Trans. Inst. Indian Geographers
Vol.9.No.1, January 1987

A "BASIC" PROGRAM FOR DRAWING MAPS FROM A GRAPH PAPER DATA BASE

Robert B. Begg, Pennsylvania.

Abstract : In some cases,departments of geography will own a microcomputer with the capacity to support BASIC, but will not be able to afford or obtain the digitizing hardware or mapping software to construct computer generated maps of their own choosing. This article presents a simple BASIC program that, with a graph paper data base, can be used to create an outline map for any geographic area.

Introduction

As Indian departments of geography and planning move towards computerized mapping, they may experience a frustrating period during which no digitizer and no mapping software are available to them for the creation of outline maps. This article presents a simple BASIC program which will generate such maps using only a graph paper data use.

The program is described in three parts. First, the BASIC program is presented in its entirety. The logic and structure of the program are explained. Even a non-programmer should have no difficulty understanding and using the program. Second, the use of graph paper to construct an appropriate boundary file is described with a simple example. Finally, the correct organization of the data file is demonstrated. The information found in this article should permit anyone with Advanced BASIC software to prepare an outline map on their microcomputer with very little time and effort.

THE "BASIC" PROGRAM

The Computer program seen below is composed of two segments. The first segment, the input module, defines the variables to be used, sets some initial parameters for the program, reads in data from an external file and performs some simple manipulations on the data. The second segment, the mapping module, consist of a call statement (GOSUB) in the main program that invokes a single subroutine beginning at line 380. It is the subroutine which actually performs the work of drawing the map. Because the call statement and subroutine are the core of the program they are described first.

DRAWING THE MAP

The actual mapping done in the BASIC program, is accomplished by two commands. All of the drawing seen on the screen is the result of the DRAW and LINE commands found in lines 460 and 470.

The first command, DRAW, simply moves the pointer to a particular location on the screen. In this program, the map is divided into a series of boundaries. The coordinates which form the boundary line are called a boundary file. Each boundary is a continuous line. The map itself may consist of many such boundaries, e.g. nation, state, district, tehsil or a river basin. Boundaries may or may not share line segments. If boundaries share line segments in this program, the line segments are only drawn once. Each time a line segment is drawn, the program invokes the DRAW command to move the stylus to an appropriate point from which a continuous line is drawn, completing a boundary or boundary segment.

In order to do this, the DRAW command contains the letter'B' that means "move without

"move to the following location". The letters D1 and D2 designate the x- and y-coordinates respectively to which the DRAW command moves the stylus. The DRAW command is invoked only when a line segment is begun. The function of the if-then statement in line 460 is to guarantee that the command is ignored at other times.
The LINE command, line 470, does all the actual drawing. The dash just before the word LINE tell the computer to draw a line from the previous location of the stylus on the screen to the x-and y-coordinates found in parentheses. Since the DRAW command has given the computer a starting location, the first line segment will be drawn from the DRAW coordinate to the first LINE coordinate. After this first step is completed, the LINE command is repeated until the boundary file coordinates for the boundary or boundary segment have all been read and drawn.

To accomplish this, the LINE command is placed inside "loop", a command that is repeated for a specified number of times. In this case the 'loop' is executed by drawing consecutive and contiguous line segments until the first boundary is completed. The logic is simple : draw a line segment from A to B, draw a line segment from B to C, etc., until A is reached again. Once the points for any boundary have been drawn, the subroutine return to the main program.

Once back in the main program, Line 370, the computer encounters the commands that sent it to the subroutine in the first place. the command to GOSUB, "go to the subroutine", is itself within a loop, the program comes back to the subroutine to draw another boundary and so on, until the specified number of boundaries have been drawn. Once this is done the program passes to the next command END and the program terminates.

To summarize, the mapping subroutine is repeated once for each separate boundary file. If the map were India, Maharashtra,Pune district, three boundaries would have been drawn. The subroutine would be executed three times. The first command, DRAW, places the screen stylus at some location on the boundary, from where the second command, LINE, draws successive line segments until the initial point is returned to or final point reached. At the completion of each boundary, the main program is re-entered so that the process may begin again. When all boundaries are drawn the program ends.

DATA INPUT

The input module defines a series of parameters and reads in data from an external file. The module begins by defining two parameters and identifying the same of the data file. The first three pieces of information are entered as responses to queries from the computer while the program is running. While the interactive nature of the program may be cumbersome for someone wishing to draw the same map repeatedly, it provides generality and flexibility in the program.

The first parameter, N,the number of boundaries, is the number of separate line segment or the number of times which the stylus must be placed to begin drawing a separate line segment. The user will see the command "ENTER NUMBER OF BOUNDARIES" on the screen and simply types in the correct response and strikes "return". The second parameter requested by the program is the total number of coordinates for all boundaries, P. This will be the number of x,y values in the boundary file. It does not include the initial point at which the program begins drawing each boundary. Finally, the program will ask for the name of the data file in which boundary points and other data are stored.

With this information the program is able to begin reading in the data. There are five variable comprising three different kinds of information. The first two variables, x and y, are arrays which contain the x- and y- coordinates of the boundaries. The values in this arrays are read by the LINE command. The next two variables, P1 and P2, are the coordinates which place the stylus to begin drawing the boundary file. These arrays are the coordinates read by the DRAW command. Finally the variable L, is an array which tells the computer the number of coordinates in each boundary file. If it takes 46 x-, y- coordinates to draw the border of India, The L array would contain the number 46. In this way the computer knows just how many times to execute the LINE loop in this subroutine to complete the first boundary. The number of P1, P2 and L variables is equal to the number of boundaries to be drawn. The number of x and y variables corresponds to the number of coordinates in all boundaries.

The information provided in the data input segment is now used by the call statement and subroutine to actually draw a map on the screen. Once the interactive parameters are given, the map should be drawn quickly and accurately before the user's eyes. The next two sections tell the reader how to construct a data base from graph paper that can be used by this program and how to organise that data base so that the computer can read it.

CREATING A DATA BASE

The key to creating the data base for an outline map is the organization of the computer monitors screen. Depending on the resolution of the monitor, medium or high resolution, your screen will consist of a fixed number of x-, y- coordinates called pixels. On a medium resolution graphics monitor there will be 320 horizontal (x) points and 200 vertical (y) points. On a high resolution graphics monitor there are 640 horizontal (x) coordinates and 200 vertical (y) coordinates. Each of the coordinates formed by the intersection of these points can be addressed by the computer : read from or written to. The numbering system begins with the coordinates 0,0 in the upper left hand corner of the screen and, for the medium resolution screen, ends with the coordinate 319, 199 in the lower right hand corner (fig.1). The coordinate system is thus inverted if compared to customary x-, y- coordinates.

In order to inform the computer what dimensions you are anticipating, the SCREEN command is used. There are many options for the SCREEN command, but the one that is of importance to this mapping program is the first option. SCREEN 1 indicates the medium resolution graphics screen, SCREEN 2 the high resolution graphics screen. While either may be used, the translation from graph paper to the screen is not straightforward in either case. Under normal circumstances the screen has an "aspect ratio" that determines the relationship of the distance between x and y points. The standard aspect causes the horizontal axis to be 4/3 as long as the vertical axis. Now in medium resolution, there are 320 horizontal points to 200 vertical points. This means that if the aspect ratio were 1/1 there would be 8 horizontal points for each vertical point. Since the standard aspect is 4/3 there are, in fact, 24 horizontal to 20 vertical points.

In lines 290 - 340 the coordinates are adjusted to account for the difference in scales of the horizontal and vertical axis. The vertical axis is, in fact, shrunk to correspond to the horizontal axis. This permits the user to use true coordinates on the graph paper in the section below without distorting the map when it appears on the screen. To make a similar correction for high resolution graphics, the ratio 48/20.

The example given in Fig.1 assumes a medium resolution graphics screen. It should make the process of data base construction clear. The user should begin by constructing a grid field on the graph paper that corresponds to the given dimensions of the screen. Since the horizontal axis is represented as longer than the vertical this is most easily accomplished by turning the graph paper so that the horizontal axis is the longer of the two and beginning the numbering with 0, 0 in the upper left hand corner. Next, the easiest procedure is to find or have reduced on a xerox machine, a map that fits well within the boundaries of the graph.

The map, for example one of India or Africa, may be turned sideways to conserve space. It will, however, then appear sideways on the screen. Having found a map of appropriate dimensions, the user then traces the map onto the graph paper. From this graph the user will simply identify and mark the points to be plotted.

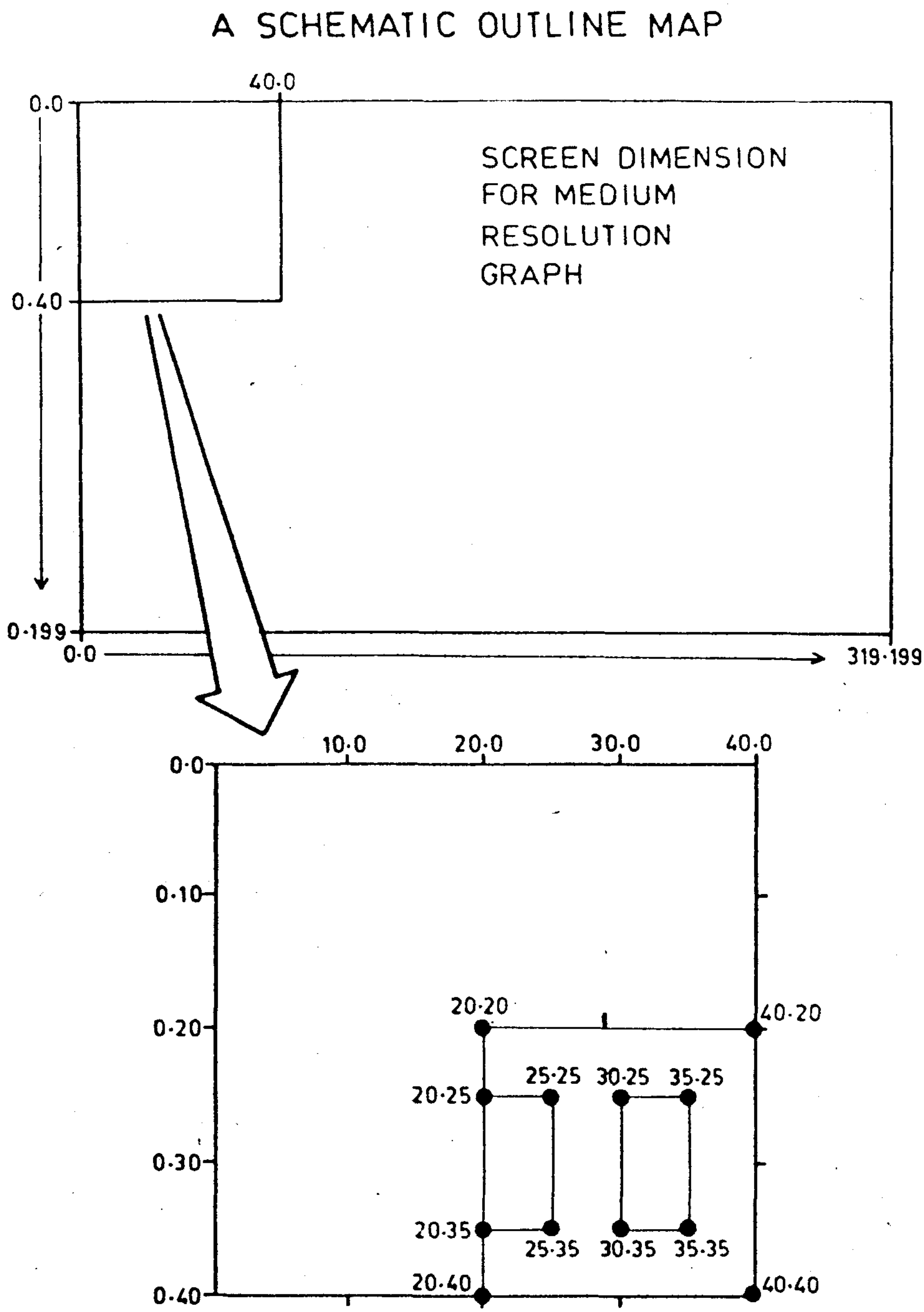

A SCHEMATIC OUTLINE MAP
40·0
0·0
SCREEN DIMENSION
FOR MEDIUM
RESOLUTION
GRAPH
0·40
0·199
0·0
319·199
10·0
20·0
30·0
40·0
0·0
0·10
0·20
0·30
0·40
20·20
40·20
20·25
25·25
30·25
35·25
20·35
25·35
30·35
35·35
20·40
40·40

Fig. 1

The coordinates of those points become the data base for the program.

In Fig.1 a simple schematic is drawn on graph paper and the points to be plotted identified. The picture will be used in the next section to make clear the process of constructing a data file for use by the computer.

CONSTRUCTING THE DATA FILE

Recall that the data file consists of five variables and three kinds of information. The picture above is a simple series of squares, but might well represent a country, and two political subdivisions with the country. One of the "political subdivisions' shares a boundary with the nation, one does not. The first thing to decide is the number of separate boundaries that will be drawn. In this case there are three.

This means that the response to the computers query concerning the number of boundaries will also be three. It also means that there will be three starting points and three numbers to indicate the size of a boundary coordinate set. In constructing the data file this information will come at he end of the data listing.

The first information to be entered in the boundary file is the boundary coordinates themselves. There are twelve coordinate points on the diagram. As will be seen, only eleven of them are actually necessary to complete the diagram. The organization of the entire data set is shown in Table 1.

Table 1 Organization of the Data File

```
     40,20
     40,40
     20,40
---  20,20
     25,25
     25,35      'boundary coordinates
     20,35      ' X,Y
---  (20,25)
     35,25
     35,35
     30,35
---  30,25
     20,20,4    'starting coordinates P1,P2
     20,25,3    'points per boundary L
     30,25,4
```

The data file looks just as it would if stored for processing by the BASIC program with three exceptions. The comments are placed to identify variables and would not be necessary. There are also three lines to mark key coordinates that would not appear in the real data file. Finally, the coordinate in parentheses is not necessary and would not, in fact, appear.

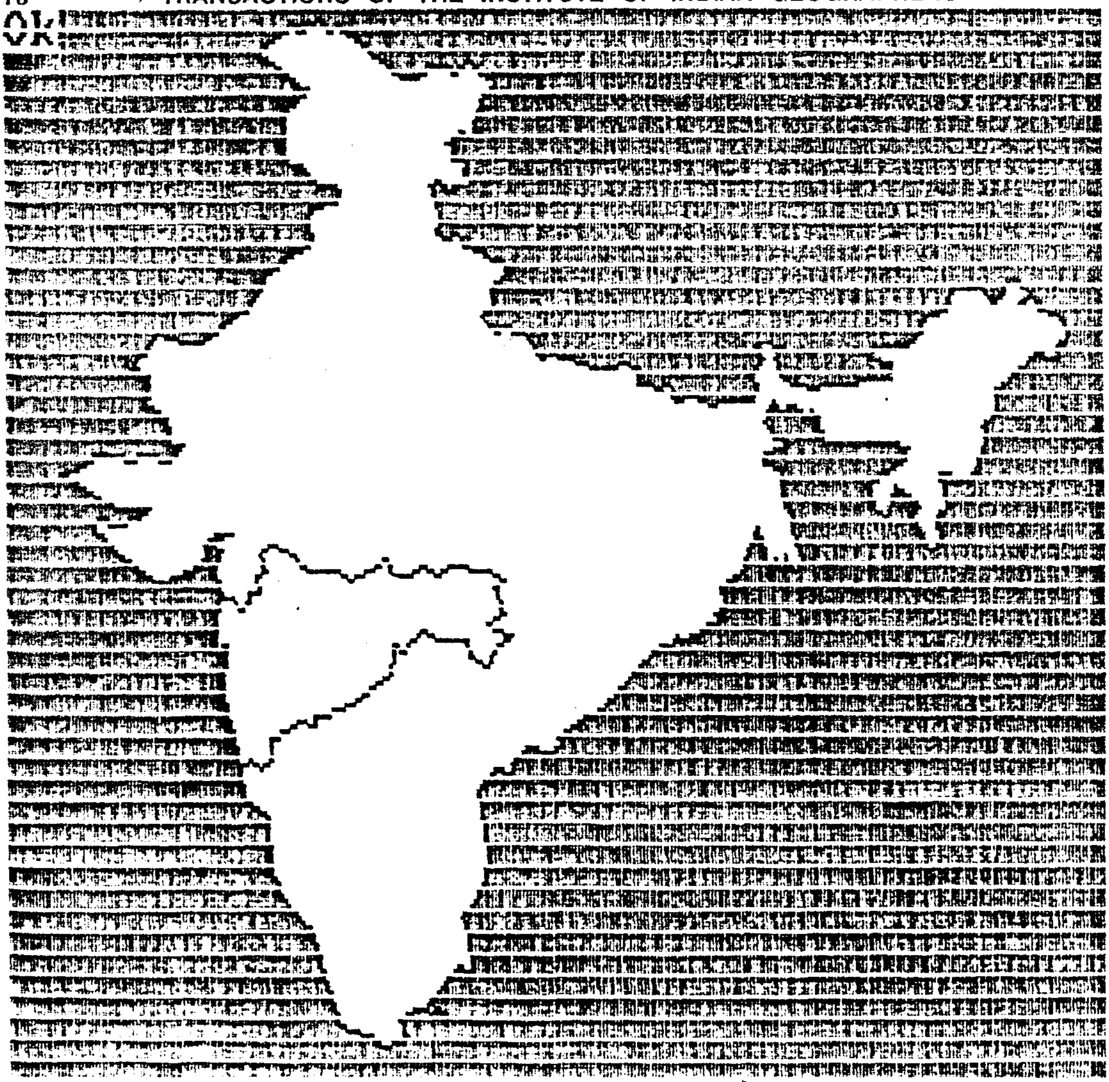
The first four coordinates draw the larger boundary. Looking at the last point, the fourth data entry, it is identical to the P1,P2 pair listed on the thirteenth line. This is because line thirteen instructs the computer where to begin drawing and the LINE command returns there to complete the picture. The same is true for the third boundary drawn.

The second boundary is a different case. Here, one side of the figure, the eastern edge, is shared by the nation or larger boundary. It is not necessary to draw it twice. Rather, the boundary begins with point 20,25 (line 14) and connects three consecutive points (here L is equal to 3) to end at coordinate 20,35. While this may seem a trivial difference in the simple figure, it will be quite important in complex maps that share many boundary segments.

To summarize, the boundary coordinates to be read by the LINE command are entered in the order they are to be drawn. The total number of such coordinates (here 11) is entered in response to the interactive query at the beginning of the program. Directly following these coordinates come the P1,P2 coordinates to be read by the DRAW command and the variable I which tells the computer how many points are to be read for each boundary segment. There is one line of such data for each boundary segment and they are entered in the order of the boundary coordinates.

CONCLUSION

The information above and the simple program appended to this article, should allow the reader to construct a boundary file for any nation and its political subdivisions. The arduous chore of plotting the coordinates is the most difficult



COMPUTER DRAWN OUTLINE MAP OF INDIA AND MAHARASHTRA

process, but even this is no more than a few hours work for most maps. The map of India with the outline of Maharashtra seen in Figure 2 was constructed in about three hours and consist of 350 boundary points.

In order to print maps such as the one shown here, the user must have a GRAPHICS.COM With this option the Print Screen command will reproduce he map appearing on the terminal on your dot matrix printer.

This is simple program. Its major intent is to provide an interim tool to departments without digitizing hardware or mapping software. The information contained in the Guide to Operations and the Basic Manual provided with most micro-computer systems should permit the user to modify and improve this program. The PAINT command might be an interesting place to start. What should be clear ;is that simple computer mapping with the BASIC language is neither difficult nor mysterious and is well within the reach of every geography department.

Appendix : BASIC Program MAPDRAW

```
10 SCREEN 1
20 CLS:KEY OFF
30 REM ****** DESCRIPTION MODULE ****
40 REM ***** NAME : MAPDRAW
50 REM ******* BASIC COMPUTER PRO-
GRAM
60 REM ******* WRITTEN BY ROBERT B.
BEGG
70 REM DATE 3 MARCH 1987
80 REM THE PROGRAMME DRAWS A MAP
(AND SUB REGIONS) FROM
COORDINATES
90 REM ******** INPUT MODULE *******
100 REM B=THE # OF SEPARATE BOUNDA-
RIES TO BE DRAWN (FOR EG.
INDIA, MAHARASHTRA, DIST)

110 REM P= THE TOTAL COORDINATES OB-
TAINED FROM GRAPH PAPER-ALL
BOUNDARIES
120 INPUT "ENTER NUMBER OF
BOUNDARIES";B
130 INPUT "ENTER TOTAL COORDINATES";P
140 REM X=X COORDINATE OR LAT. & Y=Y
COORDINATE OR LONG.
150 REM P1,P2=STARTING POINT FOR
BOUNDARY
160 REM L=# OF COORDINATES FOR EACH
BOUNDARY FILE
170 DIM X(P), Y(P), P1(B), P2(B), L(B)
180 INPUT "ENTER THE FILE NAME";A$
190 OPEN "I", #1,A$
200 FOR I=I TO P
210 INPUT #1, X(I), Y(I)
220 NEXT I
230 FOR K=1 TO B
240 INPUT #1, P1(K), P2(K), L(K)
250 NEXT K
260 N=1
270 M=0
280 K=1
290 FOR I=1 TO P
300 Y(I)=(20/24)*Y(I)
310 NEXT I
320 FOR I=1 TO N
330 P2(I)=(20/24)*P2(I)
340 NEXT I
350 REM ****** PROGRAM MODULE ******
360 CLS
370 FOR Q=1 TO B
380 GOSUB 410
390 NEXT Q
400 END
410 X1=P1(K)
420 X2=P2(K)
430 M=M+L(K)
440 K=K+1
450 FOR I=N TO M
460 IF I=N THEN DRAW"BM=X1;=X2
470 LINE-(X(I),Y(I)"
480 NEXT I
490 N=I
500 RETURN
```

REFERENCES

Guide to Operations, IBM.
Sixteen Bit BASIC, IBM.

Address of the Author :
Dr. Robert B. Begg
Dept. of Geography and Regional Planning IUP
Indiana, Pennsylvania 15705
USA