DLSPLOT: A COMPUTER PROGRAM FOR TRANSLATING DOMINION LAND SURVEY COORDINATES TO UNIVERSAL TRANSVERSE MERCATOR COMPATIBLE COORDINATES

by: A.T. Lytviak

1976

Alberta Research Council
Open File Report 1976-26

DLSPLOT:
A COMPUTER PROGRAM FOR TRANSLATING DOMINION LAND SURVEY COORDINATES TO UNIVERSAL TRANSVERSE MERCATOR COMPATIBLE COORDINATES

by

A. T. Lytviak

1976

Alberta
RESEARCH COUNCIL

CONTENTS

	Page
Abstract ...	1
Introduction ...	1
Acknowledgments ...	2
The coordinate system ...	2
Simplification of the UTM projection ...	7
Calculation of plotting coordinates ...	11
Instructions for use of DLSPLOT ...	16
References ...	19

Abstract

The computerization of many phases in the production of the reconnaissance hydrogeological map series has encountered difficulties inherent to mapping in a complex coordinate system. The National Topographic System, using Transverse Mercator projections, has polar coordinates. The Dominion Land Survey system is a synthesis of orthogonal and polar coordinate systems. The computer output device operates in an orthogonal system. If Dominion Land Survey coordinates are used as input and National Topographic Systems compatible maps are desired as output, mapping by computer must invoke a suitable translation program. The construction of such a program was undertaken by simplifying the coordinate system to the maximum extent consistent with keeping the resultant errors within set limits. The scale of 1:250 000 was used as a criterion for acceptability, as this scale is most commonly used in the reconnaissance map series. Simplicity of utilization was achieved by minimizing the parameters necessary to define the area to be plotted. Three parameters were found necessary and additional ones useful. These parameters were: horizontal field in ranges, the vertical field in townships, the scale as a ratio, and two adjustment factors.

Introduction

DLSPLOT is a computer program constructed to make it possible to plot geographic data by machine. The basic difficulty is that such data are located in the DLS system and have to be plotted in the UTM system. The goal was to make a program that would be general and demand only minimal knowledge of computer programming language and a basic understanding of the Dominion Land Survey coordinate system to produce acceptable plots of data. Acceptability of the plots was to be judged on the basis of fit to a National Topographic System 1:250 000 scale map.

In constructing such a program three coordinate systems must be considered: the terrestrial system of degrees latitude and longitude; the Dominion Land Survey System of meridians, townships, ranges, section, quarter sections and legal subdivisions; and the machine coordinates of inches vertically and horizontally. Of these only the last is orthogonal on a plane. The first is orthogonal on a sphere but suffers from distortion of direction and/or scale when represented on a plane surface.

Acknowledgments

For providing assistance with the programming, I wish to thank R. Bibby and W. Nielsen. Thanks are also due to L. M. Sebert of the Topographical Survey Directorate and R. E. Moore of the Surveys and Mapping Branch for assistance in explaining the NTS map projections. I thank R. Bibby, R. Vogwill, ... and .... for editing this paper.

The Coordinate Systems

The National Topographic System 1:250000 scale uses a Universal Transverse Mercator (UTM) projection. This projection may be imagined in the following manner. The sphere of the earth is cut by planes passing through its axis into sixty segments each six degrees of longitude wide. On the sphere's surface each plane forms a great circle through the earth's poles. If the surface of each segment is laid on a plane and contracted and expanded as necessary so that it is flat, one would have a UTM zone (Fig. 1a). The province of Alberta is contained within two such zones (Fig. 1b). Each

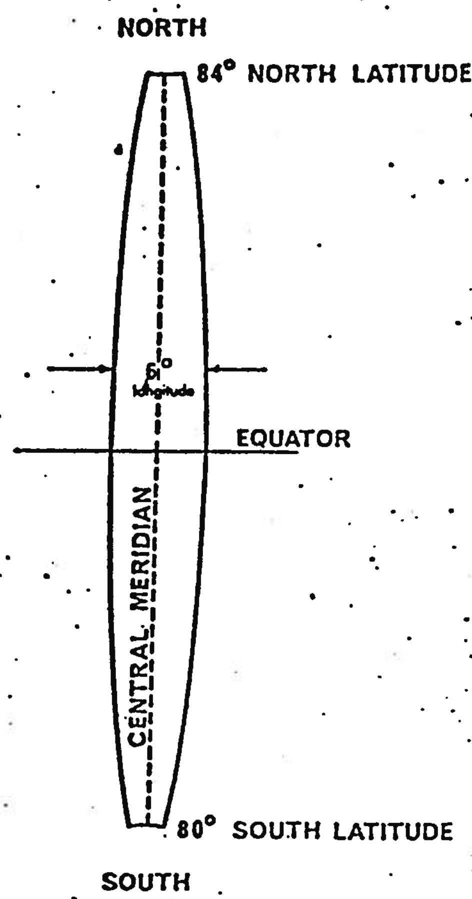


Figure 1. SHAPE of a U.T.M. ZONE

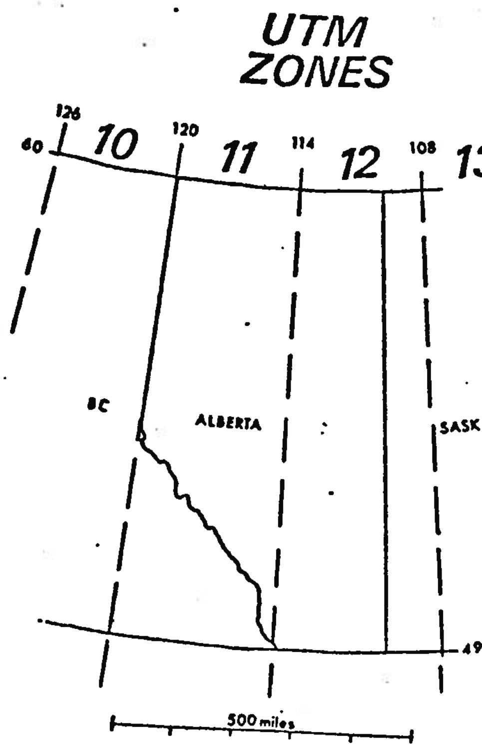


Figure 1b

zone in turn is subdivided into map sheets, the geometry and distortions of which are self-consistent within each zone. In Alberta the map sheets are two degrees of longitude wide and one degree latitude high (Fig. 2).

The Dominion Land Survey (DLS) system is a roughly plane orthogonal grid laid out on the earth's surface primarily for legal land holding description. This coordinate system is the most commonly used system in the province for specifying geographic locations. As a result of this, most hydrogeological data is located in DLS coordinates. In Alberta the land locations are referenced to the fourth, fifth or sixth meridians, which are north-south running reference lines (Fig. 2). Between the meridians are demarkated six mile wide, north-south running strips known as ranges. Ranges are numbered westward from each meridian. The province is divided, as well, into six-mile wide strips that run east-west. These, known as townships, are numbered northwards starting at the 49° parallel of latitude. Each six mile by six mile grid thus laid out is also, confusingly, called a township and is subdivided into 36 one mile by one mile squares. These, known as sections, are numbered sinusoidally starting in the bottom right-hand corner. That is, the bottom row is numbered right to left, the one above it is numbered left to right and so on until section 36 is reached in the upper right hand corner (Fig. 3). The sections are further subdivided into half mile squares producing four quarters or into 16 quarter mile squares known as legal subdivisions (LSD). LSD are also numbered sinusoidally, the first being in the bottom right-hand corner.

Thus in Alberta a land location or description usually consists of giving a meridian, a township, a range, a section and a quarter section or LSD.

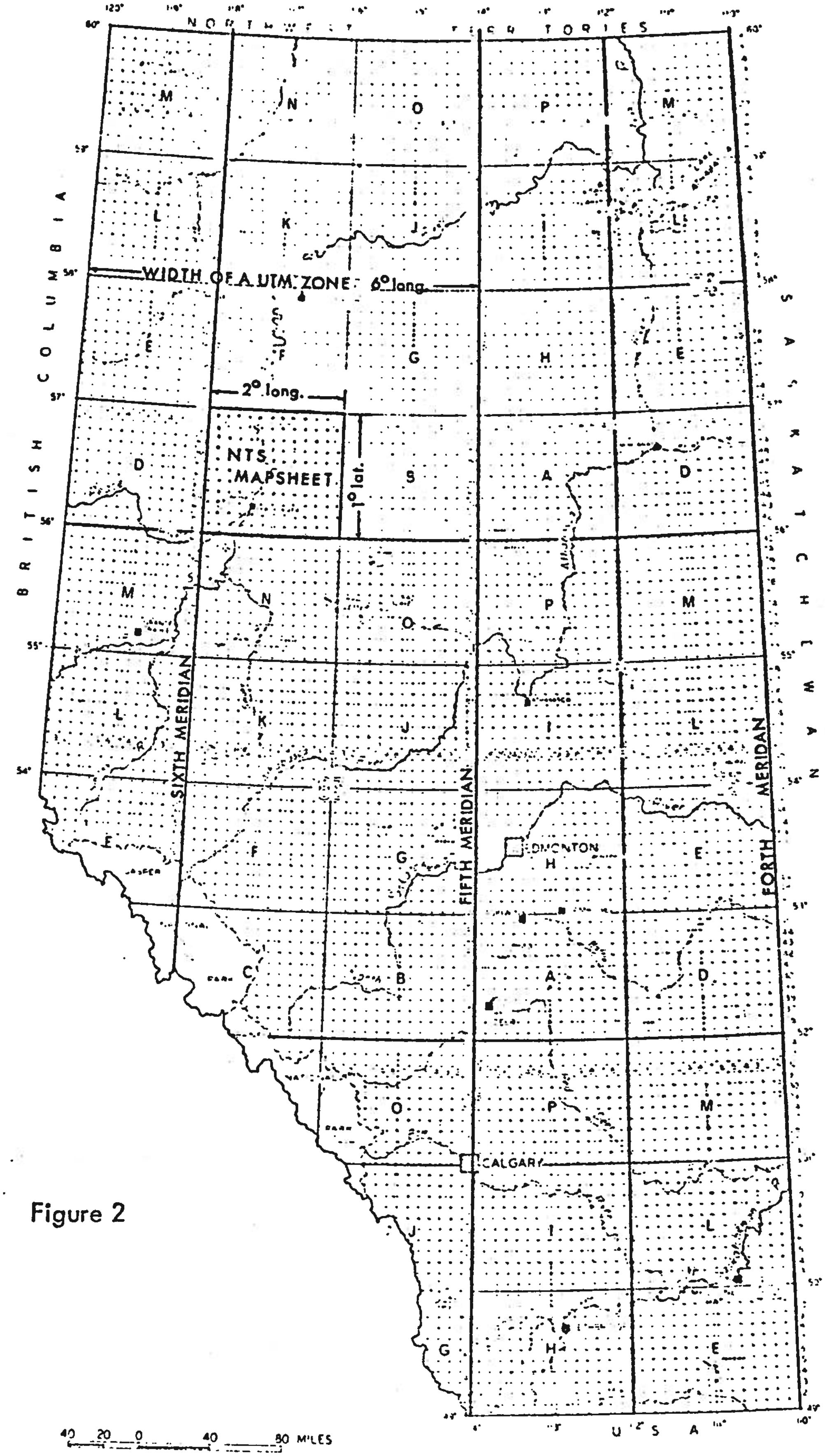

NORTHWEST TERRITORIES
BRITISH COLUMBIA
SASKATCHEWAN
WIDTH OF A UTM ZONE 6° long.
2° long.
1° lat.
NTS MAPSHEET
SIXTH MERIDIAN
FIFTH MERIDIAN
FORTH MERIDIAN
EDMONTON
CALGARY
U S A
40 20 0 40 80 MILES

Figure 2

SUBDIVISION OF A TOWNSHIP

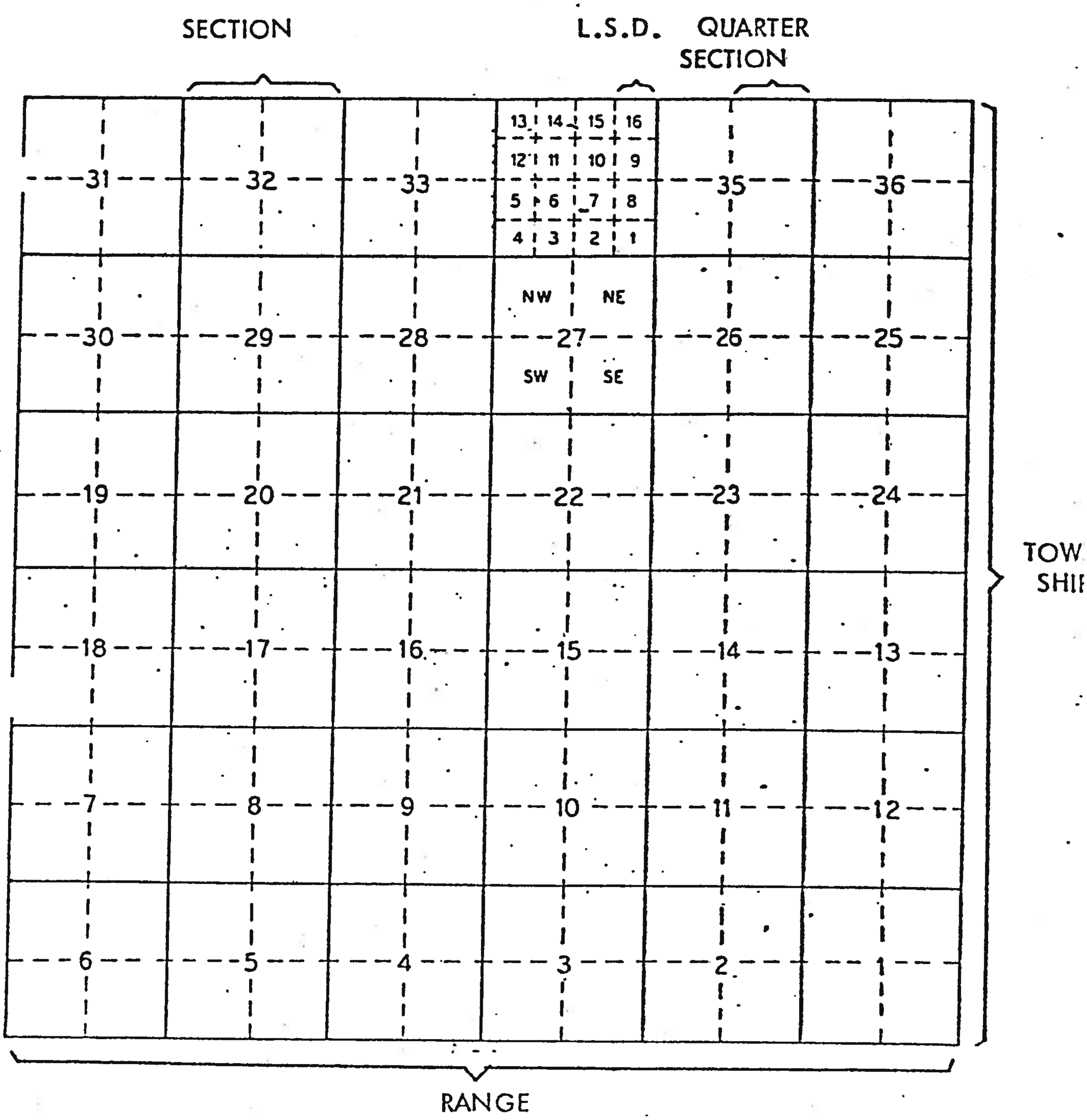


Figure 3

Simplification of the UTM Projection

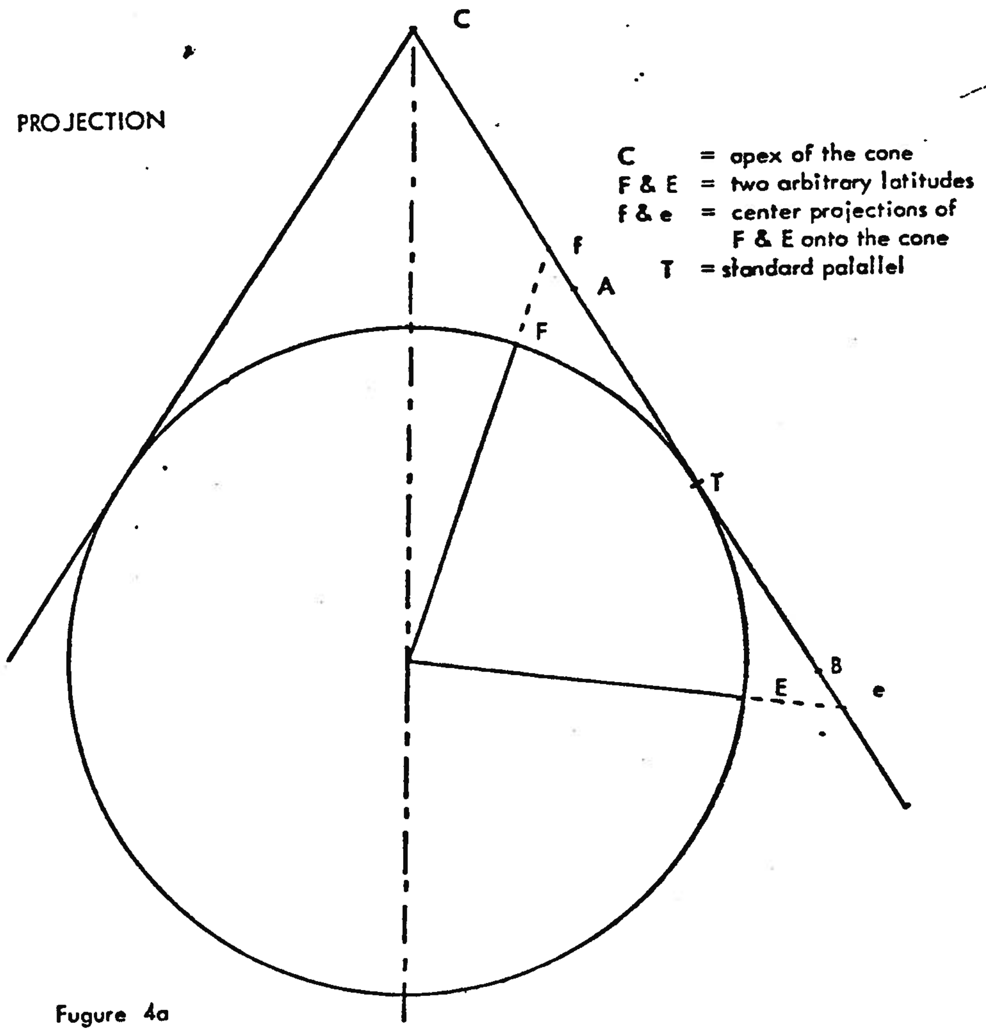
It is quickly apparent that the geometry of a Universal Transverse Mercator projection is relatively complex as both lines of latitude and longitude are represented as curved lines. From actual map sheets it appears that the lines of longitude very nearly approximate straight lines. In fact along the edge of a UTM zone, between the latitudes of 50° and 51°, the separation of the curve and a straight line is only 1.8 meters.* In the context of our reconnaissance work, this is negligible. Discounting this curvature one is left with a projection which appears to closely approximate an axial conic projection. (That is, a projection of the earth's surface on to a cone coaxial with the earth's geographic axis.)

Calculating from the curvature of lines of latitude and the convergence of the lines of longitude it was found that the apex, C (Figs. 4a, 4b), of the cone of projection would be about 3090 miles north of the 49th parallel of latitude. This distance is measured on the surface of the developed, unrolled cone. The standard parallel, the latitude at which the horizontal scale is true, was arranged to run through the center of the map sheet to be plotted. All east-west lines on such a projection are represented by arcs of constant radius and north-south lines by radii of the developed cone. This "projection" is strictly speaking not a projection, as points F and E would, if center-projected onto the cone, project as points f and e (Fig. 4a); whereas actually, the distance between points A and B (Fig. 4b) which represent F and E (Fig. 4a) is made equal to the arc distance between F and E. Thus north-south scale remains constant throughout the developed cone. The east-west scale increases away from

---

*Personal correspondence with L. M. Sebert, Head, Mapping Program Section, Topographical Survey Directorate.

CONE OF PROJECTION



Fugure 4a

DEVELOPED (UNROLLED) CONE OF PROJECTION

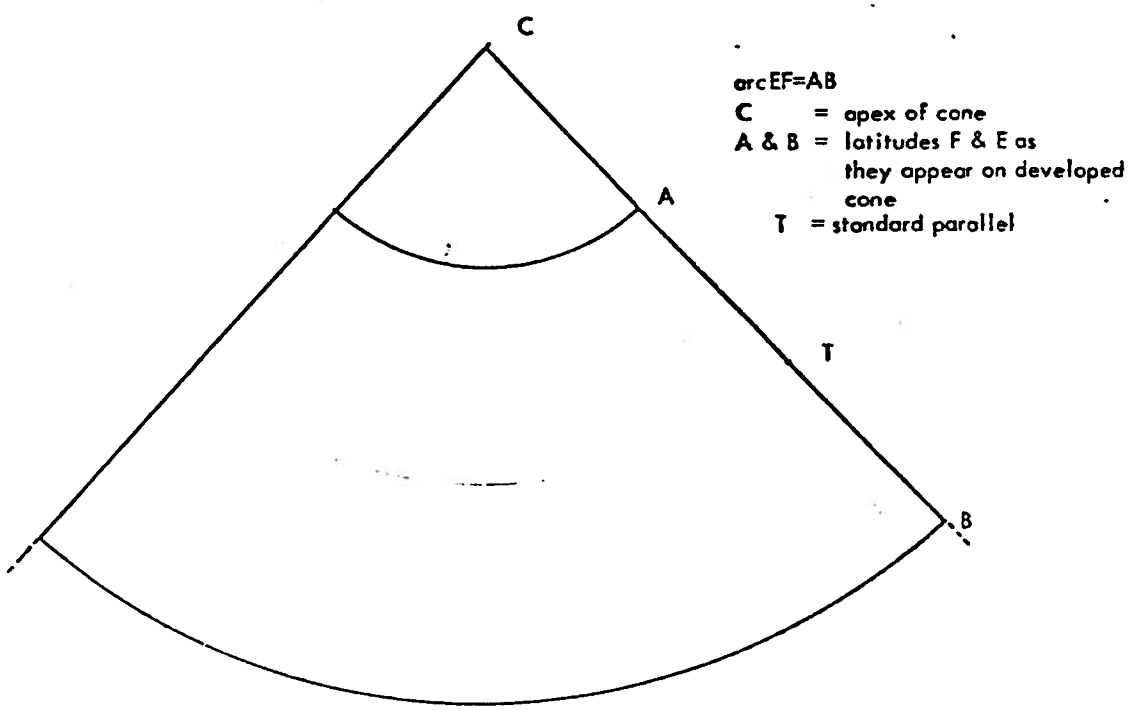


Figure 4b

the standard parallels as a function of the divergence of the radii of the cone and the sphere, at a given latitude. This function was approximated by increasing horizontal scale as a function of the sphere and decreasing it as a function of the cone in such a way that true scale was achieved at the center of the map sheet to be plotted. This was tested and found adequate in map sheets up to 15 townships wide.

A difficulty arises in the laying out of the north-south running range lines. The fourth, fifth and sixth meridians coincide with the lines of longitude 110°, 114° and 118°, respectively. Due to the decreasing circumference of the earth at increasing latitudes, any lines of constant longitude converge poleward. Range lines are referred initially to a meridian then stepped off in 6-mile strips westward. If this was done without adjustments of some sort, the range lines, excepting the meridian, would diverge increasingly from the lines of longitude westward and northward. The difficulty was overcome by establishing east-west reference lines called baselines. The 49° parallel of latitude was chosen as the first baseline. Subsequent baselines were laid out at 24-mile intervals northwards. The strips between baselines were subdivided into four 6-mile strips, thus the first and every fourth subsequent township is bordered on the south by a baseline.

Each baseline is subdivided into 6-mile segments starting at a meridian and going west to the next meridian. Lines of constant longitude, north-south lines, are then passed through these 6-mile divisions to establish ranges. Each such range line extends 12 miles (two townships) south and north of the baseline. The range lines projected from adjacent baselines meet halfway in an offset known as a correction line. Thus, this offset results from laying an approximately plane-orthogonal grid on a spherical surface keeping orientations sphere-orthogonal (Fig. 5). This offset can be

DOMINION LAND SURVEY GRID

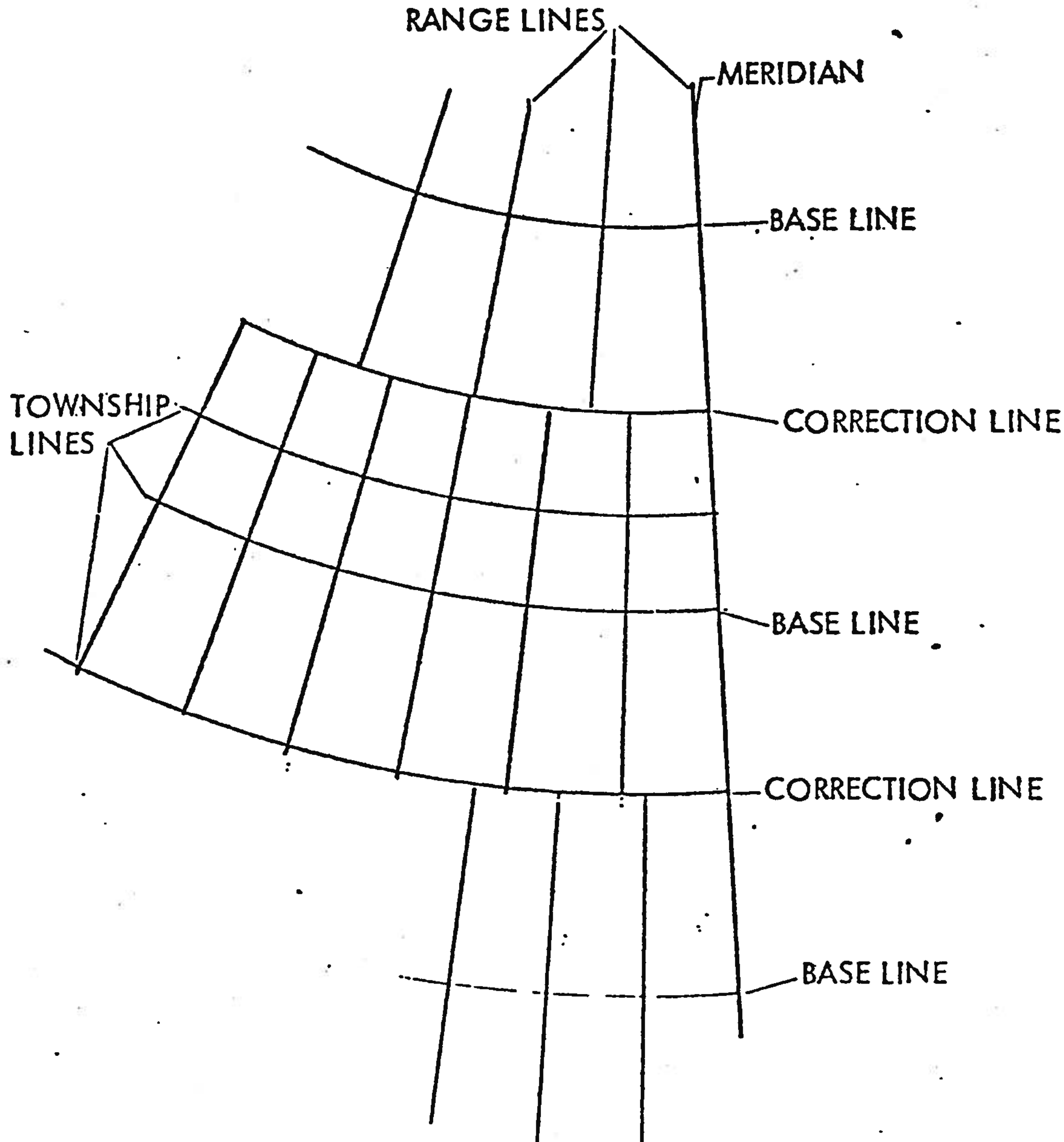


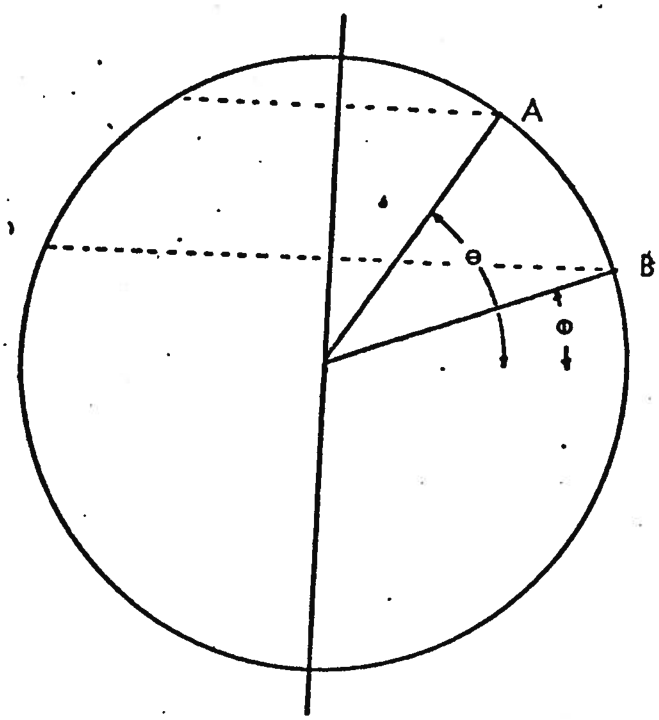
fig. 5

calculated as a cosine function of latitude (Fig. 6) and in the program it is handled as such, eliminating the need to specify the magnitude and location of each correction.

Calculation of Plotting Coordinates

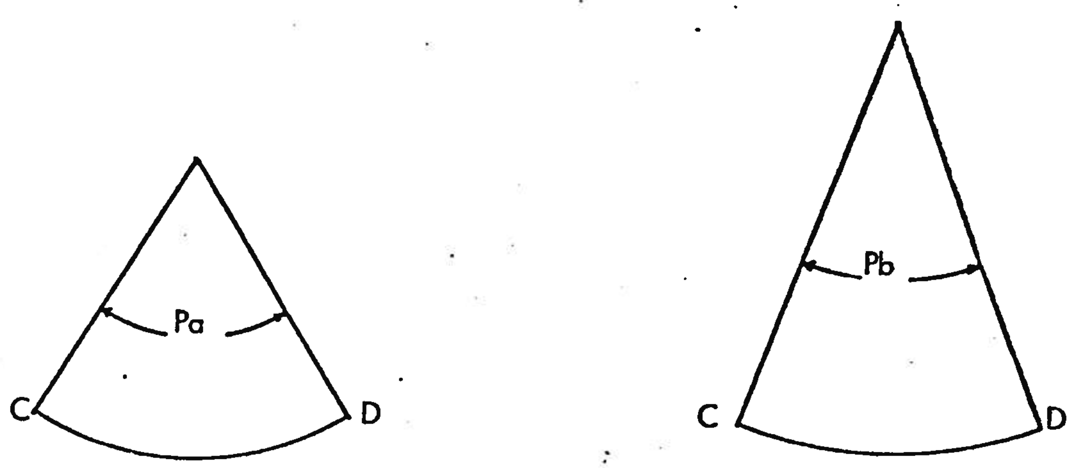
In the program initially all the coordinates are considered as orthogonal and are referenced to the meridian, which is considered vertical, and the horizontal center line of the plot. DLS coordinates are converted to miles west of the meridian and miles north and south of the horizontal reference line. This treatment produces an orthogonal grid of the township and range lines (Fig. 7a). To produce offsets analogous to correction lines, their horizontal coordinates are then multiplied by the cosine function mentioned. This is done in steps. That is, each horizontal strip of 24 miles is multiplied by the factor which corresponds to the latitude of the baseline relevant to that strip. This treatment produces an orthogonal grid of township and range lines in which the horizontal scale increases upwards in discrete jumps every 24 miles (Fig. 7b). The scale will now be true only on the horizontal center line. Next the scale is decreased upward. This decrease is done as a continuous function of the height of a cone in such a manner as to make no scale change at the horizontal center line. The height of the cone of projection is used for this calculation (Fig. 7c).

All coordinates generated up to this point are x and y orthagonal coordinates (Fig. 8a); the x-coordinate being referenced to the relevant meridian, the y-coordinate being referenced to the horizontal reference line. The next step is to curve east-west lines by converting to polar coordinates. This is done in two steps. First, x components are considered to be arc lengths and y components to be lengths in the radial direction. Second, all coordinates are put into polar form, angles and radii. The horizontal



Ca circumferance at latitude A
Cb circumferance at latitude B

$$\frac{Ca}{Cb} = \frac{\cos \Theta}{\cos \Phi}$$



Pa angle subtended by arc CD at latitude A
Pb angle subtended by arc CD at latitude B

$$\frac{Pa}{Pb} = \frac{Cb}{Ca} = \frac{\cos \Phi}{\cos \Theta}$$

Figure 6

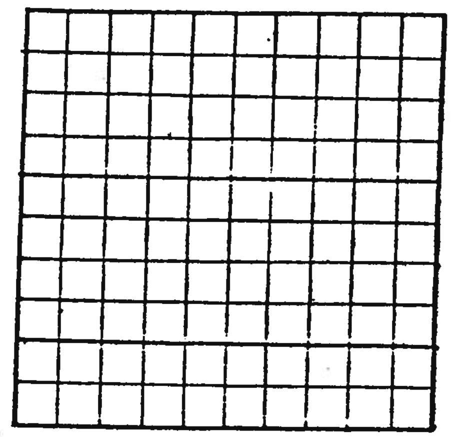


Figure 7a Original township – range grid.

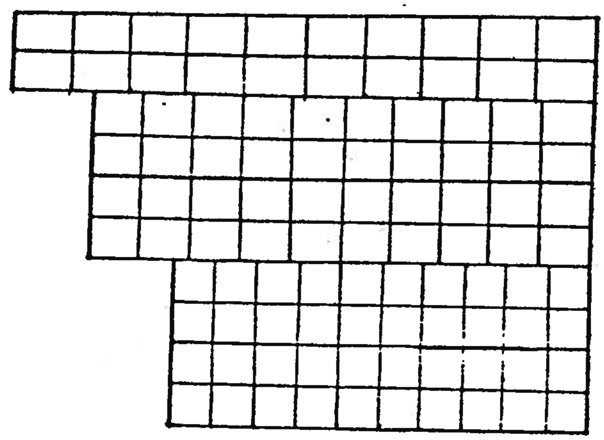


Figure 7b Effect of subroutine XWARP on an orthogonal grid. Scale increased upward as a function of the circumferance of a shere with increasing latitude.

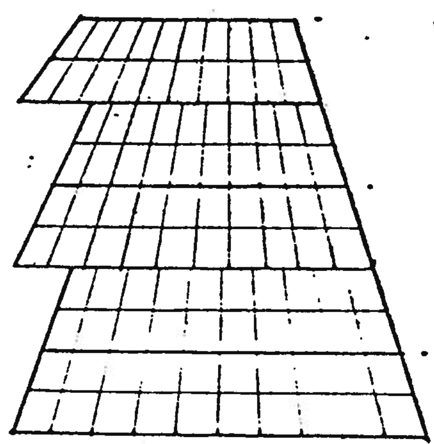


Figure 7c Effect of subroutine ACONE on grid produced by XWARP. Scale decreased upward as a function of the height of the cone of projection.

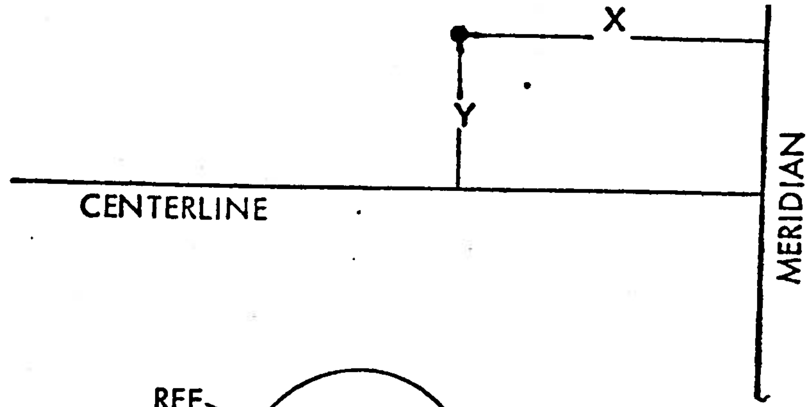
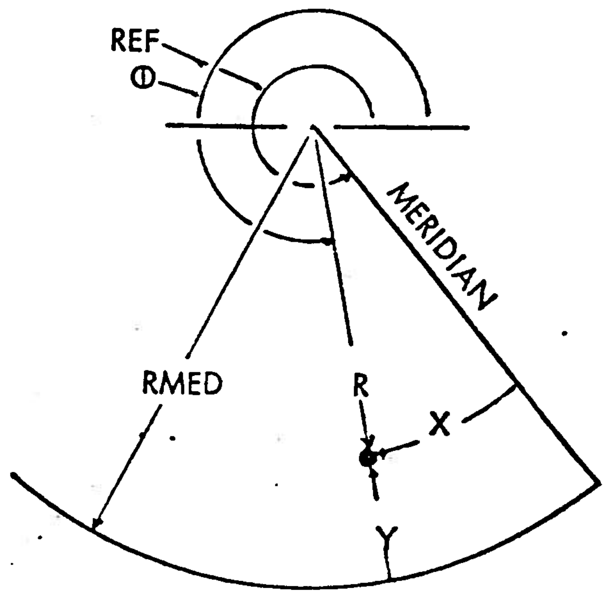
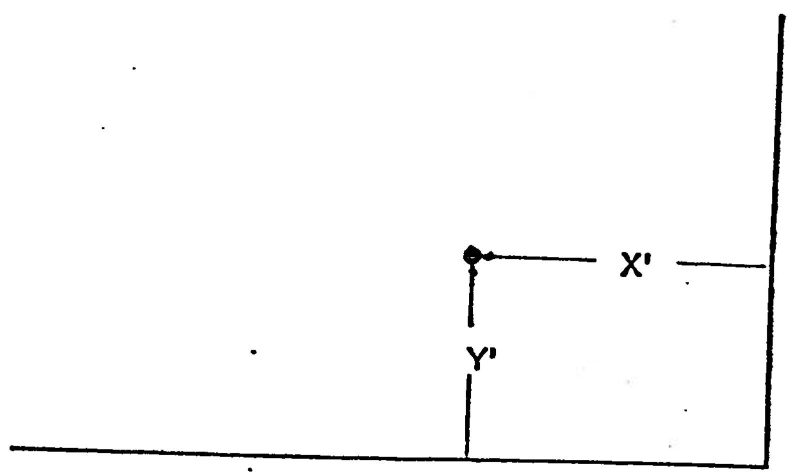


Figure 8a



$$R = RMED - Y$$
$$\Phi = REF - X/R$$

Figure 8b



$$X' = R\cos\ \Phi + x_c$$
$$Y' = R\sin\ \Phi + y_c$$

ARBITRARY X', Y' AXES

Figure 8c

center line becomes an arc at distance RMED from the center of projection (C). The relationship used for radius is R = RMED-y; R being the radial distance (miles) of the point from C. The arcs representing x components are converted to angular measure using R as a radius (Fig. 8b). Angular distance, Ø, is calculated using the relationship Ø = REF − x ÷ R.

For angular distance Ø = REF − x ÷ R; where Ø is the angle and REF is a radial reference line established in such a manner as to center the projection around the 3/2π, 270° radial. The resulting polar coordinates are then translated back to orthogonal coordinates for machine plotting, using the relationships x' = RCOS Ø + x_{adj} and y' = RSIN Ø + y_{adj} (Fig. 8c). The reference axis are translated using factors x_{adj} and y_{adj} to result in a plot that minimizes paper used in the plot.

Since each map sheet has a slightly different geometry from that produced by the initial program, two adjustment parameters were added: one to adjust the scale in the radial direction and one to adjust it in the tangential direction. This parameter is a factor which is approximately equal to unity and is linearly and directly proportional to desired scale changes.

This program has been tested on several existing data files and acceptable plots generated. The testing has so far been limited to 1:250 000 scale maps. An accurate method for predetermining the adjustment factors was found.

Further work using this program as a base should lead to a program that uses the new University of Alberta plotter and produces multiple colored plots as well as legends, titles, scales, etc. As well, the calculation of adjustment factors should be eliminated in the next program.

The program DLSPLOT as it now stands appears to have achieved its aims of simplicity and flexibility of operation.

Instructions for Use of DLSPLOT

The program DLSPLOT prompts the user for input, as well as giving him various choices of what is to be plotted. For example a DLS grid, a symbol or a value may be plotted. Also, any combination of these in various sizes can be chosen.

In the following description the computer written lines will be preceded by a dash.

Before running this program it is necessary to have one or two files at hand. You must specify which file the plot coordinates are to go into. For this explanation "-PL" will be used as the output file. Also, you must have the data to be plotted in a file; "DATA" will be used. The following sequence starts the program:

RUN DLSPLOT+*PLOTLIB 7=DATA a=-PL

After this the machine will type a number and the program will begin to execute thus:

-DLSPLOT ROUTINE This is an identifier.

-FIRST TOWNSHIP I3 The machine asks which is the first township to be plotted. I3 indicates the format in which the reply must be given.

001 This would be your reply for township 1.

-LAST TOWNSHIP I3

010

-FIRST RANGE I2

14

-LAST RANGE I2

28

Thus far you have told the machine that you are dealing with an area from township 1 to township 10, from range 14 to range 28. Next, the machine will need the scale in which it is to deal:

-SCALE 1: F8.1

250000. Be careful to get the decimal point in the correct place.

The machine will then ask for the adjustment factors. A quick and accurate method of determining adjustment factors consists of measuring the distance spanned by four ranges and four townships. This is then compared to the same distance on the uncorrected plot. The factor will consist of the map distance divided by the corresponding plot distance. It has been found that an uncorrected 1:250000 plot spans (24.2÷ 4) inches in four townships or four ranges. The program then offers a choice of plotting a grid or not plotting a grid:

-RUN GRID? YES=1, NO=0

1

This response will result in the plotting of a township- range grid.

After this part of your input will be typed out by the machine for visual checking. Also an estimate of the length of the resulting plot will be printed. A choice will next be offered. The choice will consist of retyping your input (AGAIN), going to the next step (YES), or discontinuing the program (NO). If the number corresponding to "YES" (22) is typed in you will be asked if you would like to plot a symbol. Receiving an affirmative answer the machine will ask you which symbol. You must answer with the three digit integer corresponding to the symbol of your choice. The symbols and corresponding integers are to be found in the U of A CALCOMP Manual. Next you will be asked the size, in inches, you would like this symbol plotted. The machine will then ask you if you would like a number, or value plotted. If you answer in the affirmative, it will ask you what size these numbers are to be, what rounding you would like, and the angle at which you would like them plotted.

The amount of rounding is given by the number of digits right of the decimal you would like. For example, a rounding of 0 gives you the integer portion of your number, followed by a decimal point. If you have answered in the affirmative for the symbol or the number the machine will ask you by what format these are to be read from your input file; in this case "DATA". The reply must include the brackets of a format statement. For example: (7x, 6F5.0). You will then be given a choice of reentering this last input. If you would like to terminate the program at this point, indicate you would like neither a symbol nor a number. Replying that you would like to ~~go on~~, the machine will respond with a set of numbers indicating that it has completed the program.

The entire plot will now be in your output file, in this case "-PL". To get the plot put onto paper enter the following line:

RUN *CALCOMPQ SCARDS = -PL

The machine will respond with a set of number, then a receipt for the plot and its estimated dimensions, followed by another set of numbers. Your plot will then be processed and will become available at the site of the plotting machine.

Note that the machine's plot length will be greater than that estimated during the input stage of the program. This is because in addition to the map to be plotted, an identifying label is automatically added to each plot, resulting in an increase ~~of three to four inches for~~ the output.

References

Bygot, J. (1960): An introduction to mapwork and practical geography; 7th Revised Ed., London Univ. Tutorial Press, London, 264 pages.

Dectz, C. H. and O. S. Adams (1945): Elements of map projections with applications to map and chart construction; 5th Revised Ed., U.S. Gov't. Print. Off. Spec. Publication 68, Washington, D.C., 173 pages.

Dickinson, G. C. (1969): Maps and air photographs; E. Arnold, London, 286 pages.

Melluish, R. K. (1931): An introduction to the mathematics of map projections; Cambridge Univ. Press, Cambridge, 145 pages.

Raisz, E. (1962): Principles of Cartography; McGraw-Hill, New York, 315 pages.

Sebert, L. M. (1970): Every square inch; Department of Energy, Mines and Resources, Ottawa, 25 pages.

Steers, J. A. (1965): An introduction to the study of map projections; 14th Revised Ed., Univ. London Press, London, 292 pages.

Sample Runs of DLSPLOT

Five sample runs were made on an existing file, ATOBJ, which contained LSD coordinates of data points in NTS map sheet 73L, Sand River. A portion of this map sheet consisting of the area within townships 58–63, ranges 1–5, was plotted, changing parameters on each run. The sixth run shows reaction of the program to several input errors.

Table 1 is a portion of the input data. The numbers plotted on some of the sample runs were the apparent safe yields, Q_{20app}, in ATOBJ.

Table 1

Partial listing of input file ATOBJ

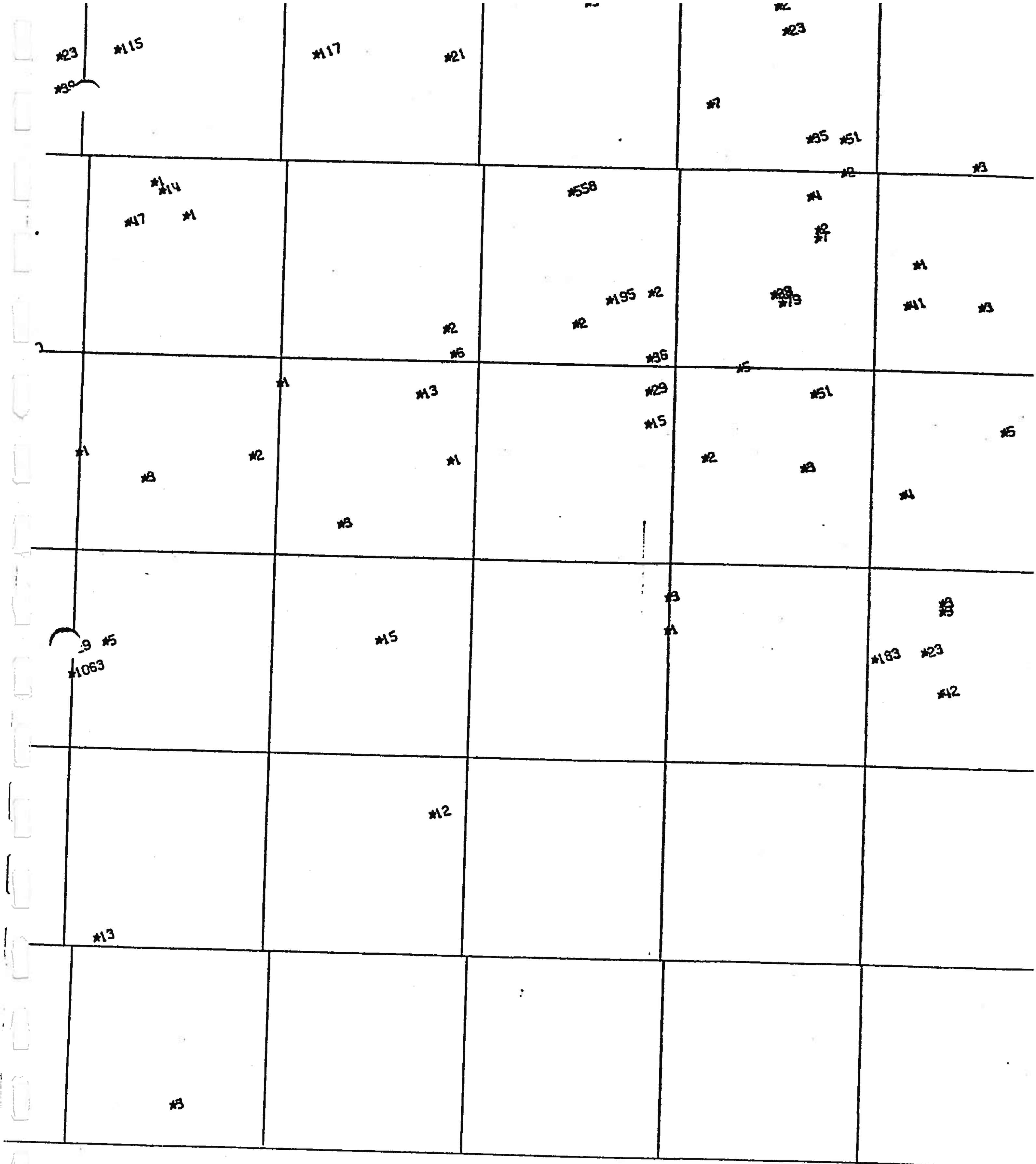
LEGEND	M	T	R	S	LSD	Tapp	220a
1	4.0	58.0	5.0	10.0	4.0	697.1	3.3
2	4.0	58.0	5.0	10.0	4.0	1038.2	4.9
3	4.0	58.0	6.0	7.0	4.0	403.5	11.3
4	4.0	58.0	7.0	1.0	13.0	415.7	2.8
5	4.0	58.0	7.0	6.0	4.0	1466.5	25.7
6	4.0	58.0	7.0	15.0	4.0	36.9	0.9
7	4.0	58.0	7.0	25.0	1.0	412.0	1.0
8	4.0	58.0	8.0	9.0	1.0	328.9	9.4
9	4.0	58.0	8.0	10.0	16.0	75.9	4.2
10	4.0	58.0	8.0	14.0	4.0	122520.5	11622.8
11	4.0	58.0	8.0	22.0	13.0	912.1	9.5
12	4.0	58.0	9.0	1.0	1.0	1344.9	148.5
13	4.0	58.0	9.0	2.0	13.0	805.1	35.1
14	4.0	58.0	9.0	6.0	15.0	3814.9	115.7
15	4.0	58.0	9.0	7.0	16.0	145.4	0.4
16	4.0	58.0	9.0	7.0	16.0	704.5	67.8
17	4.0	58.0	9.0	7.0	4.0	354.0	7.9
18	4.0	58.0	9.0	9.0	16.0	3155.9	239.3
19	4.0	58.0	9.0	16.0	1.0	1186.7	52.3
20	4.0	58.0	9.0	16.0	16.0	136.2	6.1
21	4.0	58.0	10.0	1.0	13.0	806.9	51.2
22	4.0	58.0	10.0	5.0	1.0	1469.1	27.9
23	4.0	58.0	10.0	7.0	13.0	351.9	10.3
24	4.0	58.0	10.0	8.0	1.0	668.5	31.7
25	4.0	58.0	10.0	8.0	13.0	458.4	15.6

SAMPLE RUN #1

In this run symbols, values, and a grid are called and plotted.

```
a
UNIV OF ALBERTA COMPUTING SERVICES
MTS (0042-FE93/22/0F)
#sig
#PASSWORD?
?[redacted]
#ON AT  12:47.52 ON TUE MAR 04/75  LAST ON AT  12:43.59
#r dlsplot *plotlib 9=-plot 7=atobj
#12:49.02

 DLSPLOT ROUTINE


 FIRST TOWNSHIP I3
 058

 LAST TOWNSHIP I3
 063

 FIRST RANGE I2
 01

 LAST RANGE I2
 05

 SCALE 1:    F8.0
 250000.

 VERTICAL ADJUSTMENT 1.0=NONE F8.6
 1.0041

 HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
 1.0123

 RUN GRID? YES=1,NO=0
 1


 YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
 FIRST TOWNSHIP-  58
 LAST TOWNSHIP-  63
 FIRST RANGE-   1
 LAST RANGE-   5
 SCALE, 1/ 250000.
 YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES


 CONTINUE? YES=22,TRY AGAIN=24,NO=25
 22




 DO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0
 1

 WHICH SYMBOL? FORMAT-I3 REF -CALCOMP MANUAL
```

```
.07

DO YOU WISH TO PLOT A NUMBER-YES=1,NO=0
1

SIZE OF NUMBER IN INCHES? F4.3
.07

ROUNDING? NUMBER OF DIGITS RIGHT OF DECIMAL? I2
-1

ANGLE (DEGREES) OF NUMBER PLOT I3
020

INPUT FILE FORMAT? MER,TP,RG,S,LSD,VALUE
ENTIRE INPUT MUST BE REAL(F)NUMBERS
(7x,5f8.1,8x,f8.1)

DO YOU WISH TO TRY AGAIN?-YES=1,NO=0
0
#12:53.35 4.751 RC=0
#



 r *calcompq scards=-plot
#12:54.16
 *** CALCOMP RECEIPT #  1046****  PLOT LENGTH =  12 INCHES;
#12:54.31 .924 RC=0
#
```


#23
#115
#117
#21
#23
#7
#85
#51
#3
#14
#47
#558
#195
#2
#41
#3
#6
#36
#5
#13
#29
#15
#51
#5
#2
#1
#15
#1063
#183
#23
#42
#12
#13

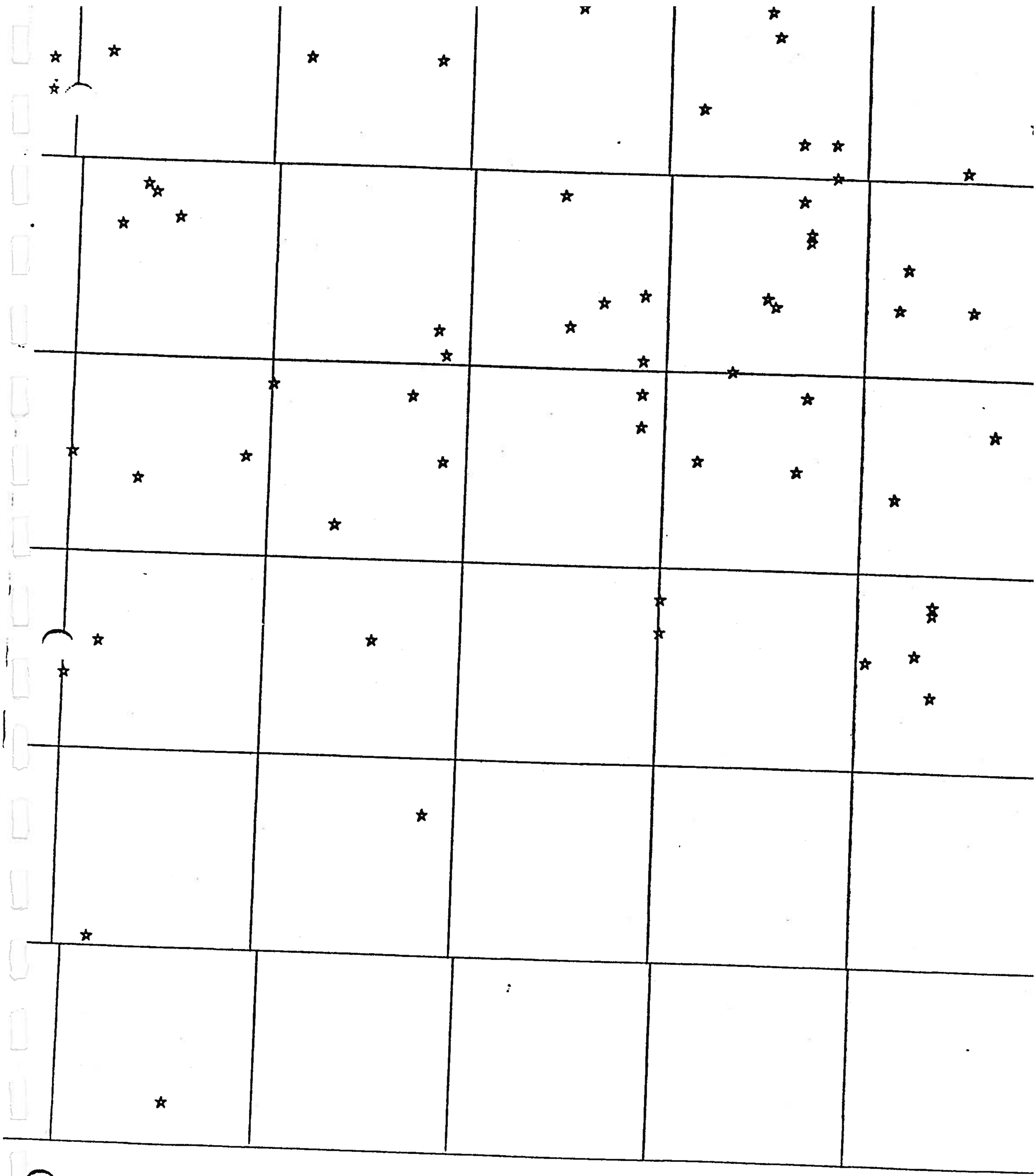
SAMPLE RUN #1

SAMPLE RUN #2

In: this run symbols and a grid are called and plotted. Note that the symbol has been increased in size relative to run #1.

```
r dlsplot+*plotlib 9=-plot2 7=atobj
#12:56.05

 DLSPLOT ROUTINE


 FIRST TOWNSHIP I3
 058

 LAST TOWNSHIP I3
 063

 FIRST RANGE I2
 01

 LAST RANGE I2
 05

 SCALE 1:    F8.0
 250000.

 VERTICAL ADJUSTMENT 1.0=NONE F8.6
 1.0041

 HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
 1.0123

 RUN GRID? YES=1,NO=0
 1


 YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
 FIRST TOWNSHIP-  58
 LAST TOWNSHIP-  63
 FIRST RANGE-    1
 LAST RANGE-    5
 SCALE, 1/ 250000.
 YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES


 CONTINUE? YES=22,TRY AGAIN=24,NO=25
 22




 DO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0
 1

 WHICH SYMBOL? FORMAT-I3,REF.-CALCOMP MANUAL
 014

 SIZE OF SYMBOL IN INCHES? F4.3
 .1

 DO YOU WISH TO PLOT A NUMBER-YES=1,NO=0
 0

 [illegible]
```

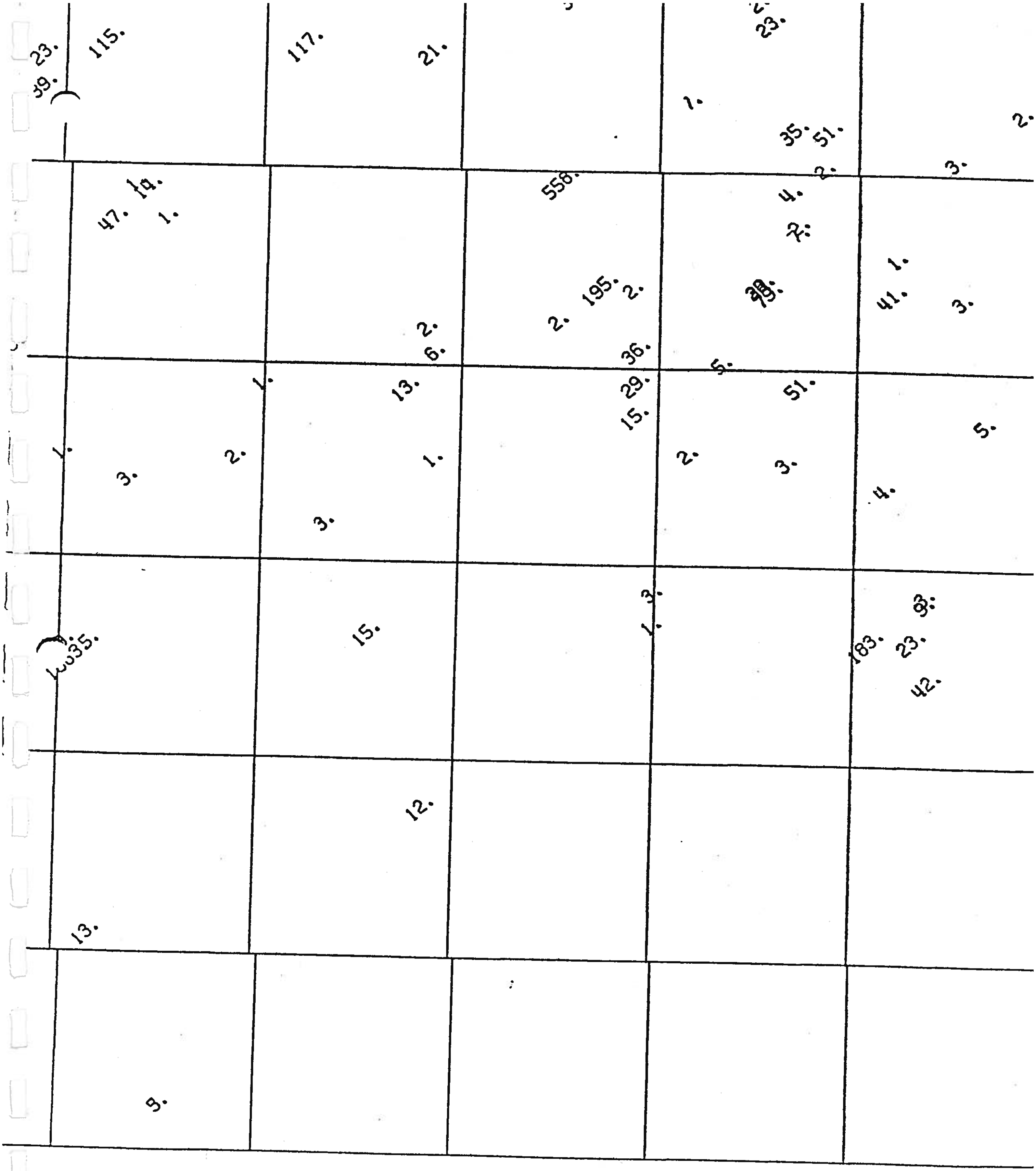
```
 DO YOU WISH TO TRY AGAIN?-YES=1,NO=0
 0
#13:00.12 3.476 RC=0
#

 r *calcompq scards=-plot2
#13:00.46
 *** CALCOMP RECEIPT #  1047****   PLOT LENGTH  =  12 INCHES;
#13:01.14 .715 RC=0
#
```



SAMPLE RUN #2

SAMPLE RUN #3

In this run numbers, values, and a grid were called and plotted. Note the numbers have been increased in size and plotted at a greater angle relative to run #1. Also note that the rounding factor has been changed from -1 as it is in run #1 to 00 resulting in the decimal point being plotted.


23.
39.
115.
117.
21.
23.
1.
35.
51.
2.
558.
2.
3.
14.
47.
1.
4.
1.
195.
2.
79.
41.
3.
2.
2.
6.
36.
5.
1.
13.
29.
51.
15.
5.
1.
3.
2.
1.
2.
3.
4.
3.
3.
1.
35.
15.
183.
23.
42.
12.
13.
5.

SAMPLE RUN #3

```
r dlsplot+*plotlib 7=atobj 9=-plot3
#13:02.58

DLSPLOT ROUTINE


FIRST TOWNSHIP I3
058

LAST TOWNSHIP I3
063

FIRST RANGE I2
01

LAST RANGE I2
05

SCALE 1:     F8.0
250000.

VERTICAL ADJUSTMENT 1.0=NONE F8.6
1.0041

HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
1.0123

RUN GRID? YES=1,NO=0
1


YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
FIRST TOWNSHIP-  58
LAST TOWNSHIP-  63
FIRST RANGE-    1
LAST RANGE-    5
SCALE, 1/ 250000.
YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES


CONTINUE? YES=22,TRY AGAIN=24,NO=25
22




DO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0
0

DO YOU WISH TO PLOT A NUMBER-YES=1,NO=0
1

SIZE OF NUMBER IN INCHES? F4.3
.1

ROUNDING? NUMBER OF DIGITS RIGHT OF DECIMAL? I2
00
```

```
INPUT FILE FORMAT? MER,TP,RG,S,LSD,VALUE
ENTIRE INPUT MUST BE REAL(F)NUMBERS
(7x,5f3.1,8x,f8.1)

DO YOU WISH TO TRY AGAIN?-YES=1,NO=0
0
#13:08.40 4.623  .=0
#


 r *calcompq scards=-plot3
#13:09.25
 *** CALCOMP RECEIPT #  1048****  PLOT LENGTH =  12 INCHES;
#13:09.33 .899 RC=0
#
```

SAMPLE RUN #4

In this run symbols and numbers were plotted. Note the increased angle of the number plot and the change of symbol relative to run #1.

```
#r dlsplot+*plotlib 7=atobj 0=-plot4
#13:11.36

 DLSPLOT ROUTINE


 FIRST TOWNSHIP I3
 058

 LAST TOWNSHIP I3
 063

 FIRST RANGE I2
 01

 LAST RANGE I2
 05

 SCALE 1:    F8.0
 250000.

 VERTICAL ADJUSTMENT 1.0=NONE F8.6
 1.0041

 HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
 1.0123

 RUN GRID? YES=1,NO=0
 0


 YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
 FIRST TOWNSHIP-  58
 LAST TOWNSHIP-  63
 FIRST RANGE-    1
 LAST RANGE-    5
 SCALE, 1/ 250000.
 YOUR PLOT WILL NOT BE LONGER THAN  0.INCHES


 CONTINUE? YES=22,TRY AGAIN=24,NO=25
 22




 DO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0
 1

 WHICH SYMBOL? FORMAT-I3,REF.-CALCOMP MANUAL
 075

 SIZE OF SYMBOL IN INCHES? F4.3
 .2
```

```
1
SIZE OF NUMBER IN INCHES? F4.3
.05
ROUNDING? NUMBER OF DIGITS RIGHT OF DECIMAL? I2
-1
ANGLE (DEGREES) OF NUMBER PLOT I3
070
INPUT FILE FORMAT? MER,TP,RG,S,LSD,VALUE
ENTIRE INPUT MUST BE REAL(F)NUMBERS
(7x,5f8.1,8x,f8.1)
DO YOU WISH TO TRY AGAIN?-YES=1,NO=0
0
#13:17.52 4.031 RC=0
#

 r *calcompq scards=-plot4
#13:18.32
 *** CALCOMP RECEIPT #  1049****   PLOT LENGTH =  12 INCHES;
#13:18.39 .758 RC=0
#
```

SAMPLE RUN #4

SAMPLE RUN #5

In this run only the grid was called for.

```
 r dlsplot+*plotlib 9=-plot5 7=atobj
#13:20.53

 DLSPLOT ROUTINE


 FIRST TOWNSHIP I3
 058

 LAST TOWNSHIP I3
 063

 FIRST RANGE I2
 01

 LAST RANGE I2
 05

 SCALE 1:    F8.0
 250000.

 VERTICAL ADJUSTMENT 1.0=NONE F8.6
 1.0041

 HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
 1.0123

 RUN GRID? YES=1,NO=0
 1


 YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
 FIRST TOWNSHIP-  58
 LAST TOWNSHIP-  63
 FIRST RANGE-    1
 LAST RANGE-    5
 SCALE, 1/ 250000.
 YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES


 CONTINUE? YES=22,TRY AGAIN=24,NO=25
 22




 DO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0
 0

 DO YOU WISH TO PLOT A NUMBER-YES=1,NO=0
 0

 DO YOU WISH TO TRY AGAIN?-YES=1,NO=0
 0
#13:23.40 1.404 RC=0
#
 r *calcompq scards=-plot5
#13:25.16
```

SAMPLE RUN #5

SAMPLE RUN #6

This run was not plotted. It is an illustration of obvious errors in input, the programmed reaction to such errors, the method of correcting input, and the terminating of program execution.

```
DLSPLOT ROUTINE

FIRST TOWNSHIP I3
015

LAST TOWNSHIP I3
001

FIRST RANGE I2
01

LAST RANGE I2
05

SCALE 1:    F8.0
250000.

VERTICAL ADJUSTMENT 1.0=NONE F8.6
1.

HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
1.

RUN GRID? YES=1,NO=0
1

YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
FIRST TOWNSHIP-  15
LAST TOWNSHIP-    1
FIRST RANGE-    1
LAST RANGE-    5
SCALE, 1/ 250000.
YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES

CONTINUE? YES=22,TRY AGAIN=24,NO=25
22
SOMTHING WRONG WITH LOCATIONS

YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
FIRST TOWNSHIP-  15
LAST TOWNSHIP-    1
FIRST RANGE-    1
LAST RANGE-    5
SCALE, 1/ 250000.
YOUR PLOT WILL NOT BE LONGER THAN  9.INCHES

CONTINUE? YES=22,TRY AGAIN=24,NO=25
24

DLSPLOT ROUTINE

FIRST TOWNSHIP I3
001

LAST TOWNSHIP I3
```

```
01

LAST RANGE I2
01

SCALE 1:    F8.0
250000.

VERTICAL ADJUSTMENT 1.0=NONE F8.6
1.

HORIZONTAL ADJUSTMENT 1.0=NONE F8.6
1.

RUN GRID? YES=1,NO=0
1


YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
FIRST TOWNSHIP-    1
LAST TOWNSHIP-  15
FIRST RANGE-    1
LAST RANGE-    1
SCALE, 1/ 250000.
YOUR PLOT WILL NOT BE LONGER THAN  2.INCHES


CONTINUE? YES=22,TRY AGAIN=24,NO=25
22
SOMTHING WRONG WITH LOCATIONS


YOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-
FIRST TOWNSHIP-    1
LAST TOWNSHIP-  15
FIRST RANGE-    1
LAST RANGE-    1
SCALE, 1/ 250000.
YOUR PLOT WILL NOT BE LONGER THAN  2.INCHES


CONTINUE? YES=22,TRY AGAIN=24,NO=25
25
#13:41.02 .866 RC=0
#
```

Program listing

```
C          MAIN PROGRAM DLSPLOT OBJECT FILE IS CALLED L1
        DIMENSION FTRM(20)
        COMMON TP1ST,TPLST,RG1ST,RGLST,CIERT,DEGRAD,PLMED,RDMED,
     *  RDTP1, NCVR,SCALE,YHIPT,XFACT,YFACT,ADJTP,ADJRG
        CALL PLOTS
   24   CONTINUE
    2   FORMAT (//,17HFIRST TOWNSHIP I3)
    3   FORMAT (/,16HLAST TOWNSHIP I3)
    4   FORMAT (/,14HFIRST RANGE I2)
    5   FORMAT (/,13HLAST RANGE I2)
    6   FORMAT (/,16HSCALE 1:     F8.0)
    7   FORMAT (I3)
    8   FORMAT (I2)
    9   FORMAT (F8.0)
        RD49=3090.0
        WRITE (6,1)
        WRITE (6,2)
        READ  (5,7)ITP1
        WRITE (6,3)
        READ  (5,7)ITPL
        WRITE (6,4)
        READ  (5,8)IRG1
        WRITE (6,5)
        READ  (5,8)IRGL
        WRITE (6,6)
        READ  (5,9)SCAL
   10   FORMAT(/,33HVERTICAL ADJUSTMENT 1.0=NONE F8.6)
   11   FORMAT(/,35HHORIZONTAL ADJUSTMENT 1.0=NONE F8.6)
   12   FORMAT(F8.6)
        WRITE(6,10)
        READ(5,12) ADJTP
        WRITE(6,11)
        READ(5,12) ADJRG
        TP1ST=ITP1
        TPLST=ITPL
        RG1ST=IRG1
        RGLST=IRGL
   17   FORMAT(/,20HRUN GRID? YES=1,NO=0)
   18   FORMAT(I1)
        WRITE(6,17)
        READ(5,18)IGRD
        PI=3.14159
        RE=3958.94
        CIERT=2.0*PI*RE
        DEGRAD=PI/180.0
        SCALE=SCAL/63360.0
        VSPRD=TPLST-TP1ST
        HSPRD=RGLST-RG1ST
        TPMED=TP1ST+VSPRD/2.0
        PLMED=(TPMED*11.0/126.0+43.0)*DEGRAD
        RDMED=RD49-TPMED*6.0
        RDTP1=RDMED+VSPRD*3.0
        RDTPL=RDMED-VSPRD*3.0
        ANGVR=(6.0*(RG1ST+HSPRD/2.0))/RDMED+3.0*PI/2.0
        ITWP=TPLST
        YCOOR=VSPRD*6.0
        CALL XWARP (ITWP,WARPX)
        XCOOR=(RG1ST)*6.0*WARPX
        YCONF=RDTPL
```

```
      XCONF=XCCNE/YCCNT
      XCCNE=ANGVP-XCONF
      YHIPT=YCCNE*SIN (XCCNT)
      ITWP=TP1ST
      CALL XWARP(ITWP,WARPX)
      XFACT=HSPRD*1.25*7.C
      YFACT=VSPRD*6.C*1.25
      PLTL=(XFACT*2.C+6.C)/SCALE
   26 CONTINUE
      CALL CHCK1    (ITP1,ITPL,IRG1,IRGL,SCAL,PLTL)
   20 FORMAT(//,75HCONTINUE? YES=22,TRY AGAIN=24,NO=25)
   21 FORMAT(I2)
   28 WRITE(6,20)
      READ(5,21)ICH
      IF(ICH.EQ.22)GO TO 22
      IF(ICH.EQ.24)GO TO 24
      IF(ICH.EQ.25)GO TO 25
      GO TO 28
   22 CONTINUE
      CALL CHCK2    (ITP1,ITPL,IRG1,IRGL,IST)
      IF(IST.EQ.1)GO TO 26
   23 CONTINUE
      CALL XLIMIT(PLTL)
      CALL TOCOR
      IF(IGRD.EQ.C)GO TO 10
      CALL GRID
   10 CONTINUE
      CALL PLOT(PLTL,C.C,999)
   25 STOP
      END
      SUBROUTINE GRID
C       SUBROUTINE GRID OBJECTIVE FILE IS CALLED L2
      COMMON TP1ST,TPLST,RG1ST,RGLST,CISFT,DSRAD,PLMED,PCMED,
     1RDTP1,ANGVF,SCALE,YHIPT,XFACT,YFACT,ADJTP,ADJRG
      RNGN=RG1ST
    1 CONTINUE
      ITWP=TP1ST
      TWPN=TP1ST
      CALL XWARP (ITWP,WARPX)
      YCOOR=1.C
      XCOOR=(RNGN-1.)*6.0*WARPX
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,3)
    2 CONTINUE
      YCOOR=(TWPN-TP1ST)*6.C
      XCOOR=(RNGN-1.)*6.C*WARPX
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,2)
      YCOOR=YCOOR+6.C
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,2)
      TWPN=TWPN+1.C
      ITWP=TWPN
      WARP2=WARPX
      CALL XWARP (ITWP,WARPX)
      IF (WARP2.NE.WARPX) GO TO 4
    3 CONTINUE
      IF (TWPN.LT.(TPLST+1.))GO TO 2
      IF (RNGN.GT.RGLST)GO TO 5
```

```
      GO TO 1
    4 XCOOR=(RNGN-1.)*6.0*WARPX
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,2)
      GO TO 7
    5 CONTINUE
      TWPN=TP1ST
    6 CONTINUE
      ITWP=TWPN
      CALL XWARP (ITWP,WARPX)
      YCOOR=(TWPN-TP1ST)*6.0
      XCOOR=(RG1ST-1.)*6.0*WARPX
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,3)
      AWIDE=(RGLST+1)*6.0*WARPX*ADJRG
    7 ASTEP=SCALE*.1
      XCOOR=XCOOR+ASTEP
      CALL ACONE (XCOOR,YCOOR,XCORD,YCORD)
      CALL PLOT (XCORD,YCORD,2)
      IF (XCOOR.LE.AWIDE) GO TO 7
      IF (TWPN.GT.TPLST) GO TO 8
      TWPN=TWPN+1.0
      GO TO 6
    8 CONTINUE
      RETURN
      END
      SUBROUTINE XWARP (ITWP,WARPX)
C       SUBROUTINES XWARP,ACONE OBJECT FILE IS CALLED L3
      COMMON TP1ST,TPLST,RG1ST,RGLST,CIRRT,DEGRAD,PLMED,RDMED,
     1RDTP1,ANGVR,SCALE,YHIPT,XFACT,YFACT,ADJTP,ADJRG
      DOUBLE PRECISION PLMED,YANG
      BADG=23.0/63.0
      IADG=0
    2 CONTINUE
      IF(ITWP-(2+4*IADG))1,1,3
    3 IADG=IADG+1
      GO TO 2
    1 CONTINUE
      YADG=IADG
      YANG=(49.0+(BADG*YADG))*DEGRAD
      WARPX=DCOS (PLMED)/DCOS (YANG)
      RETURN
      END
      SUBROUTINE ACONE (XCOOR,YCOOR,XCORD,YCORD)
      COMMON TP1ST,TPLST,RG1ST,RGLST,CIRRT,DEGRAD,PLMED,RDMED,
     1RDTP1,ANGVR,SCALE,YHIPT,XFACT,YFACT,ADJTP,ADJRG
      DOUBLE PRECISION XCONE,YCONE
      YCONE=RDTP1-YCOOR*ADJTP
      XCONE=ADJRG*XCOOR*YCONE/RDMED
      XCONE=XCONE/YCONE
      XCONE=(ANGVR-XCONE)
      XCORD=((YCONE*DCOS (XCONE))+XFACT)/SCALE
      YCORD=((YCONE*DSIN (XCONE)-YHIPT) +YFACT)/SCALE
      RETURN
      END
      SUBROUTINE TCOOR
C       SUBROUTINE TCOOR OBJECT FILE IS CALLED L4
      DIMENSION FORM(20)
      COMMON TP1ST,TPLST,RG1ST,RGLST,CIRRT,DEGRAD,PLSTS,PLSTH,RDSTN,
```

```
      REAL  SMPR,TWPN,ENGN,SECN,LSON
      CALL STUFF(ISYM,JSYM,RIG,SIZE,NN,THET,INUM,JSUT,FORM)
      IF(JSUT.EQ.0) GO TO 100
    1 CONTINUE
      READ (7,FORM,END=100) SMPR,TWPN,ENGN,SECN,LSON,RPN
      YC=6.0*(TWPN-TP1ST)
      XC=6.0*(ENGN-1)
      IS=SECN
      K=0
      DO 3 J=1,6
      K=K+6
      IF(IS-K)4,4,3
    3 CONTINUE
    4 YJ=J-1
      YC=YC+YJ
      LINE=(J/2)*2
      JLINE=J-LINE
      IK=IS+6-K
      IF(JLINE)5,5,6
    5 XJ=6-IK
      GO TO 7
    6 XJ=IK-1
    7 XC=XC+XJ
      IL=LSON
      IF(IL.GE.0) GO TO 8
      XC=XC+0.5
      YC=YC+0.5
      GO TO 14
    8 K=0
      DO 9 J=1,4
      K=K+4
      IF(IL-K)10,10,9
    9 CONTINUE
   10 YJ=J-1
      YC=YC+YJ/4.0+0.125
      LINE=(J/2)*2
      JLINE=J-LINE
      IK=IL+4-K
      IF(JLINE)11,11,12
   11 XJ=4-IK
      GO TO 13
   12 XJ=IK-1
   13 XC=XC+XJ/4.0+0.125
   14 XCOOR=XC
      YCOOR=YC
      ITWP=TWPN
      CALL XWARP(ITWP,WARPX)
      XCOOR=XCOOR*WARPX
      CALL ACONF (XCOOR,YCOOR,XCORD,YCORD)
      IF(ISYM.EQ.0) GO TO 20
      CALL SYMBOL(XCORD,YCORD,RIG,JSYM,0.0,-1)
      XCORD=XCORD+RIG
   20 CONTINUE
      IF(INUM.EQ.0) GO TO 1
      CALL NUMBER(XCORD,YCORD,SIZE,RPN,THET,NN)
      GO TO 1
  100 RETURN
      END
      SUBROUTINE CHCK1    (ITR1,ITRL,IRG1,IRGL,SCAL,PLTL)
```

```
      DIMENSION FORM(20)
   19 FORMAT(//,44HYOUR VARIABLES HAVE BEEN READ IN AS FOLLOWS-)
   20 FORMAT(16HFIRST TOWNSHIP- ,I3)
   21 FORMAT(15HLAST TOWNSHIP- ,I3)
   22 FORMAT(13HFIRST RANGE- ,I3)
   23 FORMAT(12HLAST RANGE- I3)
   24 FORMAT(10H SCALE, 1/,F8.0)
   26 FORMAT(33HYOUR PLOT WILL NOT BE LONGER THAN,F4.0,6HINCHES)
      WRITE(6,19)
      WRITE(6,20)ITP1
      WRITE(6,21)ITPL
      WRITE(6,22)IRG1
      WRITE(6,23)IRGL
      WRITE(6,24)SCAL
      WRITE(6,26)PLTL
      RETURN
      END
      SUBROUTINE CHCK2    (ITP1,ITPL,IRG1,IRGL,IST)
      IF(ITP1.GE.ITPL)GO TO 1
      IF(IRG1.GE.IRGL)GO TO 1
      IF(ITPL.GT.130)GO TO 1
      IF(IRGL.GT.30)GO TO 1
      GO TO 3
    1 IST=1
    2 FORMAT(29HSOMTHING WRONG WITH LOCATIONS)
      WRITE(6,2)
    3 RETURN
      END
      SUBROUTINE STUFF (ISYM,JSYM,BIG,SIZE,NN,THET,INUM,JSUT,FORM)
      DIMENSION FORM(20)
    1 FORMAT(//,39HDO YOU WISH TO PLOT A SYMBOL-YES=1,NO=0)
    2 FORMAT(I2)
    3  FORMAT(/,43HWHICH SYMBOL? FORMAT-I3,SEE.-CALCOMP MANUAL)
    4 FORMAT(I3)
    5 FORMAT(/,30HSIZE OF SYMBOL IN INCHES? F4.3)
    6 FORMAT(F4.3)
    7 FORMAT(/,39HDO YOU WISH TO PLOT A NUMBER-YES=1,NO=0)
    8 FORMAT(/,40HINPUT FILE FORMAT? MER,TP,RG,S,LSD,VALUE)
    9 FORMAT(/,35HINPUT FILE FORMAT? MER,TP,RG,S,LSD )
   10 FORMAT(35HENTIRE INPUT MUST BE REAL(F)NUMBERS)
   11 FORMAT(20A4)
   13 FORMAT(/,30HSIZE OF NUMBER IN INCHES? F4.3)
   14 FORMAT(/,47HROUNDING? NUMBER OF DIGITS RIGHT OF DECIMAL? I2)
   15 FORMAT(I2)
   16 FORMAT(/,33HANGLE (DEGREES) OF NUMBER PLOT I3)
   17 FORMAT(/,36HDO YOU WISH TO TRY AGAIN?-YES=1,NO=0)
   21 CONTINUE
      WRITE(6,1)
      READ(5,40)ISYM
      IF(ISYM.EQ.0) GO TO 22
      WRITE(6,3)
      READ(5,4)JSYM
      WRITE(6,5)
      READ(5,6)BIG
   22 CONTINUE
      WRITE(6,7)
      READ(5,40)INUM
      IF(INUM.EQ.0) GO TO 25
      WRITE(6,13)
```

```
      READ(5,6)SIZE
      WRITE(6,14)
      READ(5,15)NN
      WRITE(6,16)
      READ(5,4)ITHE
      THET=ITHE
21    CONTINUE
      JSUT=ISYM+INUM
      IF(JSUT.EQ.2) GO TO 23
      IF(JSUT.EQ.0) GO TO 19
      IF(ISYM.EQ.1) GO TO 24
23    CONTINUE
      WRITE(6,8)
      GO TO 25
24    WRITE(6,9)
25    WRITE(6,10)
      READ(5,11)(FORM(I),I=1,20)
49    CONTINUE
      WRITE(6,17)
      READ(5,40)MBCB
40    FORMAT(I1)
      IF(MBCB.EQ.1) GO TO 21
300   RETURN
      END
```