

REPRESENTATION
OF
GEOGRAPHIC
DATA
ON
THE
COMPUTER.

by

✓ G. NAGARAJAN 856113

guide

U. A. ATHAVANKAR.

L. D. C. Library

L. I. T. Bombay.

IDC

SPECIAL PROJECT

1987.

Representation Of Geographic Data
On The Computer

Special Project

by
G.Nagarajan

guide
U.A.Athavankar

I D. C. Library
L L T. Bombay.

Industrial Design Centre
Indian Institute of Technology
Bombay 400 076

APPROVAL SHEET

This Special Project entitled
'Representation of Geographical Data on the Computer'
undertaken by
G.Nagarajan, Roll No 856113
is approved
in partial fulfilment of
the requirements for the award of
the degree of
Master of Design
in
Product Design

GUIDE



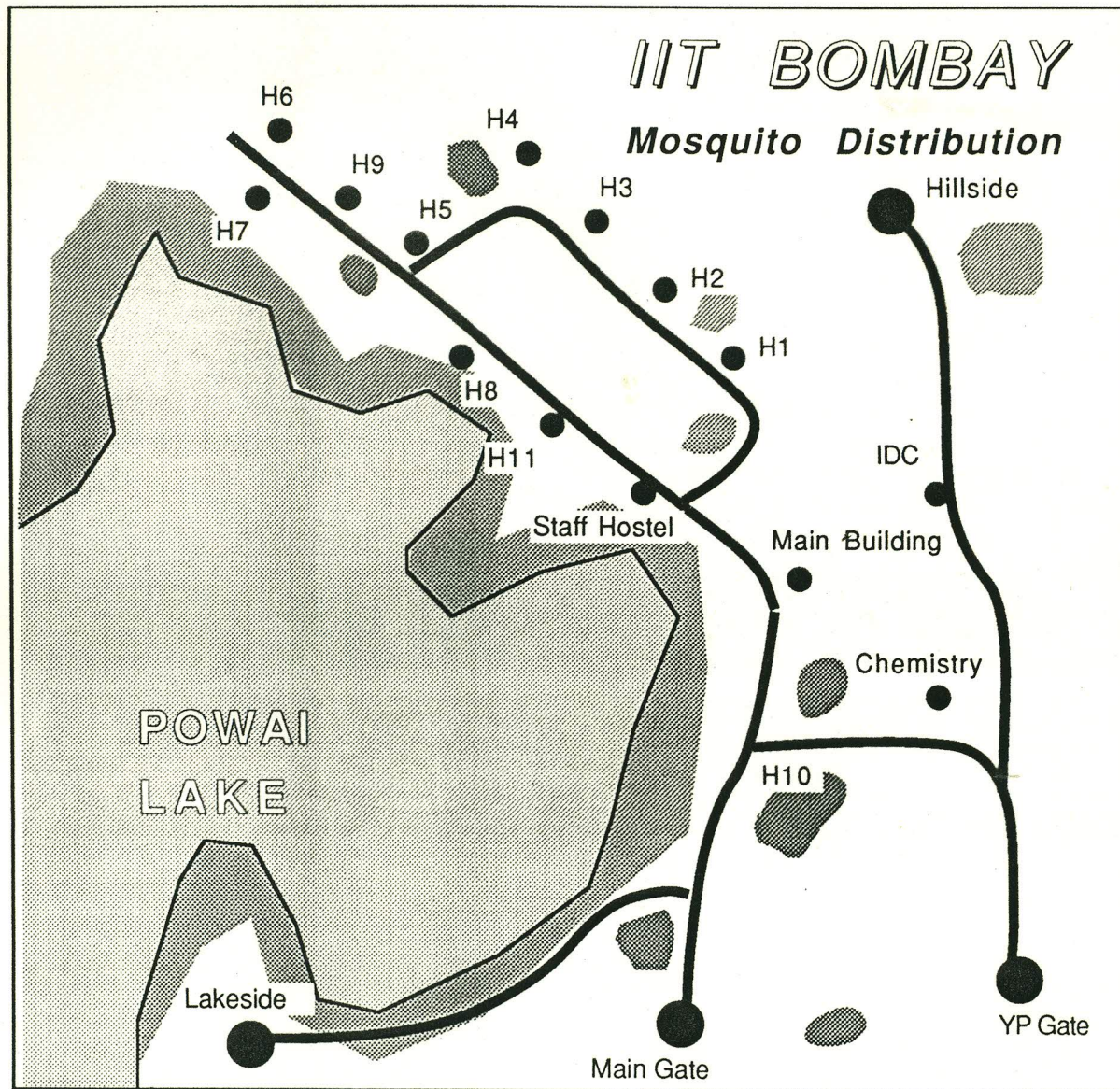
1
On
- The
Importance
Of
Maps

Geographically distributed parameters are easiest to interpret and understand when they are graphically represented (in the form of maps). When they are handled in terms of raw numbers, on the other hand they are much more difficult to assimilate or comprehend.

Take , for example, the (hypothetical) distribution of mosquitoes in IIT, Bombay (Figure 1). The first thing that strikes one is that there are high, localised concentrations in some areas and a relatively low distribution elsewhere. Then one notices that the area bordering the lake has a high concentration and this leads to the inference that the stagnant water there is the cause. The other high concentrations do not have an obvious explanation and further study may be needed to identify their causes. A few may turn out to be the result of open rubbish dumps, others may be due to a thick growth of bushes.

Thus maps besides giving an idea of the nature of a distribution can also give insight into the causes of various phenomena. This would not be possible (except for an expert) if the data were represented purely in the form of numbers and coordinates.

Figure 1



Maps are therefore ideally suited for use as a decision making tool particularly in situations which involve an analysis or understanding of geographically distributed parameters.

2
On
The
Types
Of
Maps

There are three basic types of maps in common use (excluding geographic reference maps such as navigation maps, topological maps etc.). These are

- Proportional symbol maps
- Point symbol maps
- Quantitative distribution maps

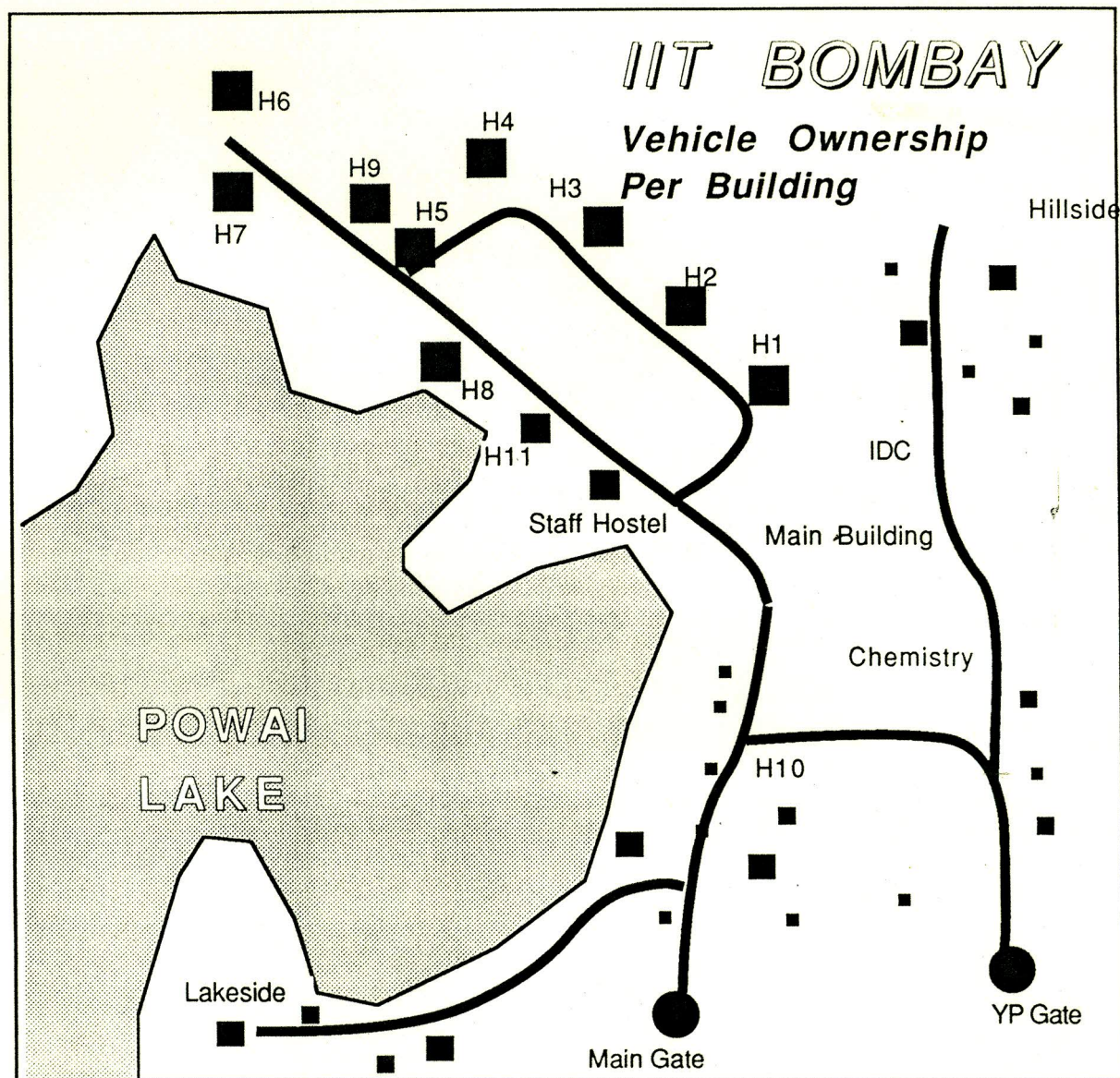
PROPORTIONAL SYMBOL MAPS

These maps have solid or outline geometric figures (such as squares, circles, triangles etc.) positioned as close as possible to the location they refer to. The size of the figure indicates the magnitude of the phenomenon under consideration at that particular location. Pictorial symbols may also be used on such maps. The major limitation of this kind of maps is the fact that it can be used meaningfully and effectively only for phenomena which are discretely distributed. An example of this kind of map is shown in figure 2 (a hypothetical map showing the vehicle ownership per building in IIT-Bombay).

POINT SYMBOL MAPS

Graphic symbols are used on these maps to indicate the existence of an item/object or the availability of a service at a particular location. The most commonly encountered example

Figure 2



of this type of map is the guide map which indicates through different symbols the locations of eating places, shopping centres, hotels, places of interest to tourists etc. These maps are not suitable for display of quantitative information.

QUANTITATIVE DISTRIBUTION MAPS

Basically, these maps use a variation of visual density to represent the quantitative variation of a parameter. The most commonly used technique is to use dots of a uniform size, the dots being located in such a way that the concentration of dots at any location is proportionate to the value of the phenomenon under consideration. An example of this kind of map is the mosquito concentration map discussed earlier. Both discretely and continuously distributed phenomena can be depicted on such maps.

Of the three types of maps described above, quantitative distribution maps are most suited for use as a decision making tool since they -

- give a good visual picture of the nature and magnitude of a distribution.
- can depict quantitative information
- are suitable for both discrete and continuous phenomena.

3
On
— The
Advantages
Of
Computer
Maps

UPDATEABILITY

Conventional maps are quite difficult to keep upto date. This is mainly because of the nature and complexity of the process by which they are produced. The production of a map involves collection of data sorting of data, conversion to a symbolic or graphic form, the preparation of the map and, ultimately, its reproduction. The use of computers promises to drastically simplify this process. One can imagine in this age of the computer, that the not-too-distant future could witness the following scenario :-

The data collector is armed with a portable computer which feeds him the questions to be asked. As he receives responses he enters them into the computer which takes these into account while feeding him the next question. The data entered is processed and stored within the computer and is later transferred to a larger, central computer where it is used to create a new map or update an existing one which could then be made available to the user either as soft copy (a display on a terminal) or hard copy (a print out).

COMPACT STORAGE

Electronic media (such as magnetic tape or floppy discs) allow

a very much higher information density than the paper media. The use of computers thus makes feasible the creation of a large data base of data in a very compact format. This data base could be periodically updated and accessed through various software packages which could give the user the precise kind of map that he is interested in.

FLEXIBILITY

The compact storage made possible through the use of computers allows a flexibility that cannot even be dreamt of with conventional maps. Once the basic map has been defined and stored, a very large number of parameters pertaining to that particular map can be stored and made available on demand. Once the algorithm for processing and storing data has been worked out and implemented, fresh parameters can be easily added and existing parameters updated.

CHRONOLOGICAL VARIATIONS

The high density packing of information which is possible with computers makes it feasible to store and make available not only the latest, up-to-date version of a map but also earlier versions. Thus the user potentially can gain the freedom of moving through time if he so desires. An existing possibility is the creation of maps showing just the chronological

variation of a particular phenomenon. For example, a map showing just the increase and decrease of tree density on the campus of IIT, Bombay would be very useful for planning out conservation strategies, deciding where to launch tree planting drives etc.

ELIMINATION OF MANUAL PROCESSING

A very large part of the manual processing required to produce a conventional map would be eliminated if computers were used. Significant savings can thus be effected in time-cost.

PRECISE DATA

In conventional maps, only a processed version of the data is available to the user, it being nearly impossible to display an overall picture without eliminating precise, quantitative information. On the computer with suitable software, the raw data itself can be accessed and made available to the user.

MULTIPLE LEVELS

With the use of computers it is feasible for the user to access information at several levels, 'Zooming in'(so to speak)

to the precise area of his interest. Consider for example, a vet fresh from college trying to decide where to set up his practise. He might start with a map of pet ownership in Maharashtra seeing a high concentration in Bombay, he might decide to move there. But where in Bombay? He might then 'zoom in ' on Bombay to learn that the Bandra/Mahim area is a promising one. Zooming in future might even help him in deciding where exactly to set up his clinic.

CORRELATIONS

This is one of the most exciting possibilities with the use of computers and is a logical extension of the map's ability to provide insight into causes of phenomena. The user can select any two parameters and the computer can compute correlations between the two and display a map showing the magnitude of the correlation between the two. High correlation areas would indicate the possibility of a cause - effect relationship between the two phenomena. Thus, correlation maps can give insights into the causes of various phenomena and hence form a valuable tool for decision making.

From the advantages stated above it becomes obvious that the use of computers to store, process and display geographically distributed parameters offers significant advantages over

conventional paper maps. Due to these advantages, computer maps are ideally suited for use as forecasting planning and decision making tools with a wide range of applications.

4
On
The
Aim
Of
This
Project

The aim of this project was basically to write the software to create and access a data base for the storage of geographically distributed data on the PDP-11 computer. This involves two main sub-tasks

- Creating and updating the data base
- Accessing and displaying data

Creating and updating the data base involves providing a user friendly format for -

- inputting a map outline
- inputting various parameters pertaining to a particular map
- Updating or modifying a parameter already in store :
- processing inputted data and converting it into a machine useable form
- Storing this processed data in an accessible location.

Accessing and displaying data involves

- providing a user friendly format for specifying the information required.
- accessing the relevant data from its storage location
- converting this data into a form suitable for the graphic display
- displaying the data.

5
On
The
Results
Of
This
Project

The end result of this project was the program whose listing is attached with this report. The program has three segments which allow the user to

- input a new maps outline
- input or update parametric data pertaining to a particular map
- get a display of the data of his choice

The data structure used is a three level hierarchical structure.

- The first level is a single master file which has data on the titles and locations of the various maps available in store.
- Each map has a file which has data on the names and locations of the various parameters available in store for that particular map. This comprises the second level.
- The last level consists of parameter files which contain the actual data on the various parameters.

Ideally the program should handle data based on a continuous geographic distribution but this introduces a very high level of complexity into the process of encoding and displaying the

data. A compromise was therefore made and the present program handles data based on a uniformly divided square grid, each element being addressed by its X and Y coordinates. This has meant some loss in terms of the resolution and graphic quality of the image but this was considered acceptable because of the advantages it offers in terms of simplicity and adaptability. Two options were available for the display of data - using the graphics capability of the computer or utilising its alphanumeric capabilities. The former approach could give better results graphically but is very difficult to adapt to other systems which may or may not have the same graphics capabilities. On the other hand, using the alphanumeric capabilities allows use extension to other systems with minimum modifications. This approach was therefore chosen. The present program displays maps by printing (on the screen) character strings but does this in graphic mode. The same can easily be extended to the conventional dot-matrix printer though some distortion would result since the printer uses a grid consisting of vertically oriented rectangles (rather than squares) for printing characters. The graphic quality of the displayed image can be improved by creating and loading a new 'font' consisting of geometric shapes of varying visual magnitudes and using this for printing.

When the program is run, it first offers the user a menu consisting of three options

- Enter a new map
- Enter/update data
- Display a particular map.

If the user chooses the first option, he then has to enter an identifying name for the map, a title and the maximum vertical and horizontal coordinates. He can then enter the outline of the map based on a sequential scanning of horizontal lines, each line being displayed as he inputs it. At the end, he has the option of correcting any lines in which he has made a mistake.

Choice of the second option leads to two alternatives

- entering a new parameter
- updating an existing parameter

In the former case he has to first choose the map he is interested in and input the name of the new parameter. In the latter case, he has to select both the map and the parameter he is interested in. After this, in both cases, he proceeds to enter the locations (coordinates) and values of the parameter. In case the location he has entered is external to the map outline, the computer tells him so.

The third option leads to a display of menu of the maps in

store. On entering his choice, the user gets a list of parameters available for that particular map. On indicating his preference he gets a display of the data he has chosen.

L. D. C. Library
L. L. T. Bombay.

PROGRAM MAP
FLOPPY NO 5
ERGONOMICS LAB.

I. D. C. Library
L. I. T. Bombay.

BEFORE THIS PROGRAM IS RUN FOR THE FIRST TIME, PROGRAM START
(ALSO AVAILABLE ON THIS FLOPPY) HAS TO BE RUN WHICH CREATES
FILES FOR STORING INFORMATION ON THE NUMBER OF MAPS, TITLES OF
MAPS AVAILABLE AND THEIR STORAGE LOCATIONS

THE FOLLOWING ARE THE VARIABLES USED IN THE PROGRAM

GRID AN ARRAY OF PARAMETER VALUES
DISP A CHARACTER STRING ARRAY USED FOR DRAWING THE MAP
FILE THE MAP MASTER FILE NAME
TITLE THE TITLE OF THE MAP
PARFIL THE PARAMETER FILE NAME
PAR THE NAME OF THE PARAMETER
NX,NY NUMBER OF ELEMENTS ALONG X AND Y AXES
XMIN,YMIN THE SCREEN LOCATION OF LOWER LEFT CORNER OF THE MAP
IHT CHARACTER SIZE TO BE USED FOR DRAWING
STEP DISTANCE BETWEEN CONSECUTIVE LINES (SCREEN COORDINATES)
NPAR NUMBER OF PARAMETERS IN STORE

```
program MAP
logical*1 icom
dimension grid(80)
logical*1 disp(80),file(6),title(25),parfil(6),par(25)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar
```

THIS SECTION INITIALIZES ALL VARIABLES

```
do 21 i=1,80
disp(i)=' '
grid(i)=-1.0
continue
do 22 i=1,6
parfil(i)=' '
file(i)=' '
continue
do 23 i=1,25
title(i)=' '
par(i)=' '
continue
npar=0
type 2
format (/// ' OPTIONS AVAILABLE '///)
type 3
format (' 1 ENTER A NEW MAP '//
1 ' 2 ENTER/UPDATE DATA '//
1 ' 3 DISPLAY A PARTICULAR MAP '//
1 ' ENTER SERIAL NUMBER OF OPTION ')
accept 4,i
format (i5)
if ((i.lt.1).or.(i.gt.3)) go to 5
if (i.eq.1) call inmap
if (i.eq.2) call input
if (i.eq.3) call visual
go to 7
type 6
format (// ' INVALID OPTION TRY AGAIN '//)
```



```

format (' ANYTHING ELSE? Y/N ')
accept 9,icom
format (a1)
if (icom.eq.'Y') go to 1
if (icom.eq.'N') go to 10
go to 7
type 11
format (' SHUT DOWN ? Y/N ')
accept 9,icom
if (icom.eq.'N') go to 1
if (icom.eq.'Y') go to 12
go to 10
call clrscr
call clrtxt
stop
end

```

subroutine INMAP

THIS SEGMENT IS USED TO ACCEPT THE BASIC OUTLINE OF THE
MAP,CREATE DATA FILES AND STORE THE OUTLINE IN ONE OF THEM

```

logical*1 file(6),title(25),par(25),parfil(6),disp(80),isymb
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar

```

```

t=1.0
b=0.0
type 10
format(' ENTER IDENTIFYING NAME.MINIMUM 4 UNIQUE CHARACTERS ')

```

THE NAME ENTERED HERE IS USED TO IDENTIFY ALL FILES
CORRESPONDING TO THIS MAP.
FOR EXAMPLE IF THE USER ENTERS 'IITB' THE MAP OUTLINE IS STORED
IN A FILE CALLED 'IITBA' AND THE VARIOUS PARAMETERS ARE STORED
IN FILES CALLED 'IITBB','IITBC','IITBD' ETC.

```

accept 11,file
format (6a1)
file(5)=' '
file(6)=' '
type 12
format (' ENTER TITLE OR DESCRIPTION OF MAP.25 CHARACTERS ')
accept 14,title
format (25a1)
type 15
format (' ENTER MAXIMUM HORIZONTAL COORDINATE ')
accept 16,nx
format (i5)
type 17
format (' ENTER MAXIMUM VERTICAL COORDINATE ')
accept 18,ny
format (i5)
if ((nx.gt.80).or.(ny.gt.40)) go to 20
go to 30

```

```

type 21
format (' NUMBER OF ELEMENTS IS TOO LARGE '//
1 ' MAXIMUM HORIZONTAL COORDINATE IS 80 '//
1 ' MAXIMUM VERTICAL COORDINATE IS 40 '//
1 ' CHOOSE SUITABLE COORDINATES AND REENTER ')
go to 13

```

```

if ((nx.le.80).and.(ny.le.40)) iht=1
if ((nx.le.40).and.(ny.le.20)) iht=2
if ((nx.le.28).and.(ny.le.12)) iht=3
xmax=0.0
ymax=0.0
step=0.0

```



```

step=step+11./3
continue
do 32 i=1,nx
xmax=xmax+step
continue
do 33 i=1,ny
ymax=ymax+step
continue
xmin=1000.0-xmax
ymin=600.0-ymax
icom=1

```

```

call ab (icom)

```

'NOMAP.DAT' IS A FILE WHICH STORES THE NUMBER OF MAPS AVAILABLE

```

open (unit=11,name='NOMAP.DAT',type='OLD')
rewind 11
read (11,40) nmaps
format (1x,i5)
nmaps=nmaps+1
rewind 11
write (11,41) nmaps
format (2x,i5)
close (unit=11)

```

'MAP.DAT' STORES A LIST OF MAPS AVAILABLE AND THE IDENTIFYING NAME OF EACH

```

open (unit=11,name='MAP.DAT',access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
write (11'nmaps) file,title
close (unit=11)

```

THIS OPEN STATEMENT CREATES A NEW MAP FILE WHICH STORES THE BASIC PARAMETERS PERTAINING TO THE MAP AND A LIST OF THE PARAMETERS AVAILABLE ALONG WITH THE NAME OF THE FILE WHERE EACH IS STORED

```

npar=0
open (unit=11,name=file,access='DIRECT',
1 type='NEW',recordsize=90,initialsize=20)
write (11'1) nx,ny,xmin,ymin,iht,step,npar
file (5)='A'
write (11'2) file,title,t,b
do 50 i=3,20
j=i
write (11'j) parfil,par,t,b
continue
close (unit=11)

```

THIS OPEN STATEMENT CREATES A FILE FOR STORING THE MAP OUTLINE

```

open (unit=11,name=file,access='DIRECT',
1 type='NEW',recordsize=90,initialsize=20)
y=600.0
do 69 i=1,ny
nrec=i
do 61 ii=1,80
disp(ii)=' '
grid(ii)=-1.0
continue

```

```

call c (nrec,y)

```

```

y=y-step
call clrtxt
continue

```

```

type 71
format (' DO YOU WISH TO MODIFY ANYTHING ? ENTER Y/N ')
accept 72,icom
format (a2)

```



```

format (a2)
if (icom.eq.'Y.') go to 80
if (icom.eq.'N.') go to 90
go to 70

type 81
format (' ENTER LINE NUMBER ')
accept 82,i
format (i5)
y=600.0+step
do 83 j=1,i
y=y-step
continue
do 84 j=1,80
disp(j)=' '
grid(j)=-1.0
continue
call swmode ('ER')
call d (disp,nx,xmin,y)
call swmode ('OV')
call box (xmin,ymin,1000.0,600.0)
do 85 j=1,80
disp(j)='0'
continue
call scolor ('GRAY1',1)
call d (disp,nx,xmin,y)
call scolor ('GRAY3',3)
do 86 j=1,80
disp(j)=' '
continue

call c (i,y)

call clrtxt
go to 70

```

```

close (unit=11)
pause ' enter RETURN to clear '
call clrscr
call clrtxt
return
end

```

SUBROUTINE INPUT

THIS SEGMENT INPUTS AND UPDATES DATA VALUES
IN A MAP OF THE USERS CHOICE

```

logical*1 disp(80),file(6),title(25),parfil(6),par(25)
logical*1 a(20),temp(4),isymb
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar

```

```

data a(1),a(2),a(3),a(4),a(5)/'B','C','D','E','F'/
data a(6),a(7),a(8),a(9),a(10)/'G','H','I','J','K'/
data a(11),a(12),a(13),a(14),a(15)/'L','M','N','O','P'/
data a(16),a(17),a(18),a(19),a(20)/'Q','R','S','T','U'/

```

```

call choose
call pick (t,b)
if ((par(1).eq.'A').and.(par(2).eq.'A')) go to 30
go to 40

```

```

continue
do 31 i=1,6
parfil(i)=file(i)
continue

```

THIS OPEN STATEMENT CREATES A NEW FILE FOR THE NEW PARAMETER


```

open (unit=11,name=file,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
npar=npar+1
write (11'1) nx,ny,xmin,ymin,iht,step,npar
type 34
format (' ENTER NAME OF PARAMETER ')
accept 35,par
format (25a1)
parfil(5)=a(npar)
n=npar+2
t=-1000.0
b=1000.0
write (11'n) parfil,par,t,b
close (unit=11)
do 32 i=1,80
grid(i)=-1.0
continue
open (unit=12,name=parfil,access='DIRECT',
1 type='NEW',recordsize=90,initialsize=20)
do 33 i=1,ny
j=i
write (12'j) grid
continue
t=-1000.0
b=1000.0
ipar=npar
ic=0
go to 400

continue
open (unit=12,name=parfil,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
ic=1
type 41
format (' READY TO ACCEPT DATA ')
call stuff(t,b,12,ic)
call image(12,t,b)
type 45
format (' DO YOU WISH TO MAKE ANY MODIFICATIONS ? ')
accept 46,icom
format (a2)
if (icom.eq.'Y') ic=2
if (icom.eq.'Y ') go to 42
if (icom.eq.'N ') go to 50
go to 44

close (unit=12)
n=ipar+2
open (unit=11,name=file,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
write (11'1) nx,ny,xmin,ymin,iht,step,npar
write (11'n) parfil,par,t,b
close (unit=11)
pause ' enter RETURN to clear '
call clrscr
call clrtxt
return
end

```

subroutine VISUAL

THIS SEGMENT DISPLAYS A MAP OF THE USERS CHOICE

```

logical*1 disp(80),file(6),title(25),parfil(6),par(25)
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar
call choose
call pick (t,b)
if ((par(1).eq.'A').and.(par(2).eq.'A')) go to 10
open (unit=12,name=parfil,access='DIRECT',

```



```

i type='OLD',recordsize=90,initialsize=20)
call image (l2,t,b)
close (unit=l2)
pause ' enter RETURN to clear '
call clrtxt
call clrscr
return
end

```

```

subroutine ab (icom)

```

THIS SUBROUTINE DRAWS A BOX,WRITES TITLES BELOW IT AND DRAWS
A GRID OF 'O's IN THE BOX.
IF ICOM=0 THE GRID IS NOT DRAWN

```

logicalA1 disp(80),file(6),title(25),parfil(6),par(25),symb
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar
symb='O'
call initgr(5)
call clrscr
call clrtxt
call swindo (0.0,0.0,1000.0,600.0)
call box (xmin,ymin,1000.0,600.0)
call stxsiz (1, )
do 1 i=25,1,-1
if (title(i).eq.' ') go to 1
go to 2
continue
x=1000.0
i=i+1
do 3 j=1,i
x=x-12.0
continue
y=ymin-20.0
call d (title,i,x,y)
do 4 i=25,1,-1
if (par(i).eq.' ') go to 4
go to 5
continue
x=1000.0
i=i+1
do 6 j=1,i
x=x-12.0
continue
y=ymin-64.0
call d (par,i,x,y)
if (icom.eq.0) go to 10
do 9 i=1,80
disp(i)=' '
continue
do 7 i=1,nx
disp(i)=symb
continue
call stxsiz (iht,iht)
call scolor ('GRAY1',1)
x=xmin
y=600.0+step
do 8 i=1,ny
y=y-step
call d (disp,nx,x,y)
continue
call scolor ('GRAY3',3)
call stxsiz (iht,iht)
return
end

```

```

subroutine d (disp,nx,x,y)

```


LOCATION SPECIFIED BY X,Y

```
dimension disp(nx)
xx=x
yy=y
call move (xx,yy)
call text (disp)
return
end
```

subroutine c (i,y)

THIS SUBROUTINE ACCEPTS THE MAP OUTLINE (LINEWISE),
SPECIFIES LOCATIONS WITHIN THE OUTLINE BY A VALUE OF 1.0,
STORES THE GRID VALUES IN THE MAP OUTLINE FILE AND
UPDATES THE DISPLAY TO INCLUDE WHATEVER HAS BEEN ENTERED

```
logical*1 disp(80),file(6),title(25),parfil(6),par(25)
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar
type 10,i
format (' LINE NUMBER ',i3//
1 ' ENTER NUMBER OF SEGMENTS ')
accept 11,ns
format (i5)
call clrtxt
if (ns.eq.0) go to 5
do 4 j=1,ns
type 1,j
format (' ENTER INITIAL AND FINAL POINTS OF SEGMENT ',i3)
accept 2,ibeg
accept 2,iend
format (i5)
if (ibeg.lt.1) ibeg=1
if (iend.gt.nx) iend=nx
if (ibeg.gt.iend) ibeg=iend
do 3 jj=ibeg,iend
grid(jj)=1.0
disp(jj)='X'
continue
continue
call d (disp,nx,xmin,y)
nrec=i
write (11'nrec) grid
return
end
```

subroutine bc(icom)

THIS SUBROUTINE DRAWS THE MAP BOX AND TITLE (WITHOUT GRID)
USING SUBROUTINE AB AND THEN DRAWS THE BASIC MAP OUTLINE
USING 'O' SYMBOLS

```
logical*1 disp(80),file(6),title(25),parfil(6),par(25),s
dimension grid(80)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar
file(5)='A'
open (unit=15,name=file,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
call ab (icom)
call scolor ('GRAY1',1)
y=600.0
do 3 i=1,ny
j=i
do 1 ii=1,80
grid(ii)=0.0
disp(ii)=' '
continue
read (15'j) grid
do 2 ii=1,80
```



```

disp(ii)='
continue
read (15,j) grid
do 2 ii=1,80
if (grid(ii).eq.1.0) disp(ii)='0'
continue
call d (disp,nx,xmin,y)
y=y-step
continue
call scolor ('GRAY3',3)
close (unit=15)
file(5)=' '
return
end

```

subroutine choose

THIS SUBROUTINE ALLOWS THE USER TO SELECT A PARTICULAR MAP FROM A DISPLAYED MENU OF THE MAPS AVAILABLE IN STORE

```

dimension grid(80)
logical*1 disp(80),file(6),title(25),parfil(6),par(25)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar

```

```

open (unit=11,name='NOMAP.DAT',type='OLD')
read (11,1) nmaps
format (1x,i5)
close (unit=11)

```

```

open (unit=11,name='MAP.DAT',access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
type 2
format (' THE FOLLOWING MAPS ARE AVAILABLE '//
1 ' WHICH ONE ARE YOU INTERESTED IN ')
do 4 i=1,nmaps
j=i
read (11,j) file,title
type 3,i,title
format (1x,i3,1x,25a1)
continue
type 5
format (/// ENTER SERIAL NUMBER ')
accept 6,i
format (i3)
if (i.gt.nmaps) i=nmaps
if (i.lt.1) i=1
read (11,i) file,title
type 7,title
format (/// CHOSEN MAP IS '//1x,25a1)
close (unit=11)

```

```

open (unit=11,name=file,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
read (11,1) nx,ny,xmin,ymin,iht,step,npar
close (unit=11)

```

```

return
end

```

subroutine image(lunit,t,b)

THIS SUBROUTINE DRAWS THE PARAMETER MAP STORED IN THE FILE OPEN ON LOGICAL UNIT 'LUNIT'.
 DEPENDING ON THE VALUE OF THE PARAMETER AT A PARTICULAR LOCATION ON A SCALE WITH A MAXIMUM OF 'T' AND A MINIMUM OF 'B' ONE OF THE SYMBOLS EL(1) TO EL(5) IS ALLOCATED TO THAT LOCATION AND USED FOR DRAWING

```

logical*1 disp(80),file(6),title(25),parfil(6),par(25)
logical*1 el(5),isymb

```



```
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par  
1 /b7/nx,ny,xmin,ymin,iht,step,npar
```

```
icom=0  
call bc(icom)  
cr=t-b  
cr=cr*0.25  
c=0.0  
do 1 i=1,5  
block(i)=b+c  
c=c+cr  
continue  
data el(1),el(2),el(3),el(4),el(5)('/:','-','+','#','*')/  
y=600.0  
do 5 i=1,ny  
do 2 j=1,80  
disp(j)=' '  
continue  
j=i  
read (lunit'j) grid  
do 4 j=1,nx  
if (grid(j).lt.0.0) go to 4  
do 3 jj=1,5  
if (grid(j).ge.block(jj)) disp(j)=el(jj)  
continue  
continue  
call d (disp,nx,xmin,y)  
y=y-step  
continue  
return  
end
```

```
subroutine pick (t,b)
```

THIS SUBROUTINE ALLOWS THE USER TO SELECT ONE OF THE PARAMETERS AVAILABLE IN STORE FOR THE MAP SELECTED WITH SUBROUTINE CHOOSE. IN CASE NONE OF THE PARAMETERS AVAILABLE IS OF INTEREST TO THE USER, A VALUE OF '*' IS RETURNED FOR PAR(1) AND PAR(2)

```
logical*1 disp(80),file(6),title(25),parfil(6),par(25),icom  
dimension grid(80)  
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par  
1 /b7/nx,ny,xmin,ymin,iht,step,npar
```

```
open (unit=11,name=file,access='DIRECT',  
1 type='OLD',recordsize=90,initialsize=20)  
i=1  
read(11'1) nx,ny,xmin,ymin,iht,step,npar  
read(11'2) parfil,par,t,b  
if (npar.eq.0) go to 15  
type 1,title  
format (' THE FOLLOWING PARAMETERS ARE IN STORE FOR ',25a1)  
do 3 i=3,npar+2  
j=i  
read (11'j) parfil,par,t,b  
k=i-2  
type 2,k,par  
format (1x,i3,1x,25a1)  
continue  
type 4  
format (/' WHICH ONE ARE YOU INTERESTED IN ? '/  
1 ' ENTER SERIAL NUMBER '/' ENTER 0 FOR NONE OF THESE ')  
accept 5,ipar  
format (i5)  
if (ipar.lt.1) go to 11  
if (ipar.gt.npar) ipar=npar  
ipar=ipar+1  
do 8 i=1,25  
par(i)=' '  
continue
```



```

continue
read (11'(ipar+1)) parfil,par,t,b
type 6,par
format (' PARAMETER CHOSEN IS '/1x,25a1)
go to 20
type 12
format (' NONE OF THE ABOVE? Y/N ')
accept 13,icom
format (a1)
if (icom.eq.'Y') go to 15
if (icom.eq.'N') go to 7
go to 11
par(1)='A'
par(2)='A'
close (unit=11)
return
end

```

```

subroutine stuff (t,b,lunit,icom)

```

THIS SUBROUTINE IS USED FOR DATA INPUT
 WHEN THE USER ENTERS THE COORDINATES OF A POINT,A CHECK
 IS MADE TO DETERMINE WHETHER THE POINT ENTERED LIES WITHIN
 THE MAP OUTLINE OR NOT.IF NOT THE USER GETS AN INDICATION
 TO THIS EFFECT.THE DATA ENTERED IS STORED AND AN INDICATION
 APPEARS ON THE SCREEN SHOWING WHICH POINTS DATA HAS BEEN
 ENTERED FOR.

```

dimension grid (80),comp(80)
logical*1 disp(80),file(6),title(25),parfil(6),par(25)
common /b1/grid/b2/disp/b3/file/b4/title/b5/parfil/b6/par
1 /b7/nx,ny,xmin,ymin,iht,step,npar

```

```

file(5)='A'
open (unit=14,name=file,access='DIRECT',
1 type='OLD',recordsize=90,initialsize=20)
if (icom.eq.0) call bc(icom)
if (icom.eq.1) call image (12,t,b)
type 2
format (' ENTER X-COORDINATE,Y-COORDINATE '//
1 ' TO EXIT,ENTER TWO ZEROES ')
do 3 i=1,80
disp(i)=' '
continue
accept 4,ix
accept 4,iy
format (i5)
if ((ix.eq.0).and.(iy.eq.0)) go to 10
if (ix.lt.1) ix=1
if (iy.lt.1) iy=1
if (ix.gt.nx) ix=nx
if (iy.gt.ny) iy=ny
if (val.gt.t) t=val
if (val.lt.b) b=val
nrec=ny+1-iy
read (14'nrec) comp
nrec=ny+1-iy
if (comp(ix).ne.1.0) go to 8
type 5
format (' ENTER VALUE OF PARAMETER ')
accept 6,val
format (f7.2)
read (lunit'nrec) grid
grid (ix)=val
disp(ix)='0'
y=600.0+step
nrec=ny+1-iy
do 7 i=1,nrec
y=y-step
continue
call d(disp,nx,xmin,y)
write (lunit'nrec) grid
call clstxt

```



```
y-y-step  
continue  
call d(disp,nx,xmin,y)  
write (lunit'nrec) grid  
call clrtxt  
go to 1  
  type 9  
format (' THE POINT YOU HAVE ENTERED IS EXTERNAL TO THE MAP'//)  
go to 1  
call clrtxt  
file(5)=' '  
close (unit=14)  
return  
end
```

I. D. C. Library
L. L. T. Bombay.

I. D. C. Library
L. L. T. Bombay.

