SURVEY
An Interactive System for The Generation of Botanical Maps

by

J.A. Smith and D.C. Gray

Research Report CS-78-20
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

March 1978/Revised February 1979

TABLE OF CONTENTS

Perth County Project — Page

1.0 Introduction

In a botanical survey, there can be a large amount of information which must be maintained. For instance, the survey can cover a large number of survey stations and the information collected at each station can be extensive.

Once the data has been collected, it is usually desired to be able to analyse the data and display the more important aspects in some form. If done by hand, this analysis and mapping is a very tedious process. Soper[1,2] has demonstrated some methods of doing this with a computer.

The SURVEY system described here was developed in conjunction with Dr. A.A. Wellwood of Wilfrid Laurier University. It assists the user in two ways.

1. It allows the user to obtain checklists and computer printouts of the botanical data ordered either by plant or by station.
2. It allows the user to interactively preview and compare the distributions of any desired plants in terms of their distribution over the stations.

The SURVEY system has been used on a botanical survey of Perth County, Ontario. That survey contained 227 stations and over 1000 plants.

The SURVEY system of programs is written entirely in FORTRAN IV and runs under VM/CMS on an IBM 370 computer. The subsequent discussion assumes that the user has only an elementary knowledge of CMS to the extent of being able to sign on and to do some simple operations in it.

---

[1] Soper, J.H. Mapping the Distribution of Plants by Machine, Canadian Journal of Botany, Volume 42, 1964.

[2] Soper, J.H. Machine-Plotting of Phytogeographical Data, Canadian Geographer, X, 1, 1966.

The SURVEY system consists of several subsystems. The CHECKLIST subsystem is used to obtain plant listings and the checklists. The MAP subsystem is used to preview the plant distributions and to generate hard copy plots of these distributions. These systems are described in Chapters 2 and 3 respectively.

The organization of data and subsystems which are used to modify the base map itself are described in the appendicies.

2. The CHECKLIST System

The CHECKLIST system is an interactive system which allows a user to generate various paper records of the survey information. For instance one can generate a list of the plants by stations or stations by plants. The following sections describe how to run it.

2.1 Running the Program

This system requires 1 m of storage and a temporary D disk to be defined before it may be run. The commands to accomplish this initialization are as follows.

```
DEFINE STORAGE 1M
IPL CMS
DEFINE D
```

The data for this program are organized into many separate files. The names of these files are as follows.

ADATA	HDATA	ODATA	VDATA
BDATA	IDATA	PDATA	WXYZDATA
CDATA	JDATA	QDATA	
DDATA	KDATA	RDATA	
EDATA	LDATA	SDATA	
FDATA	MDATA	TDATA	
GDATA	NDATA	UDATA	

The naming convention is such that ADATA contains information for all plants whose names begin with A, and so on. The program allows either an individual or the entire set of files to be processed.

To actually run the program, type

```
CKLIST filename
```

where filename is either one of the above file names, or the word ALL to process all the files. The program will then be executed and allow the user to interact with it. The following section describes this interaction.

2.2 Interaction with the CHECKLIST Program

The CHECKLIST program prompts the users to determine what he requires. The following questions are as read by the system. The answers of Y(yes) or N(no) are the only legal responses.

1. ORIGINAL INPUT ECHOED? (Y/N):

   A response of Y will cause the system to generate a listing of the plant and station information (by plant) exactly as it exists in the data file. This option is provided for checking the data when input.

2. SORTED INPUT ECHOED? (Y/N):

   A response of Y will cause the system to generate a listing of the plant and station informations (by plant) with the station data sorted into ascending order.

3. PLANTS BY STATIONS: (Y/N):

   A response of Y will cause the system to generate a listing of the plant and station information by station.

4. PLANT CHECKLIST? (Y/N):

   A response of Y will cause the system to generate an alphabetical list of all plants in the data file.

5. STATION CHECKLIST? (Y/N):

   A response of Y will cause the system to generate a list of the plants at specific stations. The system allows either all stations or individual stations to be listed and prompts the user for this information.

The output of the system for the ADATA file is shown on the next several pages. To obtain these printouts, the answer Y (yes) was replied to all the above questions. Note: Only the first page of each printout is shown.

------DATA AS INPUT------

1. ORIGINAL INPUT DATA

1. ABIES BALSAMEA
56 7 58 34 108 121 36 127 193 195 164 37

2. ACALYPHA RHOMBOIDEA
130 10

3. ACER NEGUNDO
38 64 71 79 87 89 102 105 8 55 129 130 1 69 98 24 39 47
41 5 128 25 92 112 117 91 110 199 160 152 169 178 180 187 188 147 201
204 206 208 174 209 210 212 190 216 196 173 175 194 205 215 217 177 189

4. ACER NIGRUM
11 70 54 107 7 116 102 61 28 21 94 49 84 96 140 22 150 50 153 88
168 162 215 191 202 186 166

5. ACER PLATANOIDES
199 169 172 173 189 194 210 216 190 91

6. ACER RUBRUM
9 34 50 51 17 15 22 19 30 58 33 36 68 56 57 105 8 115 117 129
91 41 130 103 81 119 80 112 154 148 149 150 192 102 198 164 195 162 171
185

7. ACER SACCHARINUM
8 9 10 11 12 13 16 17 19 22 23 24 26 30 33 37 38 40 51 2
55 56 57 59 41 43 42 36 35 34 63 62 70 71 81 88 90 85 98 103
58 6 111 113 114 115 54 91 130 61 78 124 125 97 28 47 64 65 66 75
76 128 127 80 108 94 129 49 50 96 112 100 122 135 155 154 68
145 146 149 150 157 160 132 147 152 162 166 170 172 174 179 183 190 193 194 195
79 110 201 204 205 209 214 185 210 164 168 192 180 163 206 196 220 133 202 39
148 156 167 217 173 176 191 187 169 216 189 188

8. ACER SACCHARUM
5 7 10 11 14 15 16 17 21 22 24 28 30 32 36 40 49 50 51 20

------DATA IN ORDER------

2. SORTED INPUT DATA

1. ABIES BALSAMEA (12 STATONS)
7 34 36 37 56 58 108 121 127 164 193 195
(CHECKSUM = 1136)

2. ACALYPHA RHOMBOIDEA (2 STATONS)
10 130
(CHECKSUM = 140)

3. ACER NEGUNDO (55 STATONS)
1 5 8 24 25 38 39 41 47 55 64 69 71 79 87 89 91 92 98 10
105 110 112 117 128 129 130 147 152 160 169 173 174 175 177 178 180 187 188 18
190 194 196 199 201 204 205 206 208 209 210 212 215 216 217
(CHECKSUM = 7287)

4. ACER NIGRUM (27 STATONS)
7 11 21 22 28 49 50 54 61 70 84 88 94 96 102 107 116 140 150 15
162 166 168 186 191 202 215
(CHECKSUM = 2793)

5. ACER PLATANOIDES (10 STATONS)
91 169 172 173 189 190 194 199 210 216
(CHECKSUM = 1803)

6. ACER RUBRUM (40 STATONS)
8 9 15 17 19 22 30 33 34 36 41 50 51 56 57 58 68 80 81 9
102 103 105 112 115 117 119 129 130 148 149 150 154 162 164 171 185 192 195 19
(CHECKSUM = 3756)

7. ACER SACCHARINUM (129 STATONS)
2 6 8 9 10 11 12 13 16 17 19 22 23 24 26 28 30 33 34 3
36 37 38 39 40 41 42 43 47 49 50 51 54 55 56 57 58 59 61 6
63 64 65 66 68 70 71 75 76 78 79 80 81 85 88 90 91 94 96 9
98 100 103 108 110 111 112 113 114 115 122 124 125 127 128 129 130 132 133 13
145 146 147 148 149 150 152 154 155 156 157 160 162 163 164 166 167 168 169 17
172 173 174 176 179 180 183 185 187 188 189 190 191 192 193 194 195 196 201 20
204 205 206 209 210 214 216 217 220
(CHECKSUM = 14088)

3. PLANTS BY STATION

THE PLANTS IN STATION 1 ARE:

ACER NEGUNDO
ARISAEMA ATRORUBENS F. ATRORUBENS
ASCLEPIAS INCARNATA
ASTER SIMPLEX
ALISMA TRIVIALE
ARISAEMA ATRORUBENS F. ZEBRINUM
ASCLEPIAS SYRIACA
ATHYRIUM FILIX-FEMINA
ARCTIUM MINUS
ARTEMISIA BIENNIS
ASTER NOVAE-ANGLIAE

THE PLANTS IN STATION 2 ARE:

ACER SACCHARINUM
AMBROSIA ARTEMISIIFOLIA
ASCLEPIAS INCARNATA
ACHILLEA MILLEFOLIUM
ARCTIUM MINUS
ASCLEPIAS SYRIACA
AGROPYRON REPENS
ARTEMISIA BIENNIS
ATRIPLEX PATULA

THE PLANTS IN STATION 3 ARE:

ACHILLEA MILLEFOLIUM
ASCLEPIAS SYRIACA
ANDOMEDA GLAUCOPHYLLA
APOCYNUM CANNABINUM

THE PLANTS IN STATION 4 ARE:

ACTAEA PACHYPODA
ARALIA NUDICAULIS
ACTAEA RUBRA
AGRIMONIA GRYPOSEPALA

THE PLANTS IN STATION 5 ARE:

ACER NEGUNDO
ACTAEA PACHYPODA
ALLIUM TRICOCCUM
AMELANCHIER WIEGANDII
ARCTIUM MINUS
ASCLEPIAS SYRIACA
ACER SACCHARUM
ACTAEA RUBRA
ALOPECURUS AEQUALIS VAR. AEQUALIS
ANAPHALIS MARGARITACEA
ARISAEMA ATRORUBENS F. ATRORUBENS
ATHYRIUM FILIX-FEMINA
ACHILLEA MILLEFOLIUM
AGRIMONIA GRYPOSEPALA
AMELANCHIER INTERMEDIA
ANTENNARIA PETALOIDEA
ARISAEMA ATRORUBENS F. VIRIDE

THE PLANTS IN STATION 6 ARE:

ACER SACCHARINUM
ACORUS CALAMUS
ANEMONE CANADENSIS
ATHYRIUM FILIX-FEMINA
ACER SACCHARUM
ACTAEA RUBRA
ARCTIUM MINUS
ACHILLEA MILLEFOLIUM
AGROPYRON REPENS
ASCLEPIAS INCARNATA

PERTH COUNTY SURVEY

PLANT CHECKLIST

1. ABIES BALSAMEA					
2. ACALYPHA RHOMBOIDEA					
3. ACER NEGUNDO					
4. ACER NIGRUM					
5. ACER PLATANOIDES					
6. ACER RUBRUM					
7. ACER SACCHARINUM					
8. ACER SACCHARUM					
9. ACER SPICATUM					
10. ACHILLEA MILLEFOLIUM					
11. ACHILLEA MILLEFOLIUM F. ROSEA					
12. ACHILLEA PTARMICA					
13. ACORUS CALAMUS					
14. ACTAEA PACHYPODA					
15. ACTAEA PACHYPODA F. RUBROCARPA					
16. ACTAEA RUBRA					
17. ACTAEA RUBRA F. NEGLECTA					
18. ADIANTUM PEDATUM					
19. AEGOPODIUM PODAGRARIA					
20. AESCULUS GLABRA					
21. AESCULUS HIPPOCASTANUM					
22. AGRIMONIA GRYPOSEPALA					
23. AGROPYRON REPENS					
24. AGROSTIS ALBA					

PERTH COUNTY SURVEY

CHECKLIST FOR STATION 1

1. ACER NEGUNDO					
2. ALISMA TRIVIALE					
3. ARCTIUM MINUS					
4. ARISAEMA ATRORUBENS F. ATRORUBENS					
5. ARISAEMA ATRORUBENS F. ZEBRINUM					
6. ARTEMISIA BIENNIS					
7. ASCLEPIAS INCARNATA					
8. ASCLEPIAS SYRIACA					
9. ASTER NOVAE-ANGLIAE					
10. ASTER SIMPLEX					
11. ATHYRIUM FILIX-FEMINA					

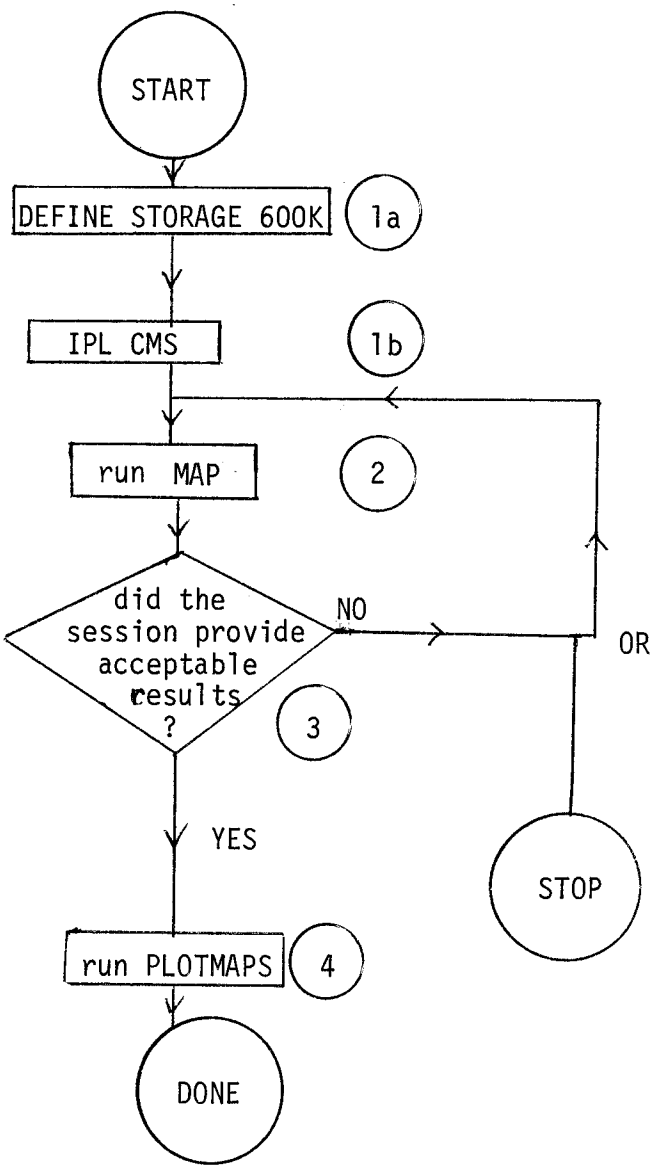
3.0 The MAP System

The MAP system is interactive and allows a user to display plant location data on a Tektronix graphics terminal and to produce hardcopy plots of this data on a Calcomp plotter.

The plants to be displayed are referenced by plant number (obtained from the plant checklist). These numbers may be entered one at a time on the terminal or may be obtained from a file containing a list of the required plant numbers.

3.1 Running the Program

The MAP system is very easy to use. The following diagram illustrates the sequence of operations to get started. A description of the diagram also follows.



1. a&b) Due to the large amount of "incore" data maintained by the MAP program, it is necessary to re-define the Virtual Machine storage capacity to 600K. This is accomplished by issuing the following CMS commands:

```
CP   DEFINE STORAGE 600K
IPL  CMS
```

2. The program produces as output an intermediate form plot file called FILE HDCPY B containing the maps that are plotted during the session.
3. If the plots requested during the session are not required in hardcopy form (Calcomp plot) or if they were not acceptable, the user may stop or rerun.
4. The program PLOTMAPS produces a Calcomp plot of the "plots" made during the session. This program must be run in the same session as to produce the desired results since the intermediate file, FILE HDCPY B, is only a temporary file.

3.2 Using the System

The Tektronics screen is divided into 3 major areas by the program. These areas are the scratch pad, the currently displayed plant number table and the map itself. Within the scratch pad area, all commands and prompts are contained. The plant number table contains both the symbols and the numbers of the plants currently displayed. This table will also be referred to as the active list. The map area contains the map and the name of the current working file. Throughout the terminal session, some file will always be designated as being the current file. When the program first starts up, a filename (the current file name) will be prompted for. This filename can be up to 8 alphanumeric digits in length, the first of which must be non-numeric. The CMS filenames used by the program are <user specified filename> PDATA A. Section 3.2.1 discusses how different data files may be accessed. Up to 82 file references can be made during one session.

At any time (except the first time!) when FILENAME? is prompted for, the current filenames can be entered as * instead of typing out the full name. When the scratch pad area becomes full, the message MORE ... will appear on the screen. Typing the RETURN key on the terminal will cause a refresh of the screen to occur.

On termination of the program, the program PLOTMAPS must be run to actually submit the maps to be plotted as was mentioned earlier.

3.2.1 User Commands

There are 8 valid commands that are accepted by the program MAP. These commands allow the user to change the screen display, manipulate files,

change the plotting symbol and terminate the session. A discussion of the commands now follows.

The add (AD) command is used to add a plant number to the active list and display the stations at which the plant is found. When AD is typed in response to COMMAND? first the plant number is prompted for. The name of the plant corresponding to the number entered is then echoed and a prompt questions whether or not the plant name is correct. Entering Y (yes) indicates that the name is correct while N (no) indicates that it is incorrect and the number is neither added to the active list or the stations displayed. To leave AD mode and return to COMMAND mode enter a plant number of zero (0).

To delete a plant number from the active list and all associated stations from the screen display on the next screen refresh, the delete (DE) command is typed. As with AD, the plant number is entered and then the name echoed and verified before the actual deletion occurs. Typing a plant number of zero causes return to COMMAND mode.

The newfile (NF) command allows the user to change the current working file to the filename entered. If the file specified already exists, the data contained in the file is read and displayed on the screen. Otherwise a new file is simply created. It may sometimes be the case that when NF is typed, the program will issue the prompt RESAVE? indicating that the active list was changed but not saved. Answering yes (Y) starts the save process to save the current active list in a file while answering no (N) does not.

It was just mentioned that a RESAVE can take place implicitly when NF is typed. To explicitly save the data necessary to reproduce the current screen display in a file the save (SA) command is used. When SA is typed

a filename is prompted for in which the active list can be saved. Any valid filename including * may be used.

To produce a hardcopy plot of the current screen display or of another data file the user simply types PL (plot). As was the case with NF, the RESAVE option can occur before the name of the file to be plotted is prompted for.

The current station plotting symbol can be changed by using the SS (set symbol) command. The number of the new symbol is prompted for then read by the program when it is entered. The number can be in the range 1 to 128, although the range 1 to 12 is more desirable since then the symbol is centred about the actual coordinates of the stations.

To refresh the current screen display the RE (refresh) command is typed. The scratch pad is erased, the plants currently displayed table is updated and the stations associated with deleted plants are removed from the map display.

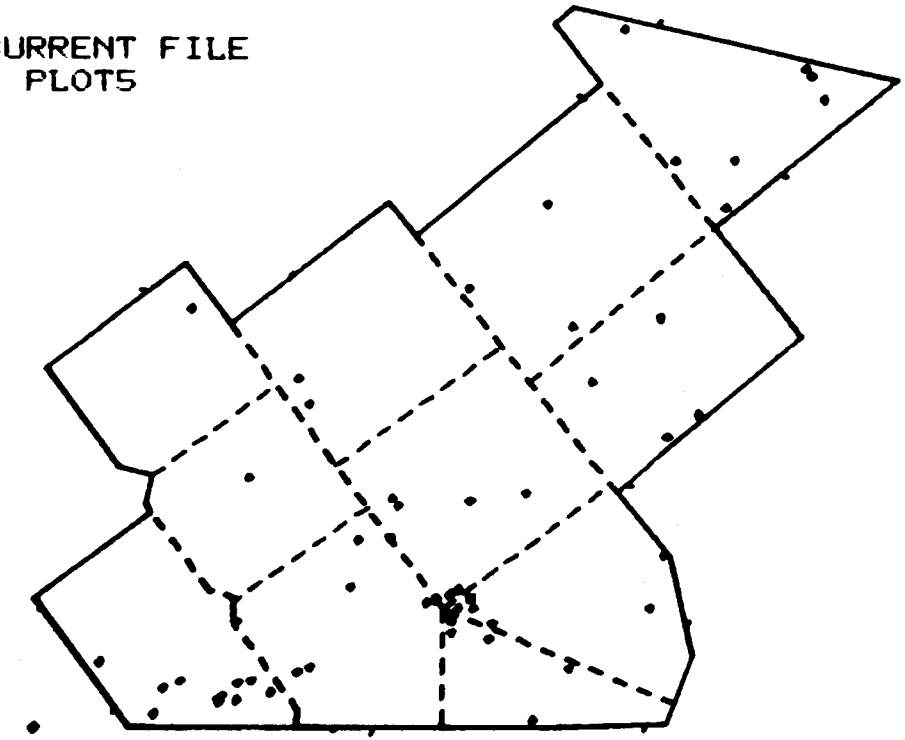
To terminate the session type DO (done).

The following pages are a sample terminal session which illustrate the use of the commands previously discussed. Sample output from the program is also included.

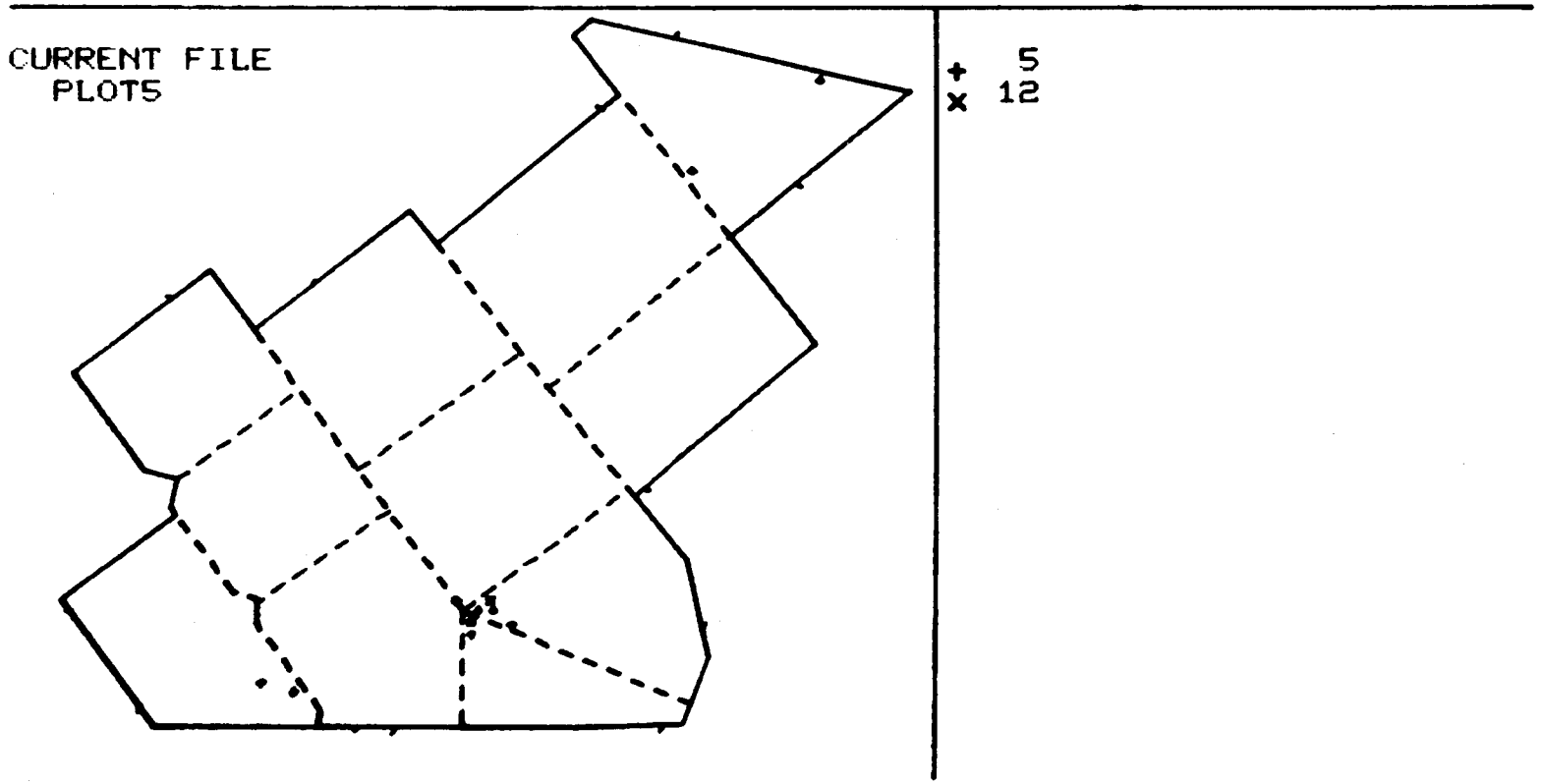
```
COMMAND ?   ss
SYMBOL NO.? 3
COMMAND ?   ad
PLANT NO.?  5          (0 TO QUIT)
ACER PLATANOIDES
OK ?(Y!N)   y
PLANT NO.?  0          (0 TO QUIT)
COMMAND ?   ss
SYMBOL NO.? 5
COMMAND ?   ad
PLANT NO.?  3          (0 TO QUIT)
ACER NEGUNDO
OK ?(Y!N)   y
PLANT NO.?  0          (0 TO QUIT)
```

```
COMMAND ?   ss
SYMBOL NO.? 4
COMMAND ?   ad
PLANT NO.?  12         (0 TO QUIT)
ACHILLEA PTARMICA
OK ?(Y!N)   y
PLANT NO.?  0          (0 TO QUIT)
COMMAND ?   de
PLANT NO.?  3          (0 TO QUIT)
ACER NEGUNDO
OK ?(Y!N)   y
PLANT NUMBER   3 DELETED
PLANT NO.?  0          (0 TO QUIT)
MORE...
```

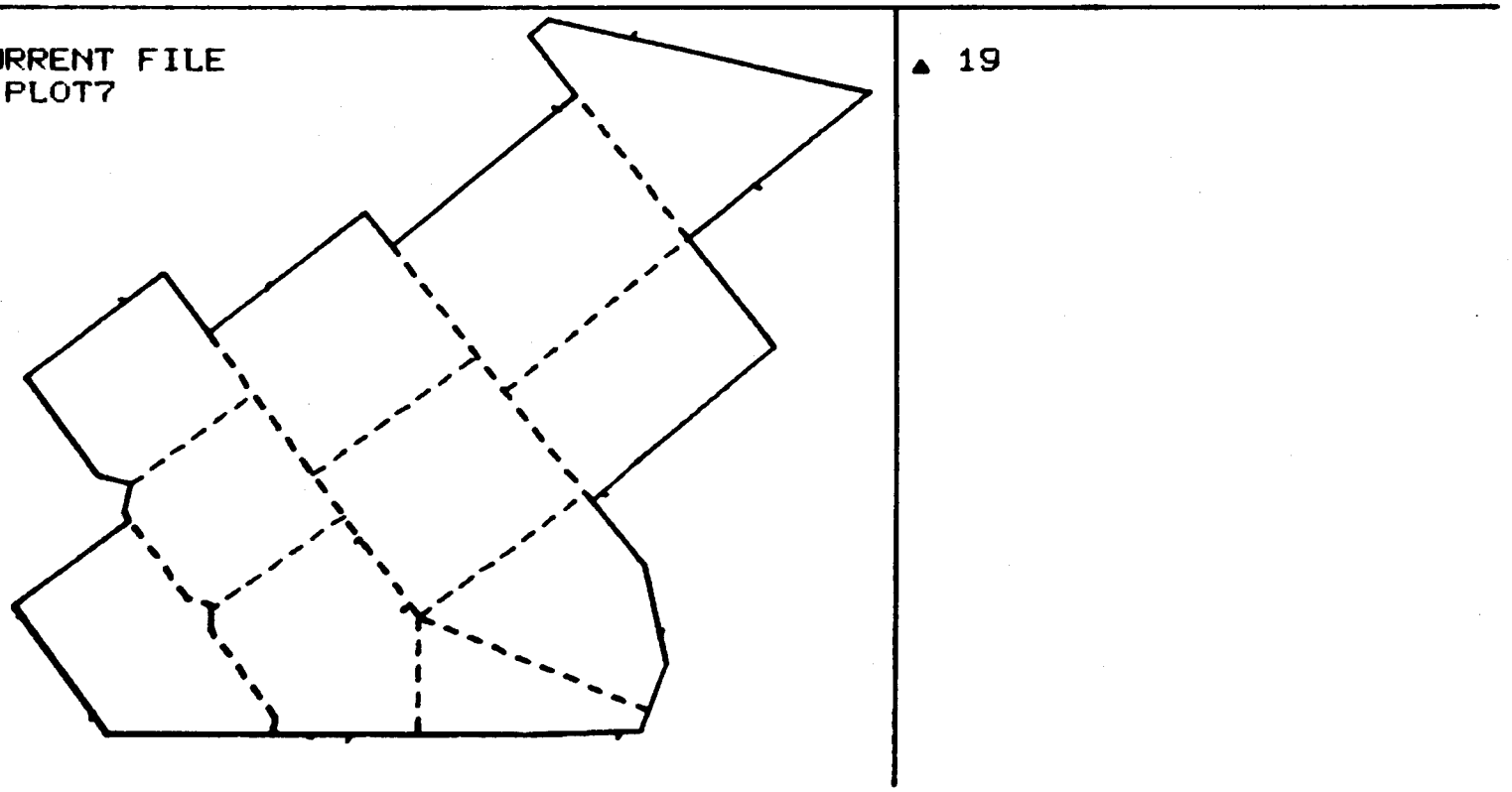
CURRENT FILE
PLOT5



```
COMMAND ?   save
FILENAME ?  *
FILE PLOTS     SAVED
COMMAND ?   plot
FILENAME ?  *
PLOTS     PLOTTED
COMMAND ?   nf
FILENAME ?  plot7
```



```
COMMAND ?   ad
PLANT NO.?  18        (0 TO QUIT)
ADIANTUM PEDATUM
OK ?(Y:N)   n
PLANT NO.?  19        (0 TO QUIT)
AEGOPODIUM PODAGRARIA
OK ?(Y:N)   y
PLANT NO.?  0         (0 TO QUIT)
COMMAND ?   save
FILENAME ?  *
FILE PLOT7     SAVED
COMMAND ?   done
  PLOT NUMBER     1   COMPLETED
R:
```



PERTH COUNTY SURVEY

SPECIES PLOTTED

- ACORUS CALAMUS
- ACTAEA PACHYPODA
- ACER PLATANOIDES

DATE: WED, APR. 27, 1977

3.2.2 Summary of MAP Commands

Command	Meaning
NF - NEWFILE	- start working with a new file. This file becomes the current file. If the file already exists and contains data, the data is read and the screen is updated accordingly.
AD - ADD	- Add a plant number to the active list and display the stations at which the particular plant is found on the screen.
DE - DELETE	- delete a plant number from the active list and remove all stations at which the plant was found from the display on the next screen refersh.
SA - SAVE	- save the current screen display information (not including deletes) in the specified filename.
PL - PLOT	- make a Calcomp plot of the filename specified.
SS - SET SYMBOL	- set the station plotting symbol
RE - REFRESH	- refresh the screen display and clear the scratch pad area.
DO - DONE	- terminat the session.

All commands which require additional information to carry out a desired action require this information by explicitly prompting for it.

Any command which prompts for 'FILENAME?' will accept '*' as equivalent to the current filename displayed on the screen.

3.3 MAP: System Messages

1. COMMAND?

   Enter one of the 8 valid commands.

2. FILENAME?

   Enter a filename up to 8 characters long.

3. SYMBOL NUMBER?

   Enter a valid symbol number to designate the current plotting symbol to be used.

4. SAVE? (Y/N)

   Enter 'Y' to start the save process, 'N' otherwise.

5. PLANT NUMBER <NUM> DELETED

   This message verifies the fact that plant <NUM> has been successfully deleted and on the next refresh of the screen, the associated station data will not appear.

6. NEWFILE

   The filename specified after a NF command was issued does not already exist and thus was created.

7. FILE SPACE EXHAUSTED

   No more newfiles may be created since the maximum number of files that may be used in one session has been reached.

8. PLANT NO.? (0 TO QUIT)

   This message is issued in both ADD and DELETE mode and requests that the user enter a valid plant number to be added or deleted respectively. Entering 0 returns you to COMMAND mode.

9. <PLANT NAME>

When a plant number is typed in response to system message 8, the plant name is typed at the terminal for verification purposes.

10. OK? (YIN)

A Yes or No answer is required to verify that the plant name just echoed to the terminal is correct with respect to the plant number entered.

11. INTERNAL STORAGE SPACE EXHAUSTED

The stations associated with 100 plant numbers are currently displayed and no more plant numbers may be added.

12. FILE <FILENAME> SAVED

This message verifies the fact that the data associated with the display description file was saved.

13. RESAVE? (YIN)

The screen display has been changed since the last save occured. Answering 'Y' starts the SAVE process, 'N' forces a return directly to COMMAND mode.

14. MORE ...

The program is in a "wait" state. To continue the user must press the RETURN key at the terminal.

15. <FILENAME> PLOTTED

This message verifies the fact that the display description file specified was used carefully to produce a hardcopy plot.

3.4 MAP System: Error and Warning Messages

Type

1. READ ERROR ON UNIT 5

   A hardware read error has occured when a read was attempted from the terminal. The system will attempt to correct this situation by re-prompting for the last command/datum entered. If read errors persist, notify a hardware technician.

2. INVALID COMMAND

   The command entered did not match anyone of the eight valid commands. Check that the spelling of the command entered was correct and that it is actually a valid command, to correct this error.

3. STATION <NUM> OUT OF RANGE

   A station number greater or less than the valid range of station numbers has been encountered in one of the input plant data files. To correct this error, locate plant <NUM> in the appropriate data file and modify the invalid station number.

4. STATION <NUM> DEFINED TWICE

   A duplicate station number has been encountered in the list of station numbers for some plant in one of the data files. Examine the data files and locate this multiple definition then, depending on the situation, either delete it or change it to the desired station number.

5. INVALID INPUT

   Alphabetic input was entered at the terminal when numeric was expected. To correct, enter the numeric input requested.

6. NUMBER PREVIOUSLY ENTERED

   The plant number just entered is already currently active and it's associated stations are displayed on the screen. Check to see that the plant number added was the one that was really desired.

7. PLANT NEVER ADDED

   An attempt has been made to DELETE a plant number which is not currently active. Check to see that the plant number typed was correct.

8. SYMBOL <NUM> OUT OF RANGE

   The station symbol number just entered via the SS command or from a data file being read, is less than 0 or greater than 128. In the SS command case, re-enter a valid symbol number. In the latter case correct the entry in the data file so that the symbol number is in the valid range.

9. PLANT <NUM> OUT OF RANGE

   The plant number just entered via the AD(D) command or being read from a data file is greater than or less than the valid range of plant numbers. In the AD command case find the correct number of the plant desired then re-enter. In the latter case, examine the data file for the invalid plant number and correct it.

10. INVALID REPLY

    A response other than Y(es) or N(o) has been entered to the question asked. To correct, answer yes or no.

11. FILENAME ALREADY USED

    An attempt has been made to save the information necessary to re-create the current screen in a file that already exists, other than the current file. The data is not saved. In order to complete the save pick a new filename or save as the current file (*).

12. PLANT <NUM> DEFINED TWICE

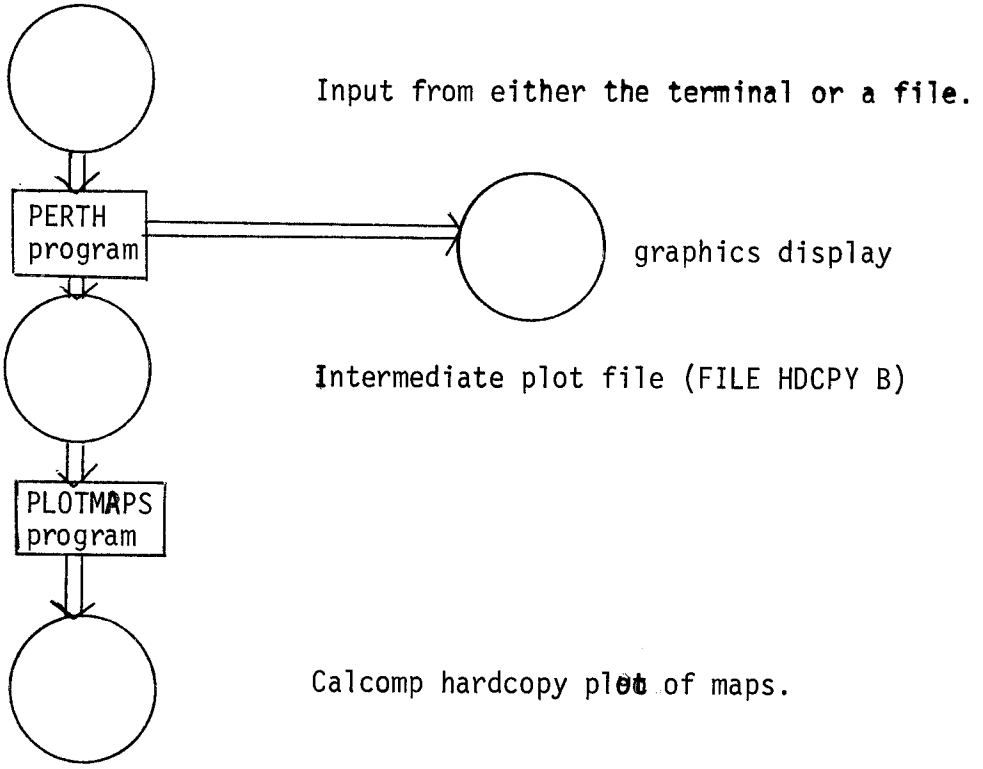
    A screen display description file was being read in when a duplicate plant number was encountered. To correct, examine this data file, locate the duplicate plant number and then correct it accordingly.

13. <FILENAME> NOT FOUND

    An attempt has been made to try to plot using a non-existent descriptor file. Check to see if the appropriate name has been issued.

3.5 Design, Implementation and Data Structures

An overview with respect to data flow of the MAP system is best described by the following diagram:



All data structures, variables and constants are documented fully at either the beginning of the main program or at the beginning of the subprogram in which they are used. Five data structures will now be discussed. The array STATION is dimensioned at NROWS(1000) by NUMSTN(227). The $(i,j)^{th}$ entry in this array is one if plant number i is found at station j and zero otherwise. The PLANT array is of length NROWS where the i^{th} entry contains the name of plant number i. NUMBPL is an array of length NROWS where the i^{th} entry is the number of stations that plant number i is found at. The arrays PNUM and PSYM are respectively the list of active plant numbers and the list of plotting symbol numbers corresponding to the entries in PNUM.

The screen display is divided into 3 areas: the scratch pad, the map and the active list. Line positioning is maintained in the scratch pad by one routine, LINPOS. The map is drawn using the same routines as were used in the RIVERS and STATIONS programs, except that the version is scaled down to fit in the allocated area. The active list is managed by the routine NUPLOT which both plots the appropriate symbol and outputs the plant number. Deleted entries in the active list are "stroked out" by routine XOUT.

The add (AD) command first reads the plant number entered at the terminal and uses it as an index into the array PLANT to echo the plant name to the terminal for verification by the user. If the number entered was correct, the plant number and the current station symbol number are inserted in the arrays PNUM and PSYM respectively. The screen display is then updated to display the stations where the plant is found and add the plant to the active list.

When the delete (DE) command is typed, the plant number entered is again used first to obtain the name of the plant to be echoed to the terminal. If the name is correct the plant number is located in the active list (array PNUM) and flagged as being deleted. When the next update takes place the deleted entries are actually removed from this list and the appropriate stations removed from the map display on the next refresh. Visually the deleted plant number is "stroked out" on the screen display until the refresh.

The NF (newfile) command first invokes a resave of the current active list if it has been altered since the last save took place. The arrays PNUM and PSYM are then reset to zero. Now if the specified file already exists, the data for these arrays is read in and displayed on the screen, otherwise the file is simply created.

The save (SA) command causes the arrays PNUM and PSYM to be output to the specified file.

The actions carried out when the PL (plot) command is typed are somewhat more complex than the other commands. The current active list is swapped out and the data from the specified file to be plotted is read into the arrays PNUM and PSYM. A call is then made to the PREVIEW package to switch from terminal to file mode. The usual calls to the map plotting routines are then made along with calls to special routines to plot the legend and title on the hardcopy plot. PREVIEW is then switched back to terminal mode and the old active list is swapped back in.

The SS (set symbol) command reads the symbol number typed and assigns this value to the current plotting symbol variable, STNSYM.

Typing RE (refresh) first causes an update of the active list to take place, removing all deleted entries. The screen is then cleared and the map redrawn on the screen using the old active list.

The DO (done) command checks to see if the active list needs to be resaved then terminates the program.

The "program" PLOTMAPS is a CMS execfile which submits the output file from the program to be plotted.

APPENDIX A

Digitizing the Perth County Map

1.1 Digitizing the Perth County Map

The Perth County map was digitized using the point-mode digitizer located in the Environmental Studies building at the University of Waterloo. Doug Dudycha is responsible for the allocation of time using this machine.

The digitized map data is recorded on magnetic tape that is readible using the IBM/360. Professor Dudycha will supply a FORTRAN program that will read the information from tape and reproduce it as a card deck. A WATFIV program can now be written to read this card deck into file in the desired format. If a transfer of data directly from tape to disk file is attempted some problems may be encountered, due to the formatting of the tape.

If the survey is to be expanded, say to include other counties, use of this digitizer is not recommended since it is a point-mode digitizer and the digitization of rivers becomes a long and somewhat erroneous process. Other digitizers exist on campus which would be better suited to the purpose.

APPENDIX B

Organization of Data

1.2 Organization of Data

1.2.1 Plant Data

files		contents
ADATA DATA A	-	plants with names starting with the letter A.
BDATA DATA A	-	plants with names starting with the letter B.
. . .		
. . .		
. . .		
UVWXYZDATA DATA A	-	plants with names starting with the letters U,V,W,X,Y or Z.
ENDF DATA A	-	a special file containing an end of plant data marker used by the program PERTH.

The format of the plant data files is as follows:

<plantname 1>

<num1><num2> . . . . . <num20>

<num21> . . .

.
.
.

<plantname 2>

<num1><num2> . . . . . <num20>

.
. etc.
.

where

<plantname i> ≡ the name of the i^{th} plant in the particular file. The plant name can be upto 50 characters long.

<num i> ≡ the number of the i^{th} station at which the plant is found. These numbers are arranged in rows with 20 entries in each.

1.2.2 Map Data

files		contents
BORDER DATA A	-	contains the map origin plus all points that make up the Perth County line, when joined sequentially.
INTERNAL DATA A	-	contains all points that occur at crossings or joins of township lines within Perth. The plotting of the township lines is accomplished by joining these points and using some of the border points, in a predetermined manner.
LATLONG DATA A	-	pairs of points which make up the latitude and longitude tick marks on the map.
TOWNAME DATA A	-	a file of points, one for each township name on the map, that designates the starting point of the first character in the name when plotting.
RIVER DATA A	-	contains the coordinate data that makes up the rivers in Perth County. This is the formatted source file of the river data (i.e. readable format). The format of each logical record in this file is as follows:

<segment #><# of points in segment>

<x1><y1>

<x2><y2>

. .

. .

. .

<xn><yn>

Each river in Perth County has been broken into logical segments of this form. The segment numbering scheme can be seen by referring to the map on page . This file is used only when editing river data or when creating a binary file.

RIVER BINDATA A - this file is the binary equivalent of RIVER DATA A and is using by the plotting subroutines to provide more efficient data transfer.

STATION DATA A - this formatted source file is made up of records of the following format:

<station i><xi><yi>
. . .
. . .
. . .

where xi and yi are the coordinates of station i on the map. All undefined station coordinates have their x and y set equal to 1.0. This file is used only for the purposes of editing and creation of a binary file.

STATION BINDATA A - this is the binary equivalent of STATION DATA A.

SURVEY COORDS A - this is the original coordinate data for the stations in Perth County (currently not used).

APPENDIX C

Calcomp Review Package

1.3 Calcomp PREVIEW Package

The Calcomp PREVIEW Package is used to allow both interactive graphics at a Tektronix Graphics Display terminal, in the programs RIVERS, STATIONS and PERTH, and hardcopy Calcomp plot production, in the program PERTH.

The PREVIEW package is used in conjunction with the Basic Calcomp Software txtlib PLOTTING. The following CMS command sets up the required text librarys:

GLOBAL < txtlib1 >< txtlib2 >< etc > PREVIEW PLOTTING

The order PREVIEW before PLOTTING in this command is important since all calls to routine PLOT must be serviced by the PREVIEW version of PLOT rather than the standard PLOTTING version. Thus, all calls to the Calcomp Software routines are serviced by PLOTTING but if they reference PLOT the PREVIEW version is used.

Besides the routine PLOT, the PREVIEW package contains 3 other routines, NEWPAG, HDCPY(FACTOR) and TSEND. A call to NEWPAG simply erase the screen. The routine HDCPY(FACTOR) allows you to switch to one of the three following modes, depending on the value of FACTOR:

FACTOR = 1 - send plotting data to both FILE HDCPY file and to terminal
= 0 - send plotting data to FILE HDCPY file
= -1 - send plotting data to Tektronix screen only.

The value of FACTOR is initially -1. The FILE HDCPY file is the standard output filename and should be setup using the following CMS command prior to using the package:

FI HDCPY DISK FILE HDCPY < fm >

The routine TSEND forces the contents of the plotting buffer to be sent to the terminal/file. On completion of an interactive plot, this routine should be called to ensure that all data making up the plot is sent to the terminal/file.

Before the FILE HDCPY file can be submitted for plotting, a REPLOT command must be issued at CMS level. The command format is as follows:

REPLOT FILE HDCPY < fm>

The function of this command is to modify the HDCPY file so as to appear as a valid plot file.

As is the case with most packages, there are some things which cannot be done, that under normal circumstances could be. For example, calls to routine OFFSET effectively do nothing. For a further description of "exception" cases refer to the on-line PREVIEWER documentation on VM.

APPENDIX D

Programs to adjust RIVER locations

1.4 RIVERS Program

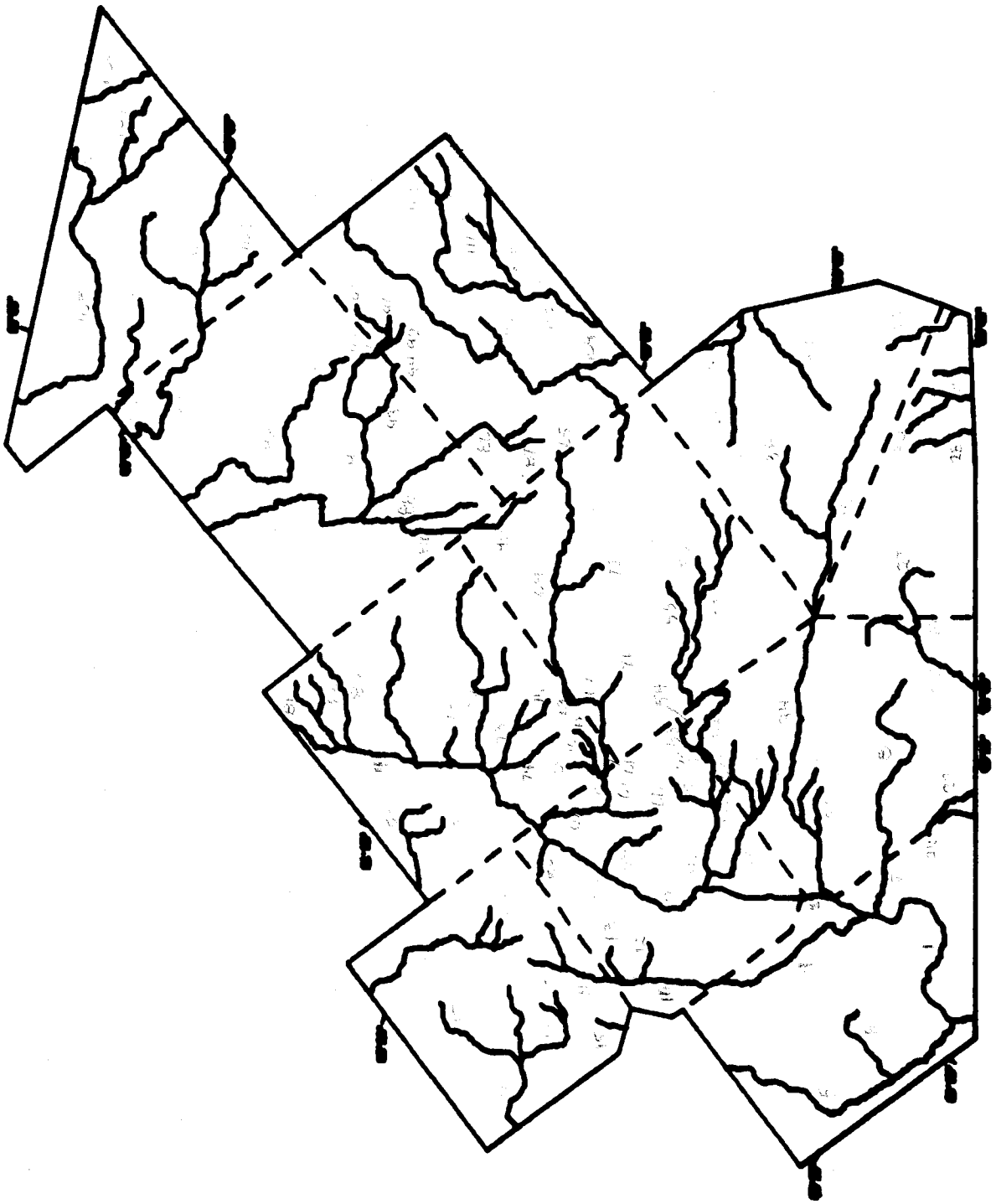
The program RIVERS provides an interactive means by which the rivers on the Perth County map can be re-positioned. The need for such a program became apparent when a plot of the digitized map and the original were compared and discrepencies were observed.

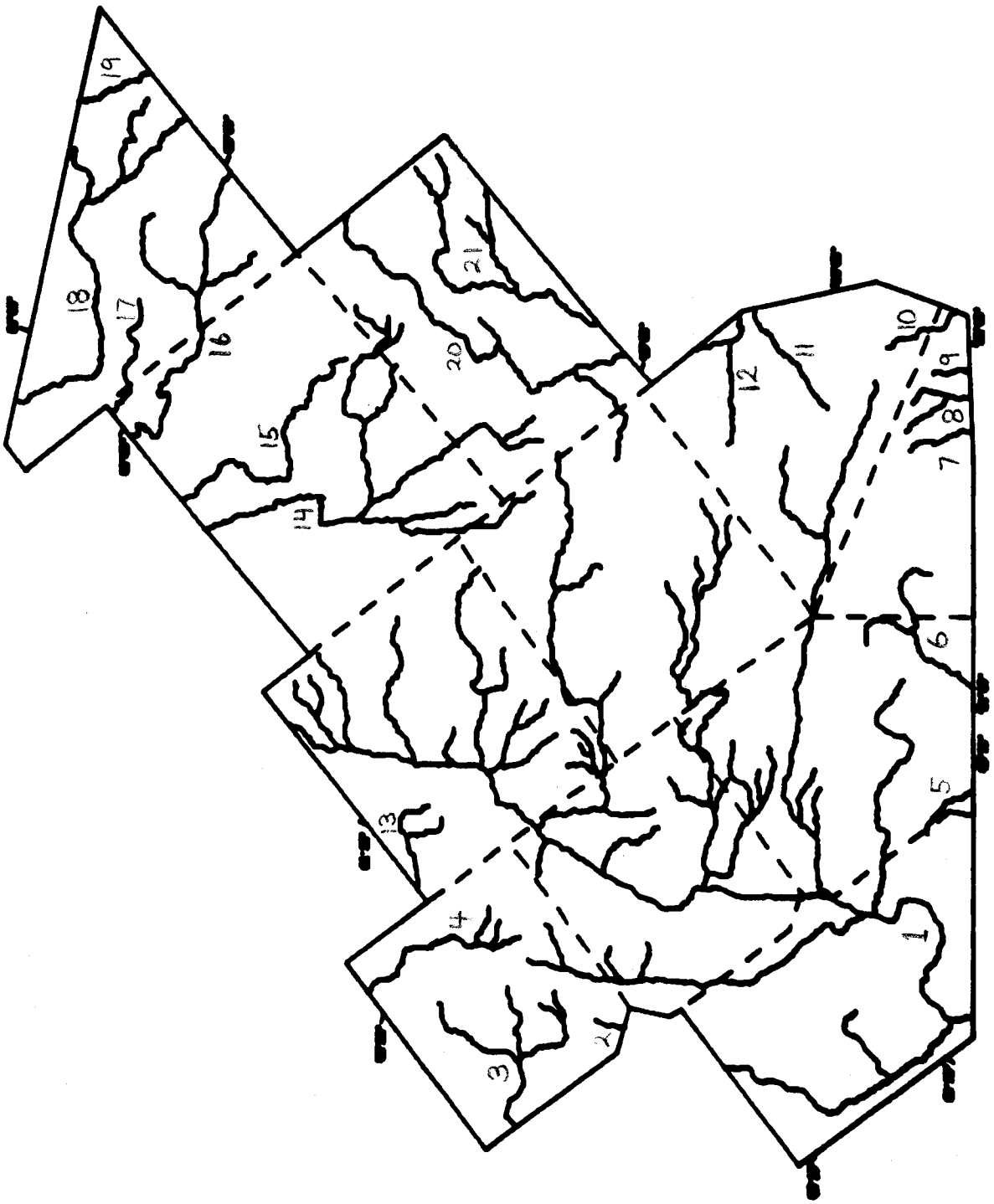
The rivers in Perth County have been sub-divided into 116 segments, each segment belonging to one of 21 river groups (systems). The map on page 10 shows the numbering of the segments and the map on page 11 shows the number associated with each river system. All references made to/from the program regarding river group numbers correspond to the numbers found on the latter of the above mentioned maps.

1.4.1 User Commands

When execution of the program has begun, the system prompts for the user to enter a command. This command may be any one of the seven valid commands to be now discussed.

The ENTER command is used to initiate the entering of x and y shifts for a particular river group. When ENTER is typed, the system responds first by asking for the number of the river group that is to be moved. If a valid number is now entered, the change in the x and y positions are requested. When this process is complete, the number of the next river group to be ENTERed is requested, etc. To return to COMMAND mode a river group number of Ø is entered.




19
18
17
16
15
14
13
21
20
12
11
10
9
8
7
6
5
4
3
2
1

To observe the changes in x and y ENTERed for the river groups, the LIST command is used. This command allows the user to verify the changes that are currently entered at any point in time. Note that only river groups with non-zero delta x's and y's are listed.

On occasion, it may become necessary to remove some of the changes that have been entered. This is accomplished easily by using the DELETE command. When DELETE is typed the system responds by asking for the river group number to be deleted. The net effect of this command is to delete the previously ENTERed change in x and y for the specified river group. As was the case with the ENTER command, entering a river group number of Ø causes return to COMMAND mode.

The RESET command, when typed, causes the program to be restored to it's original state at system startup time. Basically this involves the deletion of all changes in x and y that have been ENTERed.

When all changes have been ENTERed and have been verified to be correct, via the LIST command, the PROCESS command is typed. This causes all of the changes ENTERed to be incorporated with the old river data file to form the new updated river data file. It should be noted that each time that this command is used in a particular session, the same old river data file is used for processing. Thus all required changes must always be in the active list throughout the session.

The DISPLAY command produces a map, using the new river data file, on the Tektronix terminal screen. This allows the user to examine the map for any gross errors that have been made in repositioning the rivers.

When all changes have been successfully made and a new river data file produced, the session is terminated by typing the FINISH command. This stops program execution and returns the user to CMS. The new river data file produced is named NEWRIV DATA B.

Within the session these commands just mentioned may be used as many times as is required to produce the desired results. An important point to remember is that the changes ENTERed or DELETEd are not incorporated into the new river data file until the PROCESS command is typed.

A sample terminal session, a summary of the RIVERS program commands and a complete description of the error messages issued by the program have been included in the following pages.

```
                                                COMMAND ?
COMMAND ?                                       display
enter
ENTER GROUP NUMBER TO ENTER(0 TO QUIT)
1
ENTER CHANGE IN X
.05
ENTER CHANGE IN Y
.06
ENTER GROUP NUMBER TO ENTER(0 TO QUIT)
5
ENTER CHANGE IN X
-.3
ENTER CHANGE IN Y
0.6
ENTER GROUP NUMBER TO ENTER(0 TO QUIT)
0
COMMAND ?
list
GROUP NUMBER   DELTAX   DELTAY
      1          0.050    0.060
      5         -0.300    0.600
COMMAND ?
delete
ENTER GROUP NUMBER TO DELETE(0 TO QUIT)
5
GROUP  5 DELETED
ENTER GROUP NUMBER TO DELETE(0 TO QUIT)
0
COMMAND ?
list
GROUP NUMBER   DELTAX   DELTAY
      1          0.050    0.060
COMMAND ?
process
```


COMMAND ?

```
COMMAND ?
reset
COMMAND ?
list
GROUP NUMBER   DELTAX   DELTAY
COMMAND ?
finish
PROCESSING COMPLETE
NEW RIVER DATA IN FILE  NEWRIV DATA B
R:
```

1.4.2 Summary of Rivers Commands

Command	Meaning
LI(ST)	- list non-zero delta x's and y's for all river groups.
EN(TER)	- enter a delta x and y for a particular river group.
DI(SPLAY)	- display a map of Perth County using the updated river data.
RE(SET)	- reset all delta x's and y's to zero.
DE(LETE)	- delete (zero) the delta x and delta y for a particular river group.
PR(OCESS)	- process the inputed delta x's and y's and create an updated river data file.
FI(NISH)	- terminate the session.

In all of the above commands the bracketed portion is optional.

1.4.3 RIVERS: Error and Warning Messages

Type

1. INVALID COMMAND

   The command entered did not match any one of the seven valid commands. Check that the spelling of the command entered was correct and that it is actually a valid command, to correct this error.

2. READ ERROR ON UNIT 5

   A hardware read error has occured when a read was attempted from the terminal. The system will attempt to correct this situation by re-prompting for the last command/datum entered. If read errors persist, notify a hardware technician.

3. NUMBER OUT OF RANGE

   The river group number entered does not lie between 1 and 21, inclusive. Re-check the river group reference map to obtain the correct number that you wish to enter.

4. NO PROCESSING DONE YET

   An attempt to display a map of Perth County has been made before a new river data file has been created. To correct the situation type PR(OCESS) and then attempt the DI(SPLAY).

5. NUMBER PREVIOUSLY ENTERED

   The river group number just entered has been previously entered. Check to see if the number entered was actually the number desired. If this number is really the one you wish to EN(TER), first DE(LETE) this number and then re-attempt the EN(TER).

6. NOTHING DELETED

   The river group number specified to be deleted has never been entered.

   Check to see that the correct number was typed/selected.

7. INVALID INPUT

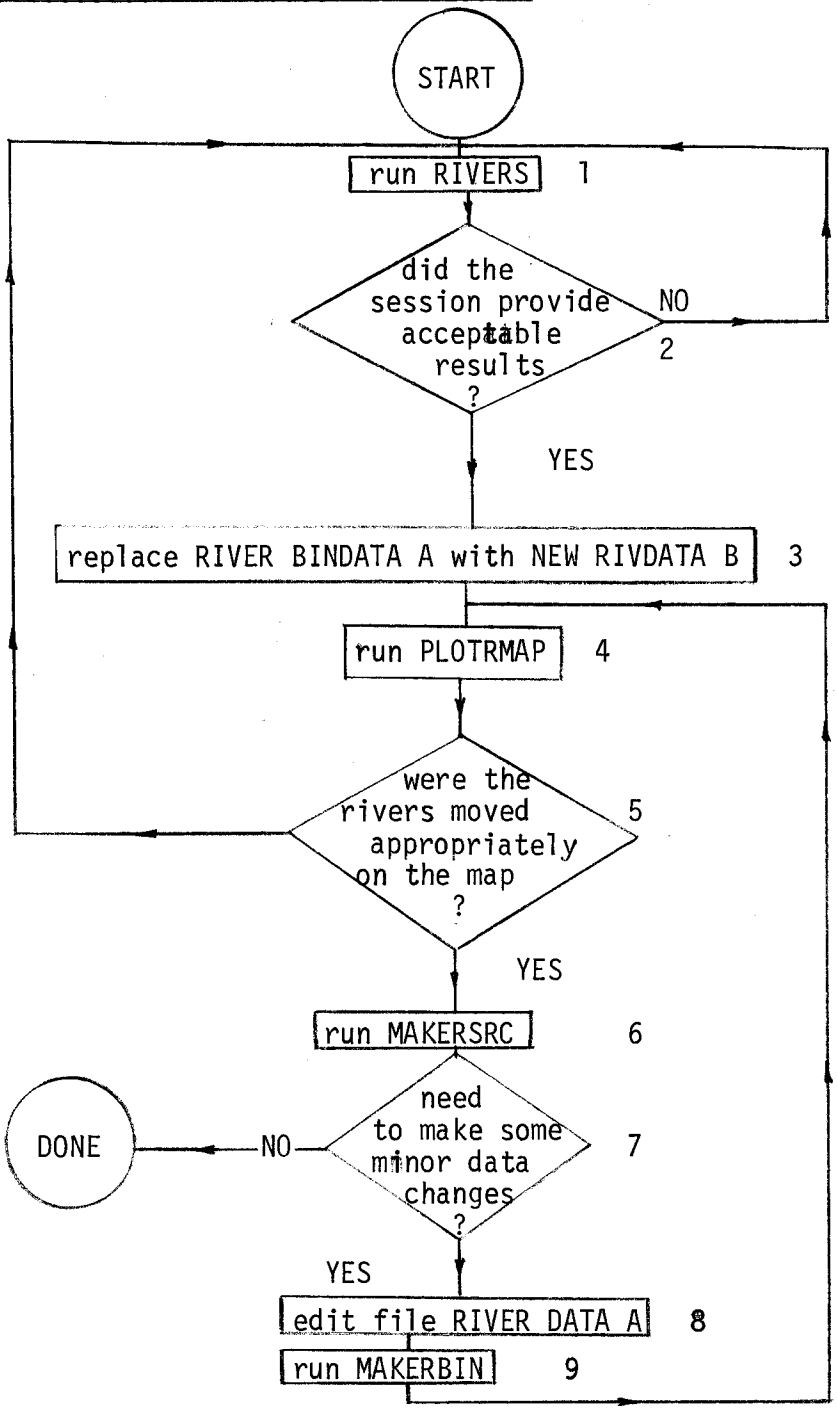
   Alphabetic input was entered at the terminal when numeric was expected.

   To correct, enter the numeric input requested.

1.4.4 Suggested Scheme for Updating River Data

The execution of other programs, besides RIVERS, and the carrying out of other tasks must be done in order to complete the "river moving" process. The following pages outline this entire process both diagramatically and verbally, and suggest a scheme which can be followed. A description of the functions of the other programs is also included.

SUGGESTED SCHEME FOR UPDATING RIVER DATA



1) The program RIVERS produces, as output, a binary file called NEWRIV DATA B which contains the new updated river data. River group (system) numbers may be obtained from the map on page

2) The user must decide whether or not the repositioning of the rivers was acceptable.

3) Since the 'river moving' session has provided acceptable results, the new data can replace the permanent data. This replacement can be carried out by issuing the following CMS commands:

```
ERASE  RIVER BINDATA A
COPY   NEWRIV DATA B RIVER BINDATA A
```

4) The program PLOTRMAP produces a map of Perth County using the updated river data on the Calcomp 30" plotter. The scale of the map is the same as the original.

5) At this point the user compares the newly plotted map to the original map and checks the validity of the changes made.

6) The program MAKERSRC produces a formatted source file, RIVER DATA A, from the binary river data file RIVER BINDATA A.

7) If the map produced needs no further improvements, the 'river moving' process is complete. Otherwise minor changes can be made by editing the river source file.

8) Edits may now be made to the file RIVER DATA A to correct minor shifts of segments over county or township lines, etc. Extreme care must be taken when editing since at the start of each river segment (in the source file) is a segment number and a count of the number of points in that segment. Thus if points are added or deleted from a segment, this count must be changed to reflect this. Segement numbers may be obtained from the map on

page 10. In order to be able to edit the large source file on VM, it will be necessary to define your Virtual Machine storage to approximately 800k. This can be accomplished by typing

```
CP  DEFINE STORAGE 800K.
IPL CMS
```

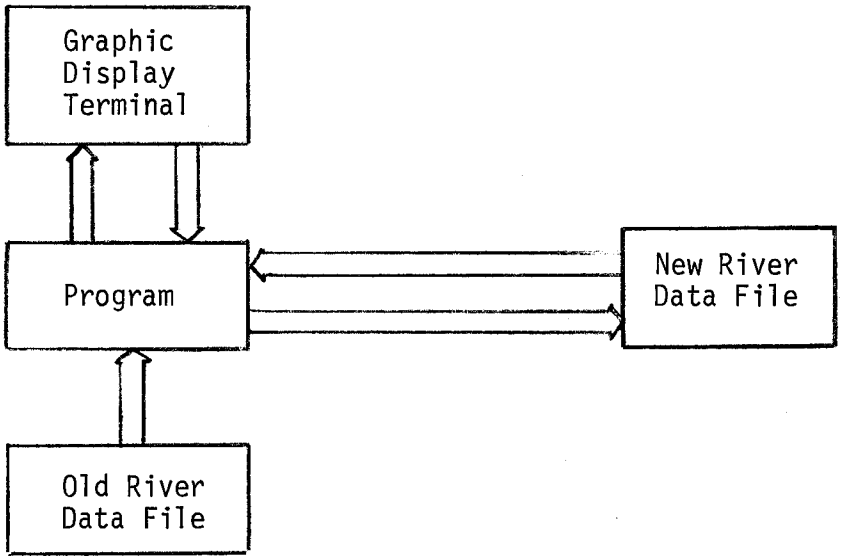
9) The program MAKERBIN produces a binary data file RIVER BINDATA A from the formatted source file RIVER DATA A. Edits made are incorporated into the binary file to be used by the plotting subroutines.

NOTE: to "run" a program simply type the program name
i.e. to run RIVERS, just type

```
RIVERS
```

1.4.5 Design, Implementation and Data Structures

The following data flow diagram best describes the requirements of the RIVERS program:



The program must be interactive and thus data transfer must occur between the terminal and the program and vice versa. Data from the old river data file must be accessible so that the program can incorporate the changes entered from the terminal into the new file. The data in the new file must also be accessible by the program in order to allow the user to graphically view the changes made to the data.

To support interactive terminal graphics the PREVIEW package (previously discussed) was chosen. Since this package is accessible only from FORTRAN, the decision to write the program in FORTRAN was not a difficult one. The format of the two files associated with the program is binary and was chosen so to increase data transfer speed.

To maintain "incore" information pertaining to the 116 segment into which the rivers are divided, the arrays RIVGRP, RIVBEG and RIVEND are used. (It should be mentioned now that at most one river segment resides "incore" at any one time). Each of the arrays is of length 116, one entry for each segment.

In the RIVGRP array the i^{th} entry is the number of the river group to which the i^{th} segment belongs. The RIVBEG and RIVEND arrays contain only 1's and 0's, indicating that the starting or ending point of a segment should be altered or not altered, respectively, when a segment is being repositioned. These arrays are used so that repositioning does not shift rivers over county or township lines and so that segments join together correctly.

It should be noted that the breakup of the rivers into segments has been done in such a way that by selecting the appropriate segments a map of any one of the townships in Perth may be easily made.

Two other arrays, DELTAX and DELTAY are also maintained "incore". Each of these arrays is of length 21, one entry for each river group. The i^{th} entry in DELTAX or DELTAY holds the x or y shift, respectively, that is required to correctly reposition the river group i. The entries in these two arrays are initially zero.

When the user selects a river group via the ENTER command, the program takes the change in x and y obtained and places them as that river groups entries in the DELTAX and DELTAY arrays. If a river group is selected via the DELETE command the appropriate entries for this river group in the DELTAX and DELTAY arrays are set to zero. The RESET command causes all entries in

the DELTAX and DELTAY arrays to be reset to zero. The LIST command simply causes the DELTAX and DELTAY arrays to be scanned for non-zero entries. If such entries are found, the river group number and the delta x and y are output to the terminal.

The actions carried out by the PROCESS command are somewhat more complex. The objective is to apply the changes that have been entered to the old data file and produce a new data file. The PROCESS routine works as follows:

1) Read a river segment into core.
2) Obtain the river group number for this segment from the RIVGRP array.
3) If the DELTAX and DELTAY array entries for this river group are zero, simply copy this segment to the new file, then go get the next segment, etc.
4) If either or both of the DELTAX or DELTAY entries for this river group are non-zero,
    i) update the starting point of the segment, if flagged as alter in the RIVBEG array.
    ii) update the ending point of the segment, if flagged as alter in the RIVEND array.
    iii) update the remaining points in the segment.
    iv) copy the updated segment to the new file.
5) Go and get the next segment (goto step 1) until all segments have been processed.

The PROCESS routine also sets a flag, PROFLG, to 1 indicating that a new river data file has been produced. This flag is used by routine DISPLY to ensure that no attempt is made to read/display a nul file.

The DISPLAY command invokes calls to the basic map plotting routines which read in their plotting data from the appropriate files, then draw the map. The river data displayed is obtained from the new data file.

The FINISH command just causes program execution to stop after outputing a information message to the terminal.

Three other programs exist to be used in the "river moving" process (PLOTRMAP, MAKERSRC, MAKERBIN) but deserve no explanation. The meer function (previously discussed) of these programs should suffice.

In all cases throughout this document, the terminology "RUN PROGRAM X" implies that "X" is to be typed at the terminal. All programs have their own CMSEXEC file of the same name. This fact has been made transparent to the user to avoid unnecessary confusion.

APPENDIX E

Program to adjust the STATION locations

1.5 STATIONS Program

The program STATIONS provides an interactive means by which the stations on the Perth County map can be re-positioned. Due to the fact that the positions of the stations on the map were not always correct, with respect to the original, this program became necessary.

Attached with each station is a unique station number. Currently there are 227 stations located across Perth County. These station numbers are required by the program when referencing a particular station. A map showing the numbers of the stations can be found on page 28.

Due to the similarity of this program to the RIVERS program, explanations may be brief in some cases.

1.5.1 User Commands

Seven commands (identical to RIVERS) are available to the user. These commands are described in the following paragraphs.

The ENTER command initiates the prompting for a station number and the change in x and y position to be applied to the specified station. To return to COMMAND mode a station number of 0 is entered.

The changes in the x and y positions of the station numbers ENTERed can be observed by typing LIST.

The DELETE command initiates the prompting for a station numbers to be deleted from the list of ENTERed stations. Again, typing a station number of 0 causes return to COMMAND mode.

The RESET command deletes all station numbers and their respective changes in x and y positions that have been ENTERed in the session. The program is thus restored to the same state as at system startup.


COMMAND ?

To incorporate the changes in the x and y positions of the ENTERed stations and produce a new station data file, the PROCESS command is typed. The same old station data is used in creation of the new file each time this command is used.

Typing DISPLAY causes a map of Perth County to be drawn on the screen. The station coordinate data used is obtained from the new station data file. Stations whose positions are currently not defined appear in the lower left hand corner of the screen.

To terminate a session the user simply types FINISH. The output from the program is found in a file named NEWSTN DATA B.

A sample terminal session, a summary of the STATIONS commands and a complete description of the error messages issued by the program have been included in the following pages.

```
                                                COMMAND ?
COMMAND ?                                       display
enter
ENTER STATION NUMBER TO ENTER(0 TO QUIT)
10
ENTER CHANGE IN X
.5
ENTER CHANGE IN Y
-.06
ENTER STATION NUMBER TO ENTER(0 TO QUIT)
99
ENTER CHANGE IN X
.7
ENTER CHANGE IN Y
.8
ENTER STATION NUMBER TO ENTER(0 TO QUIT)
0
COMMAND ?
lis
STATION NUMBER   DELTAX   DELTAY
      10          0.500   -0.060
      99          0.700    0.800
COMMAND ?
de
ENTER STATION NUMBER TO DELETE(0 TO QUIT)
10
STATION  10 DELETED
ENTER STATION NUMBER TO DELETE(0 TO QUIT)
0
COMMAND ?
list
STATION NUMBER   DELTAX   DELTAY
      99          0.700    0.800
COMMAND ?
process
```


POMMAND ?

```
COMMAND ?
reset
COMMAND ?
list
STATION NUMBER   DELTAX   DELTAY
COMMAND ?
huh
***ERROR*** INVALID COMMAND
COMMAND ?
fi
PROCESSING COMPLETE
NEW STATION DATA IN FILE  NEWSTN DATA B
R:
```

1.5.2 Summary of Stations Commands

Command	Meaning
LI(ST)	- list non-zero delta x's and y's for all stations
EN(TER)	- enter a delta x and y for a particular station
DI(SPLAY)	- display a map of Perth County using the updated station data
RE(SET)	- reset all delta x's and y's to zero
DE(LETE)	- delete (zero) the delta x and y for a particular station
PR(OCESS)	- process the inputed delta x's and y's and create an updated station data file
FI(NISH)	- terminate the session

In all of the above commands the bracketed portion is optional.

1.5.3 STATIONS: Error and Warning Messages

Type

1. INVALID COMMAND

   The command entered did not match any one of the seven valid commands. Check that the spelling of the command entered was correct and that it is actually a valid command, to correct this error.

2. READ ERROR ON UNIT 5

   A hardware read error has occured when a read was attempted from the terminal. The system will attempt to correct this situation by re-prompting for the last command/datum entered. If read errors persist, notify a hardware technician.

3. NUMBER OUT OF RANGE

   The station number entered does not lie between 1 and 227, inclusive. Re-check the station number reference map to obtain the correct number that you wish to enter.

4. NO PROCESSING DONE YET

   An attempt to display a map of Perth County has been made before a new station data file has been created. To correct the situation type PR(OCESS) and then attempt the DI(SPLAY).

5. NUMBER PREVIOUSLY ENTERED

   The station number just entered has been previously entered. Check to see if the number entered was actually the number desired. If this number is really the one you wish to EN(TER).

6. NOTHING DELETED

   The station number specified to be deleted has never been entered. Check to see that the correct number was typed/selected.

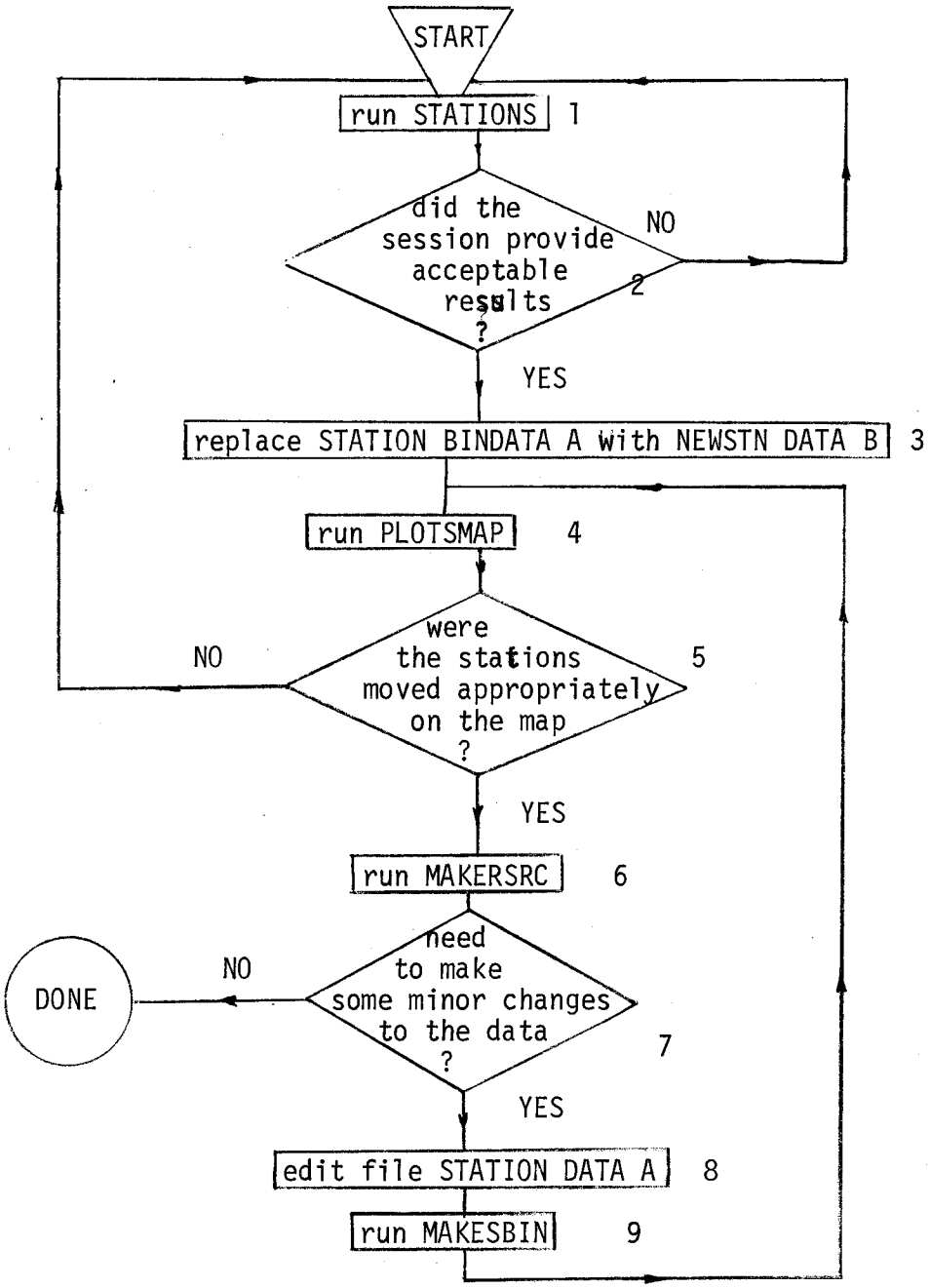
7. INVALID INPUT

   Alphabetic input was entered at the terminal when numeric was expected. To correct, enter the numeric input requested.

1.5.4 Suggested Scheme for Updating Station Data

To complete the "station moving" process, other programs must be run and tasks carried out. This sequence of operations is described both diagramatically and verbally in the following pages.

SUGGESTED SCHEME FOR UPDATING STATION COORDINATE DATA



1) The program STATIONS produces, as output, a binary file called NEWSTIN DATA B which contains the new updated station coordinate data. Station numbers may be obtained from the map on page

2) If the stations were moved successfully to their positions on the map, the user simply moves on to the next step. Otherwise the re-running of STATIONS becomes necessary.

3) To replace the old station coordinate data with the newly produced station coordinate data issue the following CMS commands:

```
ERASE  STATION BINDATA A
COPY NEWSTN  DATA B STATION BINDATA A
```

4) Running the program PLOTSMAP produces a map of Perth County using the updated station coordinate data on the Calcomp 30" plotter. Station numbers are plotted beside each station for easy reference.

5) The position of the stations on the plot may now be checked.

6) The program MAKESSRC produces a formatted source file STATIONS DATA A from the binary file STATIONS BINDATA A.

7) In most cases, it will <u>not</u> be necessary to continue on and edit the source file but steps 8 and 9 were added for completeness.

8) Changes can be made to the positions of stations by editing the STATION DATA A source file. This is basically the same operation that is carried out by the STATIONS program but will prove to be more difficult and susceptible to errors.

9) The program MAKESBIN produces a binary file, STATIONS BINDATA A, from the source file STATIONS DATA A.

1.5.5 Design, Implementation and Data Structures

The requirements that the STATIONS program must satisfy are very much the same as those of the RIVERS program.

The program must be interactive and thus terminal/program communications must be implemented. The old station coordinate data file must be accessible by the program so that the changes ENTERed at the terminal may be incorporated with it to produce the new data file. This new data file must also be read, at times, in order to allow the user to display the updated station data at the terminal.

The PREVIEW package was chosen to support terminal graphics and the language in which the code was written is FORTRAN. The old and new station data files are written in binary format to increase data transfer speed.

Two arrays, DELTAX and DELTAY, are maintained "in core", each being of length 227. This allows one entry for each station. The i^{th} entry in DELTAX or DELTAY holds the respective x or y shift required to correctly re-position station number i. Entries in these arrays are initially zero.

The ENTER command causes the change in x and y, typed by the user, to be inserted in the appropriate position in the DELTAX and DELTAY arrays for the specified station. When the DELETE command is entered at the terminal, the entries corresponding to the station number in the DELTAX and DELTAY arrays are zeroed. All entries in the DELTAX and DELTAY arrays are zeroed when the RESET command is typed. The LIST command causes the DELTAX and DELTAY arrays to be searched for non-zero entries. These non-zero entries and their corresponding station numbers are output to the terminal.

To service a PROCESS command the following actions are carried out:

1) Read the old station coordinate data into core.
2) Alter each x and y entry in the station coordinate data array by their corresponding entries in the DELTAX and DELTAY arrays.
3) Write out the new station data file.
4) Set a flag, PROFLG, indicating that a new data file has been written. The flag also indicates to the DISPLY routine that the file is no longer null.

The DISPLAY command causes a map of Perth County to be displayed on the screen using the new station coordinate data. The basic map plotting routines are called upon to accomplish this.

The FINISH command causes an information message to be output to the terminal then stops program execution, thus returning to CMS.

Due to the small size and function of the three other programs involved (PLOTSMAP, MAKESSRC, MAKESBIN) in the "station moving" process, their previously discussed function should be sufficient to understand their workings.