COMPUTER ILLUSTRATION OF NANNOFOSSILS

Jeremy R. Young, Geology Dept., Imperial College, London SW7 2BP.

1. INTRODUCTION

At the London INA meeting, a number of people expressed interest in the computer programs I demonstrated, for drawing coccoliths and discoasters, in two dimensions. Since these were written on a BBC micro in BBC BASIC the actual programs cannot be directly used on many computers, but the underlying principles are reasonably simple. The following notes are intended to help others use the same technique. Inevitably they will seem trivial to some workers, but I would have found the discussion useful when I started programming, and hopefully others will do.

My primary reason for writing the programs was to use the computer as a technical drawing tool; to remove some of the labour involved in producing complex geometrical figures. In addition, computer drawings have the considerable advantage that, like word-processed text, they can readily be altered and reused. Finally, for the more asymmetric forms of elliptical coccoliths, the rather indirect technique of computer illustration required development of a model for coccolith geometry; and provided a means of testing the predictions of this model.

2. BASIC TECHNIQUE, AND ILLUSTRATION OF DISCOASTERS.

Computer graphics work primarily by cartesian co-ordinates, dividing the screen into several thousand pixels. These pixels are illuminated or not according to the values held in a corresponding area of screen memory. This screen memory can also be readily saved on disc, or dumped to a printer, to produce hard copy of the screen display. There are two basic types of technique for producing computer graphics: The first is from within languages, such as BASIC, using simple graphics commands (such as DRAW), working directly with the cartesian co-ordinate system. The second technique is to use specialist graphics packages, which allow more direct interaction between the user and the graphics, with the aid of mice, cursor controls, or whatever. I have used the former technique since it is more suitable for handling geometrical and symmetrical figures. Drawing nannofossils, such as discoasters, with perfect radial symmetry is particularly easy. Irregular forms, such as ceratoliths, are unsuitable for illustration by this technique.

The problem of drawing discoasters consists of two main parts - defining the shape of a single ray, and symmetrically reproducing it as required. The shape of the ray needs to be defined in terms of the co-ordinates of a series of points around its edge. I did this by the most rudimentary digitising technique available, drawing the ray on graph paper, and using the resultant points as data statements at the end of the program. Alternatives would be to use a digitising tablet; or to write a separate program to interactively design the shape on the screen.

Given the co-ordinates of a set of points it is a simple trigonometrical problem to reproduce these points in other orientations, at different scales, etc. Some graphics systems include commands to do this, but writing the program directly is more flexible. The main equations are those for rotating a point (X,Y) about the origin through an angle (a), to a new position (X',Y'): `X'= X*-COSa + Y*SINa` & `Y'= X*SINa + Y*COSa` .

The listing shows how these elements can be combined into a short program. This is written in BBC BASIC, but since it contains few obscure commands it could easily be converted into other versions of BASIC, or another language. It is a convenient start point for experimenting with, and can be used for any radially symmetrical forms.

3. COCCOLITH DRAWING PROGRAM

Circular coccoliths can be drawn in a similar way to discoasters, but elliptical coccoliths have a greatly reduced symmetry and a more specialised program is needed. For this a set of assumptions, or model, of coccolith geometry is needed. I used the following assumptions; their justification, and implications, are discussed in detail elsewhere (YOUNG 1987, and in prep.).

1. All rays are similar in size and shape - although to compensate for the ellipticity they overlap, or grow sideways, to differing extents. This is based mainly on observations of electron micrographs of coccoliths, in particular rim width, and so element length can be shown to be constant.

2. The rays grow from an initial ellipse, or proto-coccolith ring, near the inner margin of the coccolith rim. This is based on the sequence of coccolith development documented in studies of coccolithogenesis.

3. The rays are evenly spaced around this initial ellipse.

4. The rays radiate from this initial ellipse at a constant angle, to the local tangent to the ellipse.

The last two assumptions, and the use of a true ellipse for the proto-coccolith ring, are convenient assumptions based on examination of electron micrographs, but are not rigorously proven.

Converting these into a program involved a straightforward, but rather complex, development of the approach used in the discoaster drawing program. I defined the shape of individual rays in the same way as for the discoasters. In addition the ellipticity and radius of the proto-coccolith ring, the ray spacing, ray angle, and ray scale, need to be specified for each drawing. One of the advantages of the technique is that each of these parameters can be altered independently; and so the relationship between them and the final coccolith form can be explored.

Additional routines were needed to determine, the origin positions, and angles, of each ray (using Assumptions 2-4). Also, to compensate for the overlap of rays, further routines were necessary; in order not to draw the portions of rays covered by adjacent rays (ie to vary the width of the rays). This was done by testing each pixel between the start and end positions of relevant lines, until the edge of the preceding ray was found. This position was then taken as the end of the line.

This program is too long to list conveniently, and would require considerable modification to convert from BBC BASIC. An experienced programmer should, however, be able to write a similar program reasonably easily from, the assumptions above.

REFERENCES

YOUNG J.R. 1987: Neogene nannofossils from the Makran of Pakistan and the Indian Ocean. Unpubl. PhD thesis, University of London.

YOUNG J.R. *in prep*: Heterococcolith structure and development. For, CRUX J.A. & van HECK S.E. (Eds.), Procs. INA Conf., London 1987.

DISCOASTER DRAWING PROGRAM: LINE NOTES

(See listing on next page)

10-40. PRELIMINARIES

20. Asks if print out is wanted. If so diagram needs to be stretched slightly to compensate for the distortion produced by the printer.

30,40. Clears the screen; sets graphics mode; sets up array *co-ords* for *DATA*; and draws border.

60-210. MAIN PROGRAM LOOP, repeated for each of the three sub-diagrams (distal, proximal and side views).

70. *READ* from *DATA*: the origin (*OX%, OY%*); *scale*; orientation (*ang*); number of rays (*R%*); and name (*name$*), of sub-diagram. Set graphics origin for sub-diagram.

80-190. Sub-loop drawing each feature of sub-diagram, (e.g. ray outlines, or central boss).

90. *READ* from *DATA* the number of steps (*S%*), and *PLOT* mode (*K%*), of feature.

100-120. *READ* co-ordinates of each step from *DATA* into array (*co-ords*). N.B. *control$* is primarily a separator in the data statements (usually *£*).

130-180. Plotting loop, repeated for each ray; *a* is the angle of the ray.

140. Move (*PLOT* mode 4) to beginning of feature.

150. Draw first half-ray of feature.

170-180. Draw second half-ray of feature (omitted for asymmetrical ray features).

230. Print screen, if required.

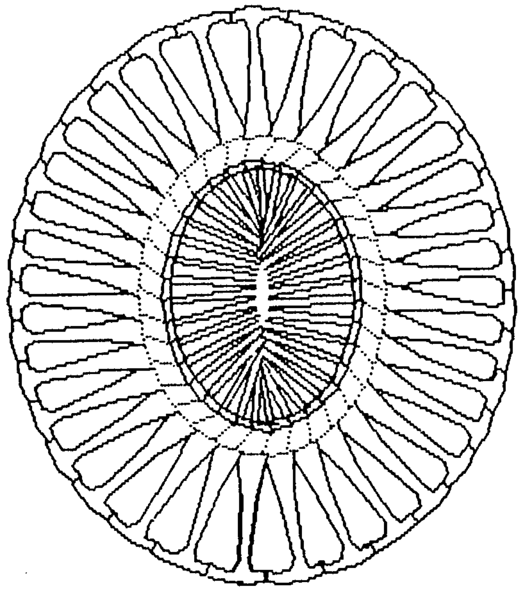
250-270. Procedure *PROCdr*: Called from the program, for drawing from current point to the next. The co-ordinates of the new point are calculated for the current ray angle (*a*), from the co-ordinates for the equivalent point on the first ray (*p,q*). This uses the general equation for rotation of cartesian co-ordinates. A line is then drawn from the current point to the new point, in *PLOT* mode *K%* (53 dotted line, 5 solid line, 4 no line).

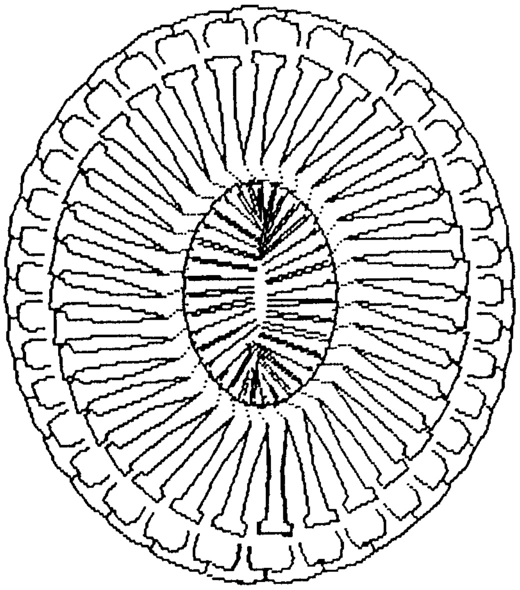
290-390. *DATA*

290. Basic *DATA* for first sub-diagram (*READ* by line 70).

300. *DATA* for first feature of sub-diagram (*READ* by lines 90 and 110). The first two figures are the number of points (*S%*) and the *PLOT* mode (*K%*). Subsequent numbers are the x and y co-ordinates of each point, separated by the filler *control$*. *control$* is also used to note what each data set draws, and to indicate the end of sub-diagrams (*STOP*), and of the entire diagram (*END*).

EXAMPLE FROM COCCOLITH PROGRAM





```
10  REM DISCOASTER DRAWING PROGRAM
20  INPUT print$: IF print$="Y" THEN dist=1.193 ELSE dist=1
30  MODE128: DIMco_ords(25,2)
40  PLOT5,5,5: PLOT5,5,1010: PLOT5,1274,1010: PLOT5,1274,5: PLOT5,5,5
50  :
60  REPEAT
70    READ OX%,OY%,scale,ang,R%,name$: VDU29,OX%;OY%;
80    REPEAT
90      READ S%,K%
100     FOR step = 1 TO S%
110       READ co_ords(step,1), co_ords(step,2), control$
120     NEXT step
130     FOR a=2*PI/R%+RAD(ang) TO 2*PI+RAD(ang) STEP 2*PI/R%
140       N=K%: K%=4: PROCdraw(co_ords(1,1),co_ords(1,2)): K%=N:
150     FOR T%=1 TO S%: PROCdraw(co_ords(T%,1),co_ords(T%,2)): NEXT T%
160       IF control$="ASYM" GOTO190
170       N=K%: K%=4: PROCdraw(-co_ords(S%,1),co_ords(S%,2)):K%=N
180     FOR T%=S% TO 1 STEP -1: PROCdraw(-co_ords(T%,1),co_ords(T%,2)): NEXT T%
190     NEXT a
200   UNTIL control$="STOP" OR control$="END"
210 UNTIL control$="END"
220 :
230 VDU26: IF print$="Y" THEN *RUN SCREENDUMP
240 END
250 DEFPROCdraw(p,q)
260 PLOT K%, scale*(p*-COSa+q*SINa), scale*dist*(p*SINa+q*COSa)
270 ENDPROC
280 REM Date to draw Discoaster bollii
290 DATA 600,50,2.5,0,1, SIDE VIEW
300 DATA 5,5,12,50,distal boss,11,64,£,14,72,£,10,80,£,0,78,£
310 DATA 3,53,63,33,prox ridges,26,28,£,22,16,£
320 DATA 10,5,0,50,main rays,50.,50,£,65,45,£,120,42,£,120,38,£,63,33,£,52,23,£,22,16,£,12,7,£,0,7,STOP
330 DATA 960,600,2.5,0,6, PROXIMAL VIEW
340 DATA 2,5,6,11,prox boss,0,19,£,3,53,1,19,£,3,30,£,3,60,£
350 DATA 10,5,23,40,body,17,43,£,11,52,£,6,70,£,5,100,£,12,110,£,14,120,£,11,120,£,5,113,£,0,111,STOP
360 DATA 320,600,2.5,0,6, DISTAL VIEW
370 DATA3,5,13,22,distal boss,10,22,£,0,14,£
380 DATA 2,53,23,40,sutures,13,22,ASYM
390 DATA 10,5,23,40,body,17,43,£,11,52,£,6,70,£,5,100,£,12,110,£,14,120,£,11,120,£,5,113,£,0,111,END
```