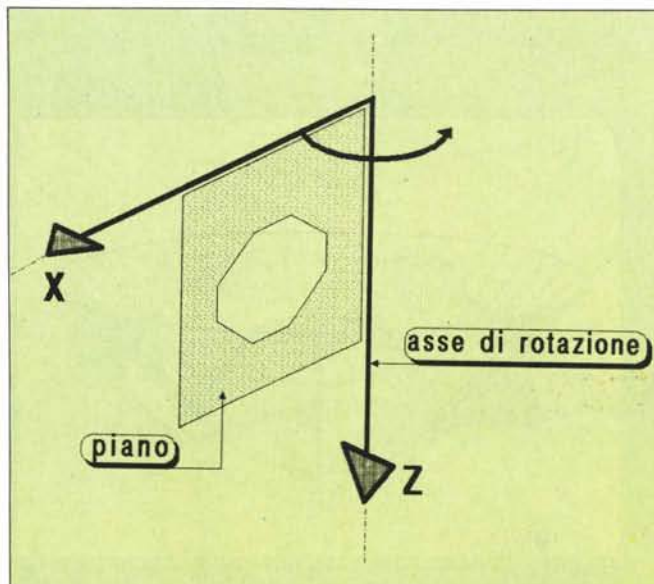
La figura base è composta da due segmentini uniti in un vertice, le coordinate relative ai tre punti sono X_i , Z_i e vengono lette tramite delle semplici *READ*.

Le routine in comune con gli altri programmi eseguono la pulizia e l'incorniciamento dello schermo (righe 900), tramite due istruzioni *WINDOW* e *LINE*, con parametri identici. Viene poi visualizzato l'asse Z (unendo i punti di coordinate $0, 0, -Q$ e $0, 0, Q$).

L'altra routine in comune è quella che trasforma le coordinate spaziali di ciascun punto (X , Y , Z) in due valori direttamente in formato video (M , N) e inizia da riga 940.

Il *MAIN* program, cioè quello che esegue la rotazione, comincia da riga

Figura 1
Schema di Lavoro.
La figura base, che sarà ruotata nello spazio, giace su un piano XZ , mentre l'asse di rotazione è l'asse Z stesso (almeno per i primi programmi).

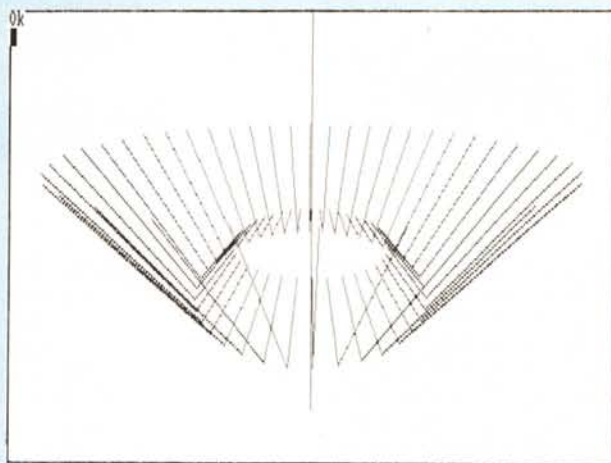


```

100 REM   rota01 - BASE
110 V=0:W=5:Q=5:B=8:G=.3:H=1.2:P=3.1416:PI=2*P:SP=P/ZO
120 S=SIN(G):C=COS(G):S1=SIN(H):C1=COS(H):GOSUB 900
130 REM   figura piana
140 READ X1,Z1,X2,Z2,X3,Z3:DATA 3,1,2,0,5,-2
150 REM   rotazione
160 FOR A=0 TO PI STEP SP
170 X=X1*COS(A):Y=X1*SIN(A):Z=Z1:GOSUB 940:X0=M:Y0=N
180 X=X2*COS(A):Y=X2*SIN(A):Z=Z2:GOSUB 940:LINE (X0,Y0)-(M,N)
190 X=X3*COS(A):Y=X3*SIN(A):Z=Z3:GOSUB 940:LINE (X0,Y0)-(M,N)
200 NEXT A:END
900 REM   iniziale. schermo e assi
910 CLS:SCREEN 2:WINDOW (V,W)-(-V,-W):LINE(V,W)-(-V,-W),,B
920 X=0:Y=0:Z=0:GOSUB 940:X0=M:Y0=N
930 X=0:Y=0:Z=0:GOSUB 940:LINE (X0,Y0)-(M,N):RETURN
940 REM   trasformazione
950 X0=-X*S+Y*C:Y0=-X*C+Y*S:Z0=-X*S1+C1-Y*S1*G
-Z*C1-B
960 M=B*X0/Z0:N=B*Y0/Z0:RETURN
    
```

Figura 2 - Programma Base - Listato. In questo listato sono presenti le routine generali (da riga 900 in poi) che vanno riportate anche negli altri quattro listati.

Figura 3 - Programma Base - Output. In tutti gli output riportiamo anche l'asse di rotazione per migliorare la comprensione del disegno.



150, e comprende il loop principale sull'angolo A, e per ogni valore assunto da quest'ultimo, calcola la posizione dei tre punti costituenti la figura base, e unisce i tre punti bidimensionali risultanti.

Collegamento tra i Vari Piani Intermedi

Il passo immediatamente successivo è quello di collegare i vari punti non solo all'interno del singolo piano per

comporre la figura, ma anche da un piano all'altro, in modo da rappresentare «tridimensionalmente» la composizione spaziale.

Per far ciò occorre memorizzare ciascuna posizione, che assume il senso di posizione precedente, in modo da collegarla alla posizione successiva, e quest'ultima una volta visualizzata diventa anch'essa precedente.

In pratica occorre utilizzare due coppie di vettori, la prima con le coordinate della figura iniziale, la seconda in cui, via via che la composizione va avanti, vengono parcheggiati i valori

delle coordinate della posizione assunta in precedenza.

Oppure si può decidere di memorizzare tutta la figura, immagazzinando in una coppia di matrici tutte le coordinate di tutte le posizioni. Le matrici sono a due dimensioni in cui la prima rappresenta il punto rispetto alla figura nel piano e la seconda il punto rispetto alla sua traiettoria circolare attorno all'asse.

Programma Canestro

Listato fig. 4 e Output fig. 5

```

100 REM   rota02 - CANESTRO
110 V=0:W=5:Q=5:B=8:G=.3:H=1.2:P=3.1416:PI=2*P+.001:SP=P/ZO
120 S=SIN(G):C=COS(G):S1=SIN(H):C1=COS(H):GOSUB 900
130 REM   figura piana
140 READ X1,Z1,X2,Z2,X3,Z3:DATA 3,1,2,0,5,-2
150 REM   rotazione
160 A=0:S2=SIN(A):C2=COS(A)
170 X=X1+C2:Y=X1*S2:Z=Z1:GOSUB 940:X0=M:Y0=N:M1=M:N1=N
180 X=X2+C2:Y=X2*S2:Z=Z2:GOSUB 940:LINE (X0,Y0)-(M,N):M2=M:N2=N
190 X=X3+C2:Y=X3*S2:Z=Z3:GOSUB 940:LINE (X0,Y0)-(M,N):M3=M:N3=N
200 FOR A=SP TO PI STEP SP:S2=SIN(A):C2=COS(A)
210 X=X1+C2:Y=X1*S2:Z=Z1:GOSUB 940:X0=M:Y0=N:LINE (M,N)-(M1,N1):M1=M:N1=N
220 X=X2+C2:Y=X2*S2:Z=Z2:GOSUB 940:LINE (X0,Y0)-(M,N):LINE -(M2,N2):M2=M:N2=N
230 X=X3+C2:Y=X3*S2:Z=Z3:GOSUB 940:LINE (X0,Y0)-(M,N):LINE -(M3,N3):M3=M:N3=N
240 NEXT A:END
    
```

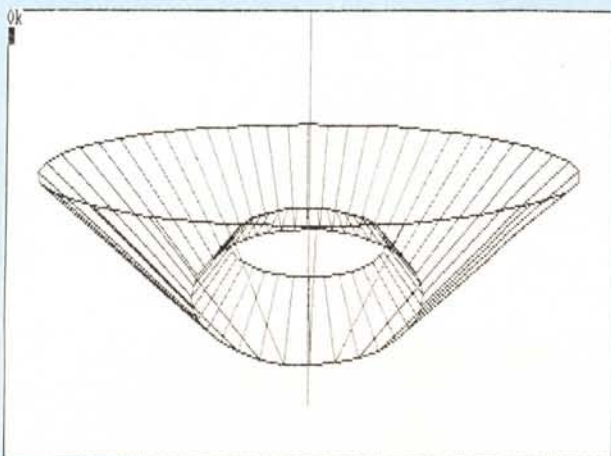


Figura 4
Programma Canestro - Listato. La prima variante consiste nell'unire con segmenti i punti nello spazio che indicano le varie posizioni assunte dall'oggetto.

Figura 5
Programma Canestro - Output. Con una semplice figura di base (due segmenti contigui) si ottiene un disegno che somiglia vagamente ad un canestro.

Le varianti, rispetto al programma precedente, sono minime e riguardano l'uso di variabili d'appoggio, sulle quali parcheggiare via via i valori dei punti di ciascun segmento da collegare con i corrispondenti punti assunti dal segmento successivo.

L'unica difficoltà consiste nel fatto che solo per la prima posizione non vi è una posizione precedente da collegare. Conseguentemente occorre isolare il primo valore dell'angolo A e iniziare il loop dal valore SP.

Vanno ripetute due volte, la prima per il solo valore A=0, le routine di calcolo e di collegamento.

Programma Poligono

Listato fig. 6 e Output fig. 7

Logica evoluzione è quella che consiste nel permettere il caricamento di figure più complesse che possono essere caricate in due vettori X(I), Z(I) nei quali sono immagazzinati i valori delle coordinate che la figura base assume nel piano XZ.

Se poi si utilizza il metodo di collegare le posizioni successive occorre disporre di due vettori per il parcheggio delle posizioni precedenti.

Va apportata un'altra modifica nella routine principale per permettere il disegno del poligono, per il quale il primo punto va trattato in maniera differente dagli altri, così come il primo angolo va trattato in maniera differente dagli altri.

L'Anello di Moebius

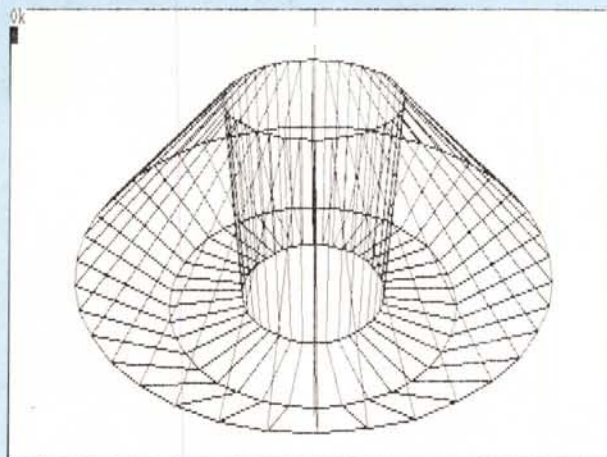
Una delle figure solide più originali è quella definita «Anello di Moebius» dal nome dello scienziato che per primo l'ha studiata. Da un punto di vista «tecnico» è una figura solida con una sola faccia e con un solo spigolo. La si ottiene praticamente con una striscia di carta lunga e stretta della quale si incollano tra di loro i due estremi corti.

Se prima di incollarli si ruotano di mezzo giro (leggi 180 gradi) i due estremi si uniscono e quindi si confondono le due facce opposte. Se con il dito percorrete uno spigolo o una faccia vi accorgete che in realtà vi è un solo spigolo e una sola faccia.

Anche l'anello di Moebius ha le sue varianti costituite dal numero di «giri» cui si sottopongono gli estremi prima del congiungimento, e dal fatto che dopo il congiungimento la striscia venga divisa in due o più parti.

```

100 REM  rota03 - POLIGONO
110 V=4;W=3;D=4;B=8;G=.3;H=.9;F=3.1416;PI=2*P+.001;SP=P/20
120 S=SIN(G);C=COS(G);S1=SIN(H);C1=COS(H);GOSUB 900
130 REM  figura piana
140 NP=S;DIM X(NP),Z(NP),A(NP),B(NP);FOR I=1 TO NP:READ X(I),Z(I):NEXT
150 REM  rotazione - prima porzione
160 A=0;I=1
170 X=X(I)*SIN(A);Y=X(I)*COS(A);Z=Z(I);GOSUB 940:PSET(M,N);A(I)=M;B(I)=N
180 FOR I=2 TO NP:X=X(I)*SIN(A);Y=X(I)*COS(A);Z=Z(I);GOSUB 940
190 LINE -(M,N);A(I)=M;B(I)=N:NEXT I
200 REM  rotazione - seconda porzione
210 FOR A=SP TO PI STEP SP:I=1
220 X=X(I)*SIN(A);Y=X(I)*COS(A);Z=Z(I);GOSUB 940:PSET(M,N)
230 LINE (A(I),B(I))-(M,N);A(I)=M;B(I)=N
240 FOR I=2 TO NP:X=X(I)*SIN(A);Y=X(I)*COS(A);Z=Z(I);GOSUB 940
250 LINE (A(I-1),B(I-1))-(M,N);LINE -(A(I),B(I));A(I)=M;B(I)=N:NEXT I:NEXT A
260 END
270 DATA 1,1,2,1,3,0,1,-2,1,1
  
```



▲ Figura 6 - Programma Poligono - Listato. Ulteriore generalizzazione. La figura di base (che giace sul piano XZ) è un poligono di NP vertici.

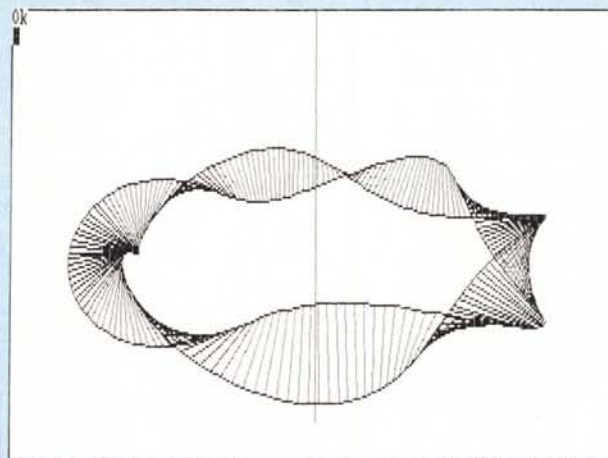
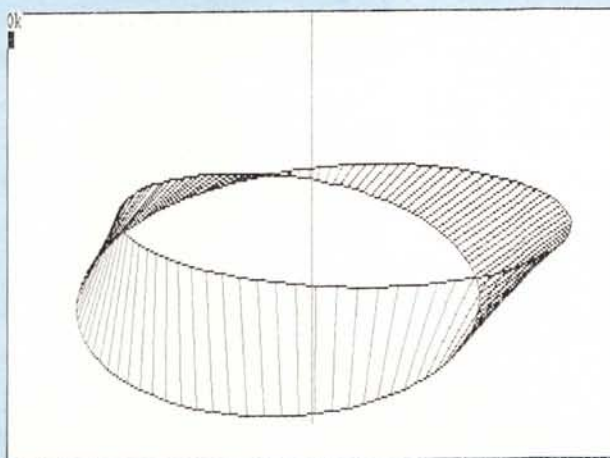
Figura 7
Programma Poligono Output. Il poligono che ruota è un pentagono irregolare, mentre il solido di rotazione risultante ne maschera la forma.

Figura 8 - Programma Moebius - Listato. Viene ruotato un segmento che, durante la rotazione attorno all'asse, ruota anche su se stesso.

```

100 REM  rota04 - MOEBIUS
110 V=6;W=4;D=4;B=12;G=.3;H=1.2;F=3.1416;PI=2*P+.001;SP=P/100
120 S=SIN(G);C=COS(G);S1=SIN(H);C1=COS(H);GOSUB 900
130 REM  figura piana
140 READ XC,ZC,R,I:DATA 4,0,.6,3
150 REM  rotazione - prima porzione
160 A=0;C3=COS(A);S3=SIN(A);C2=COS(A#K);S2=SIN(A#K)
170 X1=XC+R#C2;Z1=ZC+R#S2;X2=XC-R#C2;Z2=ZC-R#S2
180 X=X1#C3;Y=X1#S3;Z=Z1;GOSUB 940:X0=M;Y0=N;X6=M;Y6=N
190 X=X2#C3;Y=X2#S3;Z=Z2;GOSUB 940:LINE (X0,Y0)-(M,N);X7=M;Y7=N
200 REM  rotazione - seconda porzione
210 FOR A=SP TO PI STEP SP:C3=COS(A);S3=SIN(A);C2=COS(A#K);S2=SIN(A#K)
220 X1=XC+R#C2;Z1=ZC+R#S2;X2=XC-R#C2;Z2=ZC-R#S2
230 X=X1#C3;Y=X1#S3;Z=Z1;GOSUB 940:X0=M;Y0=N
240 LINE (M,N)-(X6,Y6);X6=M;Y6=N
250 X=X2#C3;Y=X2#S3;Z=Z2;GOSUB 940:LINE (X0,Y0)-(M,N)
260 LINE (M,N)-(X7,Y7);X7=M;Y7=N
270 NEXT A:END
  
```

Figure 9 e 10 - Programma Moebius - Output. Variando il parametro K, si ottiene un diverso numero di rotazioni del segmento BASE attorno a se stesso.



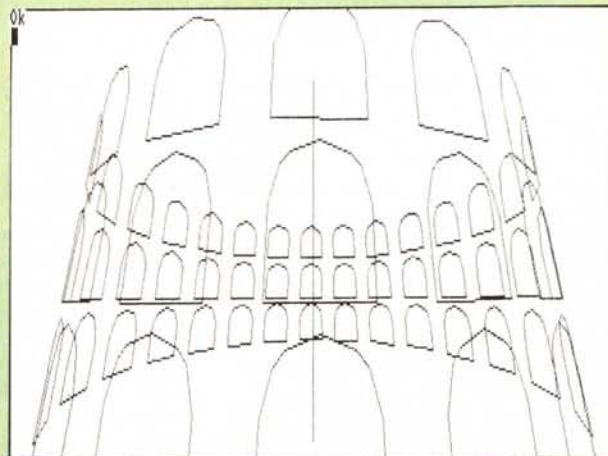
```

100 REM .rota05 - COLOSSEO
110 V=150:W=100:O=100:B=150:G=.5:H=1.2:P=3.1416:PI=2*P+.001:SP=P/15
120 S=SIN(G):C=COS(G):S1=SIN(H):C1=COS(H):GOSUB 900
130 REM figura piana
140 NP=10:DIM X(NP),Z(NP),X1(NP),Z1(NP):R=60:T=60
150 FOR I=1 TO NP:READ X(I),Z1(I):NEXT I
160 REM loop sui piani
170 FOR FP=0 TO 3:FOR I=1 TO NP:Z(I)=Z1(I)+FP*15-20:NEXT I
180 REM rotazione prima porzione
190 I=1:A=0:D=A+X(I)/T
200 X=R*SIN(O):Y=R*COS(O):Z=Z(I):GOSUB 940:PSET(M,N)
210 FOR I=2 TO NP:D=A+X(I)/T:X=R*SIN(O):Y=R*COS(O):Z=Z(I):GOSUB 940
220 .LINE -(M,N):NEXT I
230 REM rotazione seconda porzione
240 FOR A=SP TO PI STEP SP:I=1:D=A+X(I)/T
250 X=R*SIN(O):Y=R*COS(O):Z=Z(I):GOSUB 940:PSET(M,N)
260 FOR I=2 TO NP:D=A+X(I)/T:X=R*SIN(O):Y=R*COS(O):Z=Z(I):GOSUB 940
270 .LINE -(M,N):NEXT I:NEXT A:NEXT FP:END
280 DATA 5,0,5,-7,-4,-10,3,-11,0,-12,-3,-11,-4,-10,-5,-7,-5,0,5,0

```

Figura 11 - Programma Colosseo - Listato. Questa volta la figura ruota rispetto ad un asse non più giacente sullo stesso piano, ma giacente sul piano perpendicolare all'asse della figura stessa.

Figura 12 e 13 - Programma Colosseo - Output. Anche in questo caso tutti gli elementi sono parametrizzati, per cui cambiandone alcuni si ottengono viste differenti.



Dal nostro punto di vista la figura base è un semplice segmento che, durante la rotazione di 360 gradi attorno all'asse, subisce una rotazione di 180 gradi (o suoi multipli) attorno a se stesso.

Programma Moebius

Listato fig. 8 e Output fig. 9, 10

Il listato è semplificato rispetto ai precedenti in quanto il disegno base è un comune segmento. Di tale segmento però non vengono fornite le coordinate degli estremi ma le coordinate del centro e la lunghezza. In tale modo è possibile far ruotare il centro del segmento attorno all'asse e il segmento attorno al suo centro.

È stato parametrizzato il rapporto tra le due rotazioni (parametro K), per cui è possibile eseguire da 0 a N rotazioni variando sensibilmente il risultato «estetico» dell'operazione.

Nelle due figure presentate vediamo due esempi, il primo mezzo giro e il secondo due giri. Dalla prima alla seconda versione sono stati cambiati anche il passo del loop sull'angolo e la lunghezza del segmentino.

Il Colosseo

Un ulteriore sviluppo della problematica riguarda la eventualità di far ruotare l'oggetto non più attorno ad un asse giacente sullo stesso piano della figura base, ma attorno ad un asse comunque posto nello spazio. Se quest'asse giace su un piano perpendicolare all'asse della figura si ha una rotazione della stessa su una superficie pressoché cilindrica.

L'abbiamo definito Colosseo perché in definitiva anche il Colosseo può essere assimilato ad un solido di rotazione ottenuto ruotando un singolo arco per più volte attorno ad un asse nello spazio.

Anche in questo caso adotteremo tutte le semplificazioni che tale problema permette e che consistono sostanzialmente nel fatto che il piano è perpendicolare all'asse. Questo permette, ancora una volta, di evitare l'uso di coordinate tridimensionali per definire il disegno base.

Infatti ciascun vertice dell'arco è comunque distante dall'asse di rotazione di un valore R, il valore Z, non varia durante la rotazione, mentre il valore X assume il significato di spostamento

in avanti o indietro del vertice rispetto all'asse dell'arco.

Quindi in pratica ciascun vertice è individuato da R (che è fisso) da Z, che non varia con l'angolo, e da due valori X e Y che dipendono dall'angolo A e da un piccolo spostamento aggiuntivo che a sua volta dipende dal valore X(I) originario.

Programma Colosseo

Listato fig. 11 e Output figg. 12, 13

Il singolo arco è fornito via READ DATA in riga 280. I parametri aggiuntivi sono R, raggio della figura e T altro parametro che indica il fattore di spostamento di ciascun punto.

Va poi notato che poiché viene disegnato un arco per ogni angolo, il numero totale degli archi dipende dalla misura del passo tra gli angoli, e quindi a sua volta dal raggio.

È evidente la relazione reciproca che lega dimensioni dell'arco, lunghezza del raggio e numero di archi disegnabili per piano. Tali elementi possono essere modificati con facilità per ottenere varianti del disegno. **MC**



Windows



Con WINDOWS, completamente in italiano, userete il vostro personal con una facilità mai raggiunta finora, in un nuovo ambiente grafico con icone e menù, eseguendo in multitasking più programmi nello stesso momento. Con WINDOWS avrete tutti i vantaggi dei convenzionali pacchetti integrati, senza subirne i compromessi: potrete voi stessi unire le applicazioni di vostro interesse per ottenere un integrato personalizzato. WINDOWS contiene inoltre una serie di moduli tutti in italiano quali WRITE, programma di elaborazione testi semplice da apprendere e da usare, PAINT per disegno e grafica, e tutta la serie di funzioni da scrivania con Agenda, Schedario, Blocco Note, Appunti, Calcolatrice, Orologio ed Emulazione Terminale. E, con Spooler, potrete anche stampare mentre state lavorando su un qualunque programma. Il tutto, per il vostro personal computer IBM®, Olivetti®, o compatibile, a lire 216.000. WINDOWS è disponibile presso i Distributori Autorizzati Microsoft ed i Rivenditori più qualificati.

MICROSOFT®

Potenza e semplicità. Insieme.

MICROSOFT S.p.A. - 20093 Cologno M^o (MI) - Via Michelangelo, 1
tel. (02) 2549741 r.a. - fax (02) 2549745