

COMPUTER TECHNOLOGY

A low-cost interactive coordinate plotter

BRIAN L. PARTRIDGE and J. MICHAEL CULLEN

Department of Experimental Psychology, South Parks Road, Oxford, England

An inexpensive system for on-line coordinate plotting is described. Users with access to a computer running ALGOL or FORTRAN, and equipped with analog-to-digital converters, can build the system for less than \$50. The program is readily modified for the individual user's requirements, and interactive features allow data correction and analysis to be carried out during coordinate plotting, with operators warned of plotting errors as they occur.

In recent years researchers have turned increasingly to film and video techniques for detailed analysis of animal movement. Using such techniques generally involves spending vast amounts of time analyzing the resultant records and then keypunching the information into a form recognizable by computer. Several automatic systems for coordinate plotting are commercially available (e.g., D-MAC), but most suffer from two major drawbacks: (1) They are relatively inflexible, that is, they make no provision for data correction or formatting the output, and (2) they are prohibitively expensive. We have developed an interactive coordinate plotter which solves both of these problems. Because the progress is written in a high-level language (ALGOL), it is readily tailored to meet individual requirements and can be run on most computers. High-level language programming not only allows users to scale coordinates in real space and to format data output as they choose, but also allows substantial data analysis and correction while plotting. Moreover, data can be stored in arrays and only output after an entire frame (or other specified unit of analysis) has been checked for mistakes.

The sorts of errors that plague operators of most systems are easily eliminated: Users are warned by the computer if they depart from the expected plotting sequence by, for example, plotting too few or too many points in any film or video frame or plotting points out of order. This reduces errors and speeds up the plotting immensely; the operator can plot more quickly knowing that if he makes a mistake it will be caught and corrected by the machine.

The second major advantage of the Oxford on-line plotting system over its commercial counterparts is its low cost. Provided a user has access to a computer running ALGOL or FORTRAN (such as a PDP-8, Digital Equipment Corporation) which is equipped with analog-

digital (A-D) conversion, he can build the entire system for under \$50, which is about 1/200 the cost of most commercially available plotters.

Only the basic principles of the system are described here, since prospective users' requirements vary and anyone with reasonable programming experience will be able to write a program to suit his needs. (A "bare-bones" version of the program is presented in Appendix A.)

The flexibility of the machine is demonstrated by the fact that it has already been used to analyze red fox (*Vulpes vulpes*) radio tracking data, rodent grooming movements, visual fixation by doves, and, in its most complicated form, three-dimensional position and spacing within fish schools. Even in cases where x and y coordinates are readily determined, using the plotter is still much faster and more accurate than typing the coordinates by hand. The applications of the machine are not limited to movement data or serial analysis. For example, it has been used in a simple form to plot positions of fox scats on maps of their territories, and is even suited for analysis from pen-recorder tracings. (For this, the y coordinate corresponds to time and the x coordinate to pen displacement.)

PRINCIPLES

The Plotter

The film plotter consists of a horizontal table supporting a ground-glass or Plexiglas screen. A mirror mounted underneath allows film to be back-projected onto the screen (Figure 1a). When many points must be plotted in each frame and their order is important, accuracy is increased by placing counters (we use numbered glass overslips) over the corresponding locations. A hinged sheet of clear Plexiglas covers the screen and holds the counters in place.

At the head of the screen, two 10-K 10-turn linear potentiometers (Radio Spares, U.K. Ltd.) are mounted a known distance apart and connected to a 10-V power supply (Behaviour Research Development Ltd.). Aluminium spindles with a circumference just over 1/10 the

We thank Dr. Richard Dawkins for assistance throughout the development of the system and Mr. T. W. Harper for designing and building the electronics. Jon Erichsen criticized the manuscript. J. M. Cullen is now at the Department of Zoology, Monash University, Clayton, Victoria, Australia 3168.

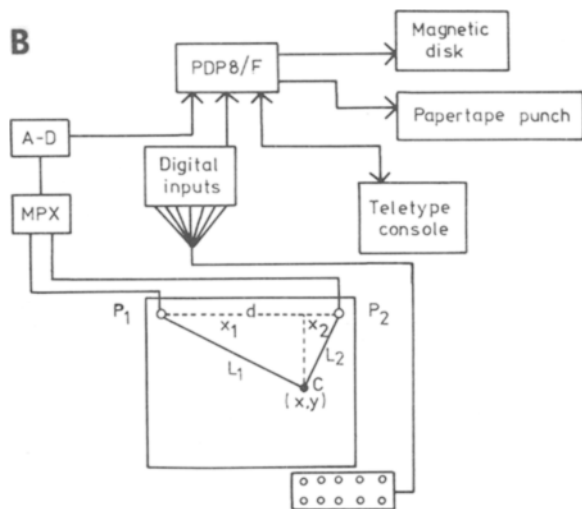
A**B**

Figure 1. (a) The coordinate plotter in use. Weights hanging at the ends of the strings attached to the cursor keep them taut against the potentiometers. (b) Schematic representation of the plotting system. MPX = multiplexor, A-D = analog-to-digital converter, and P_1 and P_2 = potentiometers.

diagonal of the screen are fitted over the potentiometers. The potentiometers, which should be further apart than the width of the screen, are connected to a cursor by lengths of thin inelastic string, forming two sides of a triangle. The strings are wrapped around the potentiometers and the free ends tied to 500-g weights. Output from the potentiometers is linked via a multiplexor to an A-D converter (Cambridge Instruments Ltd.) which is on-line to the computer (in our case, a PDP-8/E).

The exact hardware is not important, since the objective is not the absolute values of the potentiometers or

the power supply range (and, in fact, we have used several different supplies on different computers); rather, the objective is to choose the combination that will produce the full allowable range in the A-D input, thus giving the system maximum resolution.

So long as the string is not allowed to become greasy, little or no slippage occurs. Since all coordinates are subtracted from a reference point in each frame, the relative coordinates will be correct as long as little slippage occurs within a single frame. (In plotting 100 frames of 50 points/frame, slippage on our plotter is rarely as much as $\frac{1}{2}$ in., that is, $1/200$ in. per frame.)

Ten pushbuttons are connected to a rack of digital switches which are also on-line. These are not absolutely necessary, however, since, if they are unavailable, a comparable system can be built around a Teletype, with different keys taking the place of the on/off switches.

Since the position of the two potentiometers is known and the voltages across them are proportional to the lengths of the sides of the triangle described by the two strings, simple geometry determines the coordinates of the cursor. If L_1 and L_2 are the lengths of the sides of the triangle and D is the distance between the potentiometers, as in Figure 1b, then

$$x_1^2 + y^2 = L_1^2 \text{ and } x_2^2 + y^2 = L_2^2.$$

Algebraic manipulation gives:

$$x = (L_1^2 - L_2^2 + D^2) / 2D$$

and

$$y = (L_1^2 - x^2)^{1/2}.$$

Systematic Errors

Discrepancies between calculated and true coordinates may be due to inaccuracy of the plotting system or to parallax and systematic distortion caused by the lenses of the camera and projector. The system can be checked for accuracy by plotting points on a piece of graph paper and then comparing the calculated points with the original grid. Precision can be checked by plotting the same two points many times in succession and measuring the variation in the coordinates plotted. Using such tests, our system gives an accuracy of better than $\frac{1}{2}\%$; repeatability is similarly good. This inaccuracy is less than the human error made when plotting the points at high speed.

Errors due to parallax or lens distortion can also be overcome. In the most complicated version of the plotting program (where three-dimensional coordinates of fish are calculated), an iterative geometrical procedure corrects for parallax. Lens distortion, which provides the major source of error in our system, can be overcome by calibration. This is done most easily by plotting film in which true locations of many points are known and then calculating a "fudge factor" by multiple regression to transform the calculated coordinates into true coordi-

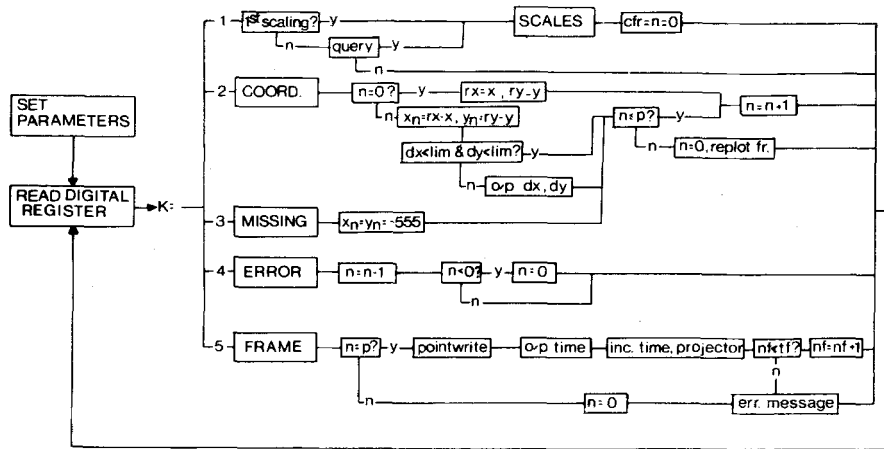


Figure 2. Flow chart of the basic plotting program. P = expected number of points per frame, TF = expected number of frames in the sequence, N = number of points plotted in current frame, NF = current frame number, x, y = coordinates of cursor, rx, ry = coordinates of reference point, and x_n, y_n = coordinates of n^{th} point in current frame.

nates. The accuracy of the plotting system is also increased, since the procedure would correct, for example, nonlinearity of the potentiometers.

The Program

Machine-code instructions for reading the A-D converters and checking the switch registers are written as ALGOL input/output device specifications [ALGOL is available for PDP-8s as ROGALGOL, R.H.A. (Minisystems) Ltd., 83 Gidley Way, Horspath, Oxford, England].

The program basically consists of a loop with the computer waiting until the operator pushes one of the digital switches in front of him, whereupon it enters the procedure specified by the switch. (A flow chart of the basic procedures is shown in Figure 2.) For instance, the operator might push "point," in which case the computer would determine the current location of the cursor; or he might push "frame," in which case the computer would check that the right number of points had been plotted, output the data from the current frame in a specified format, advance the projector, type out the time of the next frame, and offer the operator the opportunity to begin plotting again.

Whether the progress can be run simultaneously with others in a timesharing system depends on the hardware being used and the amount of on-line computation being undertaken. The only limit to the number of individual procedures that can be specified is the number of digital switches available. Of course, each switch or button can be given several meanings depending upon context. In this way, the same button might mean "plot point" or "continue," depending upon whether or not an error had been flagged by the computer.

In its simplest form, the program consists of five

procedures. (1) Scales: calibrates the voltages across the potentiometers to distance moved by the cursor, scales the coordinates to correct for the size of the projected image, and initiates a dialogue to define various parameters. (2) Point: determines the current cursor position and subtracts the coordinates from a reference point in the picture, since no projector positions each frame in exactly the same place. (3) Error: deletes the previous point plotted by decrementing the counters. (4) Missing: outputs characteristic x, y coordinates for missing data. Our program outputs -555, -555 (numbers outside the possible range of x, y values) so that the coordinates are left out of further analysis. (5) Frame: outputs the current frame of data to magnetic disk or paper tape, writes the frame number and current time (calibrated to original film speed), advances the projector, and increments the frame counter.

Interactive Features

The degree to which the program interacts with the operator depends, of course, on the aims of the user. For any particular plotting task, operators quickly discover the types of errors they are likely to make and corrective procedures can be added to the program. For instance, after once plotting for 3 h without turning on the power supply, we added a procedure to warn us at once, should we forget again. In most applications, however, the following procedures from our current system will be helpful.

In the initial dialogue with the program (scales), the operator sets a number of parameters so that the computer knows the number of frames in the film sequence to be plotted, how many points to expect per frame, the starting time of the sequence, and the interframe interval in 1/100s of seconds. Plotting too many points,

or too few, results in an error message and the invitation to replot the frame. Repeated errors result in increasingly abusive messages from the computer.

The order in which points are plotted is also checked. The operator sets criteria which correspond to the maximum distance that a point is likely to move from one frame to the next. Any point that is more than the criterion from its location in the previous frame is flagged as an error, and the operator is given the option of replotting it. Maddeningly, the computer is usually right when it suggests, on the basis of these criteria, that points have been plotted out of order.

Since the program is much faster than the operator,

it can be made to do a great deal of computation without slowing the plotting appreciably. In one version of the program, radio tracking fixes on free-ranging foxes and the recorded compass bearings are plotted; the computer calculates and outputs the actual position of the fox on a map and determines its velocity and heading at each interval. The current version of the fish plotting program is even more complicated. It calculates three-dimensional coordinates from the positions of fish and their shadows (Cullen, Shaw, & Baldwin, 1965), corrects for lens distortion and parallax, and carries out a moving average of z coordinates before outputting the data.

APPENDIX A

```
'BEGIN'
'COMMENT' OXFORD LOW-COST INTERACTIVE PLOTTING MACHINE
PROGRAM. 1 JULY 1977. VERSION 3 (BASIC PROCEDURES ONLY)
B.L. PARTRIDGE & J.M. CULLEN, DEPT. OF EXPTL. PSYCHOLOGY, OXFORD.

N.B. INPUT/OUTPUT FEATURES OF ROGALGOL:
CHANNEL 1=TELETYPE,2=NON-PARITY PAPER TAPE,
3=MAGNETIC DISK,4=PARITY PAPER TAPE,
5=DIGITAL SWITCH REGISTER,6=POTENTIOMETER #1,
7=POTENTIOMETER #2. I.E. A CALL OF CHIN(7) DELIVERS
THE CURRENT VOLTAGE OF POTENTIOMETER #2.;

'ARRAY' PL[1:2],P[1:2],DATA[1:50,1:4],SCL[1:2,1:2],SM[1:2];
'BOOLEAN' FRAME,ERR,AG,FSCAL;
'INTEGER' A,BT,OUT,I,J,K,L,NF,TF,CN,G,CFR,N;
'REAL' X,X0,Y,Y0,D,REFX,REFY,
SC,CS,TIME,IFI,DC,DP,SI,MINS,SECS,T;

'PROCEDURE' SCALES;
'BEGIN' 'COMMENT' SCALE PLOTTING COORD. SYSTEM TO REAL SPACE-
OPERATOR PLOTS POINT MIDWAY BETWEEN POTENTIOMETERS, THEN POINTS
AT PLUS AND MINUS 5CM., FOLLOWED BY A POINT AT THE DESIRED ORIGIN
AND A POINT A KNOWN DISTANCE (IN REAL COORDINATES) AWAY.;
'IF' NOT FSCAL 'THEN' 'BEGIN' TEXT(1," ARE YOU SURE? ");
'IF' CHIN(1) #& Y 'THEN' 'GOTO' NOSC;'END';
SC:=0;NF:=0;PL[1]:=CHIN(6);PL[2]:=CHIN(7);
'FOR' A:=1,2 'DO' 'BEGIN'
SAMP1: K:=CHIN(5);'IF' K=2 'THEN'
'BEGIN' SCL[A,1]:=CHIN(6);SCL[A,2]:=CHIN(7);'END'
'ELSE' 'GOTO' SAMP1;'END';
'FOR' A:=1,2 'DO' SC:=SC+ABS(SCL[1,A]-SCL[2,A]);
SC:=SC/10;
'COMMENT' SC=VOLTAGE CHANGE CORESP TO ONE CM;
'FOR' A:=1,2 'DO' 'BEGIN'
SAMP2: K:=CHIN(5);'IF' K=2 'THEN' 'BEGIN'
COORDINATES;'IF' A=1 'THEN' 'BEGIN' X0:=X;Y0:=Y;'END'
'ELSE' CS:=SQRT((X-X0)**2+(Y-Y0)**2)/100;
'END' 'ELSE' 'GOTO' SAMP2;'END';
'COMMENT' X0,Y0 = COORDINATES OF ORIGIN. CS=VOLTAGE CHANGE CORRESPONDING
TO 1 CM IN TANK. 100=DIST OF REFERENCE POINT FROM ORIGIN, IN
REAL SPACE;
SKIP(1);TEXT(1,"SC=");
RWRITE(OUT,SI,10,4);TEXT(OUT," ");
RWRITE(1,SC,10,4);RWRITE(OUT,TF,6,0);RWRITE(OUT,N,6,0);
SKIP(1);SKIP(OUT);BLANKTAPE(1);
CN:=0;NOSC:FSCAL:='FALSE';
'END' SCALES;

'PROCEDURE' COORDINATES;
'BEGIN' 'COMMENT' DETERMINES X,Y COORDS OF CURSOR;
P[1]:=CHIN(6);P[2]:=CHIN(7);
```

```

FRAME:='FALSE';
'FOR'I:=1,2'DO'P[I]:=D/2+(P[I]-PL[I])/SC;
X:=(D**2+P[1]**2-P[2]**2)/(2*D);
Y:=SQRT(P[1]**2-X**2);
'END' COORDINATES

'PROCEDURE'ERROR;
'BEGIN' 'COMMENT' BACKS-UP ONE POINT FOR EACH TIME PUSHED;
CN:='IF'CN>0'THEN'CN-1'ELSE'0;
TEXT(1,'ERR ');
'END' ERROR;

'PROC' MISSING;
'BEGIN' 'COMMENT' O/P CHARAC. DATA FOR MISSING PTS.;
DATA[CN,1]:=-555;DATA[CN,2]:=-555;
TEXT(1,'M');CN:=CN+1;SUBREF;
'END' MISSING;

'PROCEDURE'CHECK;
'BEGIN' 'COMMENT' CHECKS THAT CORRECT # PTS. HAVE BEEN PLOTTED;
ERR:='FALSE';
'IF'CN=N+1'THEN'GOTO'ZZZ;
'IF'AG'THEN'TEXT(1,'IDIOT! AGAIN ');
TEXT(1,' TOO FEW PTS.-REPLOTT FRAME! ');SKIP(1);
AG:='TRUE';ERR:='TRUE';
ZZZ:CN:=0;
'IF'NOT'ERR'THEN'AG:='FALSE';
'END'CHECK;

'PROCEDURE'TIMEOUT;
'BEGIN' 'COMMENT' PROGRAM KEEPS TIME IN SS.HH, BUT OUTPUT
TO TELETYPE IS IN MM SS.HH FOR COMPARISON WITH VIDEO CLOCK;
MINS:=ENTIER(TIME/60);SEC:=TIME-(MINS*60);
'END'TIMEOUT;

'PROCEDURE'PRETIME;
'BEGIN' 'COMMENT' OUTPUTS CURRENT FRAME'S TIME IN SS.HH, INCREMENTS
CURRENT TIME BY IFI;
SKIP(OUT);RWRITE(OUT,TIME,8,2);TEXT(OUT," , ");
TIME:=TIME+IFI;
'END'PRETIME;

'PROCEDURE'POINTWRITE;
'BEGIN' 'COMMENT' PROVIDE CHARAC. DATA FOR MISSING PTS. AND
OUTPUT IN SPECIFIED FORMAT;
'INTEGER'J,LF;LF:=2;
'FOR'I:=N+1'STEP'1'UNTIL'TF'DO'
'FOR'J:=1,2'DO'DATA[I,J]:=-555;SKIP(OUT);
'FOR'I:=1'STEP'1'UNTIL'TF'DO'BEGIN'
'FOR'J:=1,2'DO'RWRITE(OUT,DATA[I,J],9,2);
LF:='IF'LF=1'THEN'2'ELSE'1;
'IF'LF=2'THEN'SKIP(OUT);
'END';LASTPT;
'END'POINTWRITE;

'PROCEDURE'NINTERVAL;
'BEGIN' TEXT(1,' NEW IFI= ');IFI:=READ(1);SKIP(1);
'END'NINTERVAL;

'PROCEDURE'BACKUP;
'BEGIN' 'COMMENT' ALLOWS FRAMES TO BE REPEATED OR PLOTTING TO
BE STARTED MID-SEQUENCE;SKIP(1);TEXT(1,' BACKUP TO FR# ');
NF:=READ(1);TEXT(1,' NUMBER OF FISH? ');N:=READ(1);
CN:=0;TEXT(1,' TIME= ');INTIME:NINTERVAL;
'END'BACKUP;

'PROCEDURE'INTIME;
'BEGIN' 'COMMENT' READS TIME FROM TTY IN MM SS.HH,
CONVERTS TO SS.HH;
TIME:=READ(1)*60+READ(1);
'END'INTIME;

```

```

'PROC' BLANK TAPE(N);'VALUE'N;'INTEGER'N;
'BEGIN'INTEGER'J,K;
    'FOR'K:=1'STEP'1'UNTIL'N'DO'
    'FOR'J:=1'ST'1'UN'50'DO'CHOUT(OUT,128);SKIP(OUT);
'END' BLANK TAPE;

'PROCEDURE'COMCORD;
'BEGIN'REAL'I;'INTEGER'A;
'COMMENT' COMPARE COORD WITH LAST FR. IF IFF COORD IN LAST FR WASN'T
    -555 (I.E. MISSING);
    'IF'NF>1'THEN'BEGIN'IF'DATA[CN,3] #-555'AND'
        DATA[CN,4] #-555'THEN'
    'BEGIN'FOR'I:=1,2'DO'
    'IF'ABS(DATA[CN,I+2] - DATA[CN,I])>SM[I]
        'THEN'GOTO'OUTPUT;GOTO'CSKIP;
OUTPUT:SKIP(1);TEXT(1," PT# ");WRITE(1,CN);
TEXT(1," DX,DY = ");'COMMENT'OUTPUTS DIFF. FROM LAST FR.;
'FOR'I:=1,2'DO'RWRITE(1,DATA[CN,I+2] - DATA[CN,I],10,2);
CSKIP:'END';'END';
'END'COMCORD;

'PROCEDURE'LASTPT;
'BEGIN'    'COMMENT' UPDATE LAST PT. COORDS.;
    'INTEGER'I,J;
    'FOR'I:=1'STEP'1'UNTIL'N'DO'FOR'J:=1,2'DO'
        DATA[I,J+2]:=DATA[I,J];
'END'LASTPT;

'PROCEDURE'SUBREF;
'BEGIN'    'COMMENT' SUBTRACT PTS. FROM REFERENCE PTS.;
    DATA[CN,1] := 'IF'DATA[CN,1] #-555'THEN'REFX-DATA[CN,1]
        'ELSE'-555;
    DATA[CN,2] := 'IF'DATA[CN,2] #-555'THEN'REFY-DATA[CN,2]
        'ELSE'-555;
    WRITE(1,CN);CN:=CN+1;'IF'CN>N+1'THEN'BEGIN'CN:=0;
    TEXT(1," TOO MANY POINTS! TRY FRAME AGAIN. ");
    SKIP(1);'END';
'END'SUBREF;

'COMMENT'*****
*****
----- MAIN BODY OF PROGRAM -----;
START:D:=48.4;FRAME:='FALSE';SC:=1;NF:=0;CN:=0;CS:=1;
'COMMENT' D EQUALS DISTANCE BETWEEN POTENTIOMETERS;
CFR:=0;AG:='FALSE';FSCAL:='TRUE';
TEXT(1," THIS IS THE OXFORD INTERACTIVE PLOTTER ");SKIP(1);
TEXT(1," VERSION 3 (BASIC PROCEDURES ONLY) ");SKIP(1);

'FOR'I:=1'STEP'1'UNTIL'50'DO'BEGIN'
    'FOR'J:=1'STEP'1'UNTIL'4'DO'DATA[I,J] := -555;'END';

TEXT(1," DATE & COMMENTS: ");
    'COMMENT' IGNORES ALL TEXT UNTIL SEMICOLON;
    REPT:I=CHIN(1);'IF'I#187'THEN'GOTO'REPT;SKIP(1);
    SKIP(1);TEXT(1,"OUTPUT CHANNEL? ");OUT:= READ(1);
'COMMENT' 1=TTY,2=NON-PARITY PAPER TAPE,3= DISK,
    4=PARITY PAPER TAPE;
'IF'OUT=3'THEN'DISK(2);
TEXT(1,"SEQUENCE # ");SI:=READ(1);SKIP(1);
TEXT(1," STARTING # OF PTS/FRAME ? ");N:=READ(1);
TEXT(1," TOTAL # PTS/FRAME? ");TF:=READ(1);SKIP(1);
TEXT(1," STARTING TIME (MM SS.HH) ? ");INTIME;
TEXT(1," INTERFRAME INTERVAL? ");IFI:=READ(1);
SKIP(1);TEXT(1," MAX. DELTA-X/FRAME? ");SM[1]:=READ(1);
TEXT(1," MAX. DELTA-Y/FRAME? ");SM[2]:=READ(1);SKIP(1);
TEXT(1," READY FOR SCALING ");SKIP(1);
AGAIN:K:=CHIN(5);'COMMENT'MAIN LOOP OF PROGRAM!!
'IF'K=2'THEN'
    'BEGIN' COORDINATES; X:=(X0-X)/CS; Y:=(Y0-Y)/CS;
    'IF'CN=0'THEN'BEGIN'REFX:=X;REFY:=Y;

```

```

      CN:=1;'END"ELSE'
      'BEGIN'DATA[CN,1]:=X;DATA[CN,2]:=Y;SUBREF;'END';
      'END"ELSE'
'IF'K=4'THEN' 'BEGIN'ERROR;FRAME:='FALSE' 'END"ELSE'
'IF'K=8'THEN' 'BEGIN'MISSING; FRAME:='FALSE'; 'END' 'ELSE'
'IF'K=16'THEN' 'BEGIN'BLANK TAPE(3); 'END' 'ELSE'
'IF'K=32'THEN' 'BEGIN' SCALES;FRAME:='FALSE';'END' 'ELSE'
'IF'K=64'THEN' 'BEGIN'
      'IF' 'NOT' FRAME 'THEN' 'BEGIN' CHECK;
      'IF'NOT'ERR'THEN'BEGIN'POINTWRITE;PRESTIME;
      SKIP(1);TEXT(1," NT= ");TIMEOUT;
      NF:=NF+1;RWRITE(OUT,NF,4,0);RWRITE(1,NF,4,0);
      SKIP(OUT);FRAME:='TRUE';CN:=0;
      CFR:='IF'CFR=10'THEN'0'ELSE'CFR+1;
      'IF'CFR=0'THEN'BLANKTAPE(3);
      'END"END"END'
'ELSE"IF'K=128'THEN'NINTERVAL
'ELSE"IF'K=256'THEN'BACKUP
'ELSE"IF'K=512'THEN'BEGIN'
      SKIP(1);TEXT(1,"QUERY DATA, FR# ");RWRITE(1,NF+1,4,0);
      SKIP(1);'END'
'ELSE"IF'K=1024'THEN'BEGIN'
      TEXT(1," NEW TIME= ");INTIME;
      'END' 'ELSE' 'GOTO'AGAIN;
'GOTO'AGAIN;

'END'
$

```

REFERENCE

CULLEN, J. M., SHAW, E., & BALDWIN, H. Methods for measuring three-dimensional structure of fish schools. *Animal Behaviour*, 1965, **13**, 534-543.

(Received for publication June 7, 1977;
revision accepted July 26, 1977.)