TRAVPLUS II

G & L SURVEYING & ENGINEERING

Rt.2, Box 246

Sheridan, OR
97378

1983

T R A V P L U S I I

by

William A. Gille, P.E., P.L.S.

and

Dan E. Linscheid, P.L.S.

1983
G & L Surveying and Engineering
Route 2 Box 246
Sheridan, Oregon 97378

G & L Surveying and Engineering makes no express or implied warranty with regard to either the program material contained herein or its fitness to any particular purpose. This program material is sold solely on an "as is" basis, and the entire risk as to its quality and performance is with the user should the program material prove defective. The purchaser and/or user shall bear the entire cost of all necessary correction and all incidental or consequential damages arising from the use of this program material. G & L Surveying and Engineering shall not be liable for any incidental or consequential damage in connection with or arising out of the furnishing, use or performance of this program material.

TABLE OF CONTENTS

Chapter 1

PROGRAM FAMILIARIZATION

The program herein is the end result of several years of effort by the authors. We have attempted to design a program having appeal to a wide spectrum of users, while lending itself to a certain flexibility for various endeavors. Both newcomers and the "old pro's" to the survey calculations and programming fields are encouraged to read this chapter to gain a better understanding of the system structuring of our program.

1.1 INTRODUCTION

This program was designed to give the user a quick, easy and accurate system of performing survey calculations. The basic program structure was designed for the HP29C about 4 years ago. With the arrival of the HP41C, a new dimension was added to survey calculations, in that the user could take a hand held "computer" into the field, allowing one unskilled in programming to perform various survey calculations with speed and dependably accurate results.

Every effort has been made to deliver software having programming procedures which can be easily understood. The program is produced only in a "non-private" format, allowing the user flexibility in amending and deleting portions, or incorporating entirely extrinsic programs or routines. It will be updated as the authors become aware of better procedures, hardware, etc.. Should you have questions or suggestions regarding the structure or use of this program, please contact the authors at the address given in the inside cover of the booklet.

All coordinate pairs are stored in the extended memory modules. Total coordinate pair availability depends upon the amount of extended memory:

- X-Function/X-Memory = 60 pairs

- plus 1 X-Memory = 180 pairs

- plus 2 X-Memory = 300 pairs

Attention is also directed to Appendices A and B wherein the user determined flag usage is explained and all routines are flowcharted.

1.2 DEFINITION OF TERMS

Terms used herein have been for the most part, universally adopted by the surveying profession. Those listed below are given only to establish a reference for the particular meaning used herein:

Angle — The difference in degrees, minutes and seconds, between 2 convergent horizontal lines.

Azimuth — The direction of one point or object, with respect to another, where the direction of the line is expressed as the clockwise angle from 0 to 360, from the reference meridian of North.

Back Azimuth — The azimuth of a line at the end opposite the reference end, or 180 from the forward or ahead azimuth. Used herein, this can be either an arbitrary or calculated value.

Closing Point — The position of a traverse station obtained by computation through a closed traverse which fails to fall at the initial position.

Closing Latitude and Longitude — Those values calculated as representing the magnitude of change required to adjust, using the Compass or Bowditch rule, the closing point to the values of the initial point or position.

Coordinates or coordinate pairs

The values of latitude (Northing) and longitude (Easting) calculated and used in accordance with the cartesian coordinate system. This program calculates, defines, recalls and prints these values by assigning an arbitrary point number to each pair.

Deflection Angle A horizontal angle measured from the prolongation of the preceding line, right or left, to the following (ahead) line.

Distance The amount of separation, in user defined units, between two points, lines or objects measured upon a horizontal two dimensional plane.

Field Angle The traverse routine used herein which allows the user to input measured horizontal angles between the backsights and foresights or points occupied in a closed or open traverse. The alternate to this routine allows use of predetermined or record azimuths or bearings instead of measured values.

Inverse Computations of the length and azimuth of a line based upon the coordinate values of its endpoints.

<u>Intersections</u>

Azimuth - Azimuth A method of determining the coordinate values of the point of intersection of two lines, the azimuths of each being known, along with the coordinates of a point on each line. See Section 3.8 for an example.

Azimuth - Distance Intersection

A method of determining the coordinate values of the point of intersection of a line of known azimuth passing through known coordinates with another line of known length originating from a point with known coordinates. Two solutions are possible in this type of intersection. See Section 3.8 for an example.

Distance - Distance Intersections

A method of determining the coordinate values of the point of intersection of two lines of known length radiating from two separate points of known coordinates. Two solutions are possible in this type of intersection. See Section 3.8(d) for an example.

Sideshots

A measurement from a point to locate a point or object not intended to be used as a base for the extension of the survey. Vertical Angle

In this program, zenith is reference 0, while horizontal is 90. If your application uses the opposite reference system, you can change the SIN statement at line to COS to accommodate your system.

1.3 FORMAT OF USER INSTRUCTIONS

Instructions herein are composed of 5 columns, using the same format given in most Hewlett Packard program use instructions. On the far left, the step column informs the user of the current step number. Next, the instructions column relates any required instructions or comments regarding operations being performed to the user. The input column directs you to input data, units of data or alpha responses to specific questions. Appropriate inputs may include the digits 0 thru 9 and alpha keys "Y" for yes and "N" for no.

The function column instructs you to press certain keys following execution of a routine or input of previous data. The subscript a to the indicated function instruction directs you to press the alpha key before and after inputting the alpha data. For example, XEQ SIZEa 020 indicates the user should press XEQ, alpha, 020 to change the data storage register/program memory size allocation to the indicated value.

The display column shows prompts, intermediate answers and final answers to calculations.

1.4 LOADING PROGRAM INTO CALCULATOR

This system is comprised of a total of 11 magnetic cards, which are loaded into the calculator in the same manner as any other program. If you wish to retain any of the programs currently in program memory, press GTO.. prior to loading the mag cards. Upon receipt of your software you may wish to separate the magnetic cards into five groups. The cards consist of program sets, broken down as follows:

CARD MARKED	FUNCTIONS	TOTAL CARD
TR	Traverse, inverse, fld. data traverse sideshots, define & print points	5
CO	Compass adjust and rotation	2
IS	Intersections	2
~~CL~~ CH	Card load	1
ML	Memory load	1

Note:

Prior to executing any of these programs, a working file must be created within the extended function module, to allow coordinate storage. Here's the process:

Assuming an empty directory (see EMDIR, page 23, Extended Functions Owners Manual) exists to allow maximum coordinate pair storage, execute the following:

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	execute EMDIR (extended memory directory)		EMDIRa	DIR EMPTY
2	input SP (file name) create file	SP	CRFLDa	nnn*

*this number is 2 times the total number of coordinate pairs available in your particular system.

Should you desire, you can specify a smaller value than the one displayed, in order to store programs or data using other file names. Refer to pages of the above noted manual for help.

You must have the TR program loaded to use either the CO or IS programs. Clearly the TR program handles the bulk of the calculations. You may wish to clear the Compass Adjust and Rotation programs following their execution. Here's a few things you may wish to keep in mind if you intend to use or design other programs for use with this system:

1. Flag settings in TR (usually print in the print/define point routines) may interfere with your program...check Appendix A for clarification on this.

2. Executing A, the Begin Traverse Key, will cause a preprogrammed size statement(size 020) to be executed. This sets aside the first 19 registers for data only. You may want to change this to fit your particular needs. You'll know you've exceeded the allowable data vs. program memory partition when you receive a "Try Again" error message upon trying too large a size number.

3. Become familiar with the local, Alpha and Global labels listed in Appendix D. This will help alleviate inopportune routing to other program areas.

Chapter 2

COORDINATE POINT OPERATIONS

This chapter describes utility routines used to define and recall coordinate pairs in association with assigned point nombers. As part of the define point routine, values are stored in specific registers of extended memory. Upon execution of the print point routine, the same values are recalled into the stack. A method of printing out an entire block of coordinates is also described. Further, the routine enabling more permanent storage of coordinate pairs onto magnetic cards or data transfer from mag cards to the 41C are explained in this chapter.

2.1 POINT STORAGE AND RETRIEVAL ROUTINES

2.1.1 Storage of Coordinate Point Values

Using this routine, you can define (assign) coordinate values to any arbitrary point #, subject to the internal capacities of the calculator and the extended memory modules. In the following example, northing and easting values of 5000 and 2000, respectively, will be assigned to pt. #1:

STEP	INSTRUCTIONS	INPUT	FUNCTIONS	DISPLAY
1	Enter the Northing value	5000	Enter	5000
2	Enter the Easting value	2000	Enter	2000
3	Enter the Pt.#, define	1	[D]	Pt. 1

Note: This program tests the storage registers addressed by the assigned Pt.# to insure the user will not unintentionally "write" new coordinate values over those previously generated and stored. If values were resident in the memory register addressed, the Easting value of this "old" coordinate is displayed, and a "beep" is heard.

Thus, in the above example, had different values been already residing in the registers addressed by pt.#1, the Easting value of the previously recorded point would have been displayed and the "beep" tones would have been heard. To save these values already in these memory registers as well as the values we wish to store (now currently residing in the T and Z registers of the stack) perform the following steps:

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
4	switch to non-user mode		user	nnn
5	hit roll down key twice		R↓,R↓	2000
6	input your next choice of a point #	2	[D]	pt. 2.0000

If you should decide that you do not want to save the coordinates in these registers you can simply "write" over them after the "beep" and display of the Easting value by hitting the "run-stop" key. It is also possible to override this test (the beep if the register currently has data in it); refer to Appendix A, FLAG USAGE for further details.

2.1.2 Retrieval of Coordinate Point Values

To retrieve previously assigned northing and easting into the Y and X stack registers respectively, perform the following:

STEP	INSTRUCTION	INPUT	FUNCTION	DISPLAY
1	Input the pt.#, initialize	1	[E]	2000*

*This is the easting value of pt.#1 as previously defined; to view the northing value, press the user key, then press x≷y.

2.1.3 Printing a List of Coordinate Values

By setting flag 01, you can print out a "block" of coordinates, with associated point numbers if the printer is attached and turned on. This routine begins at the first point recalled, prints the set consisting of the point # and the northing and easting values, followed by subsequently higher point # sets until instructed to stop by the user:

STEP	INSTRUCTION	INPUT	FUNCTION	DISPLAY
1	Set enabling flag 01		SF 01	0.0000
2	Initiate listing	1	E	*
3	Terminate listing		R/S	

* Program execution symbol

2.2 LOADING DATA TO AND FROM MAGNETIC CARDS

With this program, coordinates will be loaded onto mag. cards or into x/memory without getting rid of resident programming. The cards will contain 16 coord. pairs each. With this option it is imperative that you:

1. Write the total number of coord. pairs stored on the first card in pencil or felt-tip pen,

2. NUMBER the cards sequentially as they exit the card reader when first storing them to cards.

In using this program, 32 registers will contain "blocks" of 16 coord. pairs each, and the calculator will now be unable to determine whether or not the cards were re-entered in correct order when putting the coords. back into X/Memory.

STEP	INSTRUCTION	INPUT	FUNCTION	DISPLAY
1.	Initialize program		XEQ CHa	To cards?
2.a	If you're putting coords ON cards:		R/S	First Pt#?
2.b	If you're storing coords in X/Mem:		[Shift,H]	First Pt#?
3.	Enter 1st coord. #	1	R/S	Last Pt#?
4.	Enter last coord. #	54	R/S	Cards Ldd.
or	Coords. Stored.*			

* Depending on which action your were taking.

This page was intentionally left blank.

This page was intentionally left blank.

Chapter 3

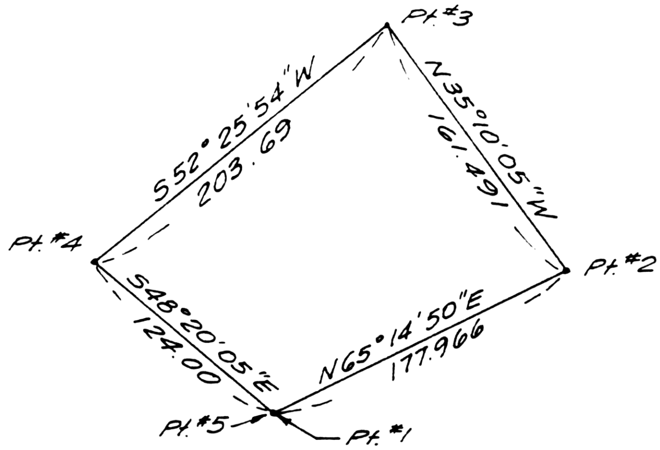
SURVEY CALCULATIONS

The routines herein have been designed to help you quickly and accurately perform various survey calculations. Before using any of the following routines, please become familiar with the format of user instructions given in Section 1.3 herein. You may wish to familiarize yourself with the routines by running through the examples given. Examples of the actual use of these various routines are given to help you. Make sure the working file used by this program has been created as outlined in Section 1.5.

3.1 TRAVERSE

Assuming the previously assigned coordinate values of N5000, E2000 for Pt. #1 are still in the calculator, (Section 2.1.1) we will now perform calculations for the traverse of the following courses:

	Pt.#	Data Bearing	Distance
Point of beginning	1		
		N65°14'50"E	177.966
	2		
		N35°10'05"W	161.491
	3		
		S52°25'54"W	203.690
	4		
		S48°20'05"E	124.00
	5		



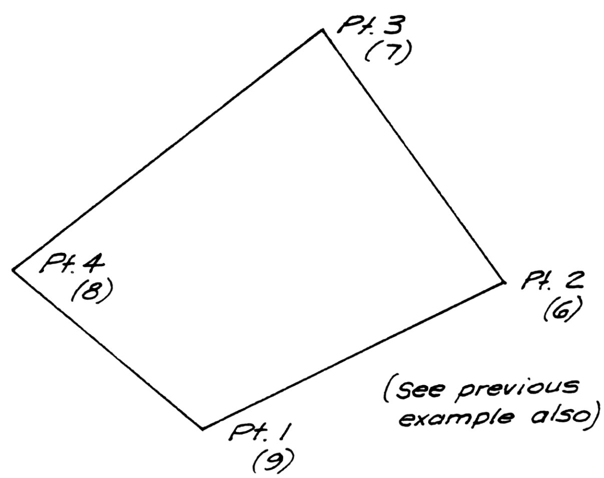
(Values of pt. #5 should theoretically equal the values of pt. #1)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Input beginning pt. #, initialize	1	[A]	INPT Az. D
2	Enter Azimuth	65.1450	Enter↑	65.1450
3	Enter Distance, R/S	177.966	R/S	Pt. 2.0000
4	Affirm Pt. # assignment		R/S	INPT.Az, D
5	Enter next Azimuth	324.4955	Enter↑	324.4955
6	Enter Distance, R/S	161.491	R/S	Pt. 3.0000
7	Affirm Pt. # assignment		R/S	INPT Az, D
8	Enter next Azimuth (Brg + 180)	232.2554	Enter↑	232.2554
9	Enter Distance, R/S	203.690	R/S	Pt. 4.0000
10	Affirm pt. # assignment		R/S	INPT Az, D
11	Enter next Azimuth	131.3955	Enter↑	131.3955
12	Enter Distance, R/S	124.00	R/S	Pt. 5.0000
13	Affirm pt. # assignment		R/S	INPT Az, D

Note: The azimuth entered at Step 5 above could have been entered as a negative 35°10' 05", which would cause the HP41C to calculate the Polar to Rectangular conversion in a counterclockwise manner; the results are the same. In like manner, the azimuth entered at Step 11 could have been entered as a bearing, added to 180 and finally changing the sign would have placed it in the correct quadrant. (-228.2005) As you can see, the format used will not change the calculated values; the easest method of calculation for you should be the determining factor.

3.2 AUTOMATIC COMPASS RULE ADJUSTMENT ROUTINE

This routine allows application of the Compass Rule Adjustment to an enclosed polygon, following angular error adjustment and initial calculations. The unadjusted coordinates are not disturbed while the adjusted coordinates are stored at a location you decide upon. The main polygon must be consecutively numbered beginning with point #1. Upon adjustment, the new points may be redefined using the old point numbers, to bring the calculations into conformity with field notes or previous calculations. Inversing between the adjusted coordinates will usually result in slightly different azimuths and distances than those initially calculated. The below example assigns adjusted values for old points 2,3,4 and 5 as point #6,7,8 and 9 respectively. (We assume the values generated in section 3.1 are still in the HP41C)



It is imperative that the summation of Horizontal Distances be stored in Register 15 prior to attempting this adjustment. In this example, this value is 667.147. This total Horizontal distance was calculated in example 3.1 above. Now assign CO to the C key (ASN, ALPHA, CO, ALPHA, C).

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Inverse between 1st pt. and closing pt.	1	[1]	INV. to pt.?
2	Input closing pt. #,R/S	5	R/S	Az=246.1844
3	Display distance		R/S*	DIST=0.2371
4	Store next available pt. # in J	6	[STO][J]	
5	Initialize routine		[C]	FRST pt.=?
6	Input 1st pt. being adjusted, R/S	2	R/S	New pt.=6.00
7	Affirm adjusted pt. 6		R/S	New pt.=7.00
8	Affirm adjusted pt. 7		R/S	New pt.=8.00
9	Affirm adjusted pt. 8		R/S	New pt.=9.00
10	Affirm adjusted pt. 9		R/S	

*Not required if printer is enabled

Note: See section F., Appendix E, pg. 44 if you wish to print out newly defined pts.

Following this procedure, a display of Pt. 9 shows the same values as Pt. 1. Had a printer been enabled, the coordinate values would have been printed out for both old and new points. You may wish to redefine pts. 2,3 and 4 to the newly adjusted values:

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Print (display) adjusted pt. 2*	6	[E]	2161.673
2	Define adjusted values as pt. 2*	2	[D]	

The above routine allows you to use the actual field note pt. numbers following the Compass Rule Adjustment. * repeat for points 3 and 4.

Note: In some situations, you may have more than one closed traverse within the same job. Because our program uses Pt.#1 as the initial Pt.# each time, you may wish to redefine Pt.#1 with the values of the "True" beginning Pt. For example, say you ran a second polygon, beginning at Pt.#4. (Pts. 1 thru 4 are your initial polygon). In this case you should redefine (temporarily) the coord. values of Pt.#4 to Pt.#1. Upon completion of the compass adjust routine, you can give Pt.#1 back its true values. In all other respects, this second "loop" should be given the same treatment, procedure-wise, as the first one.

3.3 AREA OF ENCLOSED POLYGON

The following routine displays the area, in acres, lying within an enclosed polygon, using the D.M.D. method of calculation. The area will be calculated in either the inverse routine or the traverse routine. The former might be used when checking areas based on calculated points, while the latter might aid in plat or deed area calculations.

STEP	INSTRUCTION	INPUT	FUNCTION	DISPLAY
1	Routine initialization		[J]	Area=nn.nn ac.

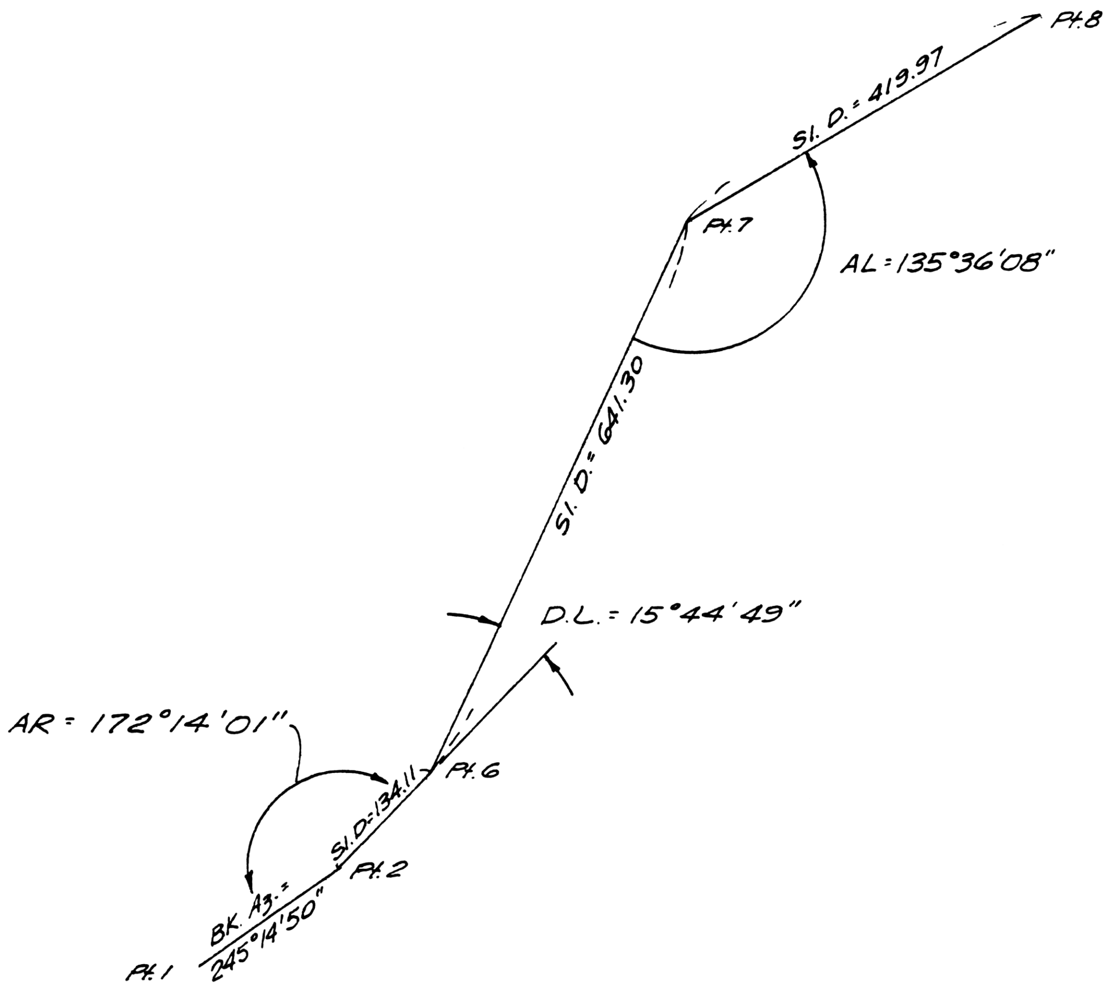
3.4 FIELD DATA TRAVERSE ROUTINE

This routine allows you to calculate an open or closed traverse based upon the measured horizontal angles (meaned) and unreduced (slope) distances. Should you need to use a record or assumed backsight point, you may have to calculate (traverse to or define) it before initializing this routine. If you measured any horizontal distances, simply press R/S at prompt for Vert. angle and put the measured distance in at the prompt for Sl. dist.

We will calculate points 6,7 and 8, based upon these measured values:

Pt. #		Horiz.∢	Vert.∢	Dist.	to Pt.#
2	AR	172°14'01"	89°59'41"	134.11	6
6	DL	15°44'49"	90°00'00"	641.30	7
7	AL	135°36'08"	91°01'58"	419.97	8

Where: AR = Angle right
DL = Deflection angle left
AL = Angle left



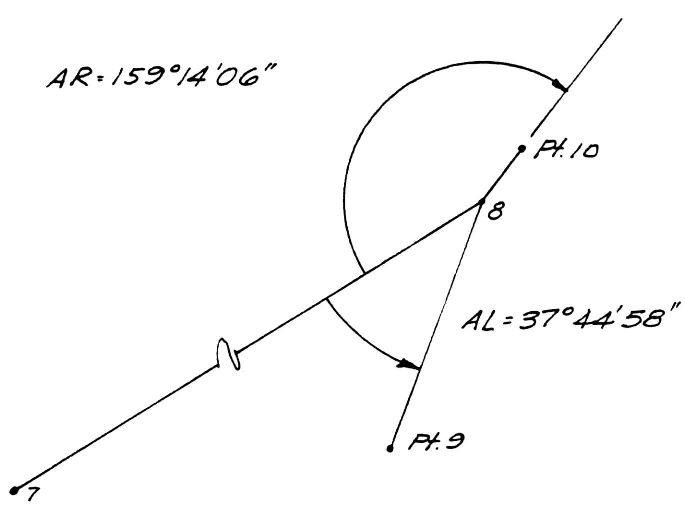
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize		[B]	INSTR Pt.=?
2	Input occupied pt. #	2	R/S	BKSTE Pt.=?
3	Input backsight pt. #	1	R/S	∢ turned
4	Input 1st horiz. angle	172.1401	R/S	Vert∢ = ?
5	Input 1st vert. angle	89.5941	R/S	SL. DIST = ?
6	Input 1st slope angle	134.11	R/S	Pt.# 3.0000?
7	Input correct pt. #	6	R/S	∢ Turned
8	Input 2nd horiz. angle	-195.4449	R/S	Vert.∢ = ?
9	Input Press R/S (90°)		R/S	SL. DIST = ?
10	Input 2nd slope dist.	641.30	R/S	Pt.# 7.0000?
11	Affirm suggested pt.#		R/S	∢ Turned
12	Input 3rd horiz. angle	-135.3608	R/S	Vert.∢ = ?
13	Input 3rd vert. angle	91.0158	R/S	SL. DIST = ?
14	Input 3rd slope dist.	419.97	R/S	Pt.# 8.0000?
15	Affirm suggested pt.#		R/S	∢ Turned*

* ignore this display

3.5 SIDESHOTS ROUTINE

This routine allows the user to calculate points radiated from the point occupied by the instrument, backsighting a known point. In the following example, pt. #8 will be occupied and while backsighting pt. 7, new pts. 9 and 10 will be generated based upon the following measurements:

Pt. #	Horiz. ∢	Vert. ∢	Sl. Dist.	Pt. #
8	37°44'58"AL	87°14'12"	57.81	9
8	159°14'06"AR	Horiz.	12.98	10



Step	Instructions	Input	Function	Display
1	Initialize		Shift b	SDSHOT FRM=?

2	Input occupied pt. #	8	R/S	BKSTE pt. = ?
3	Input backsight pt. #	7	R/S	Hrz.∢ turned=?
4	Input 1st horiz. angle	-37.4458	R/S	Vert.∢ = ?
5	Input 1st vert. angle	87.1412	R/S	Sl. Dist=?
6	Input 1st Sl. Dist.	57.81	R/S	Pt.# 9.0000?
7	Affirm suggested pt. #		R/S	Hrz.∢ turned=?
8	Input 2nd horiz. angle	159.1406	R/S	Vert.∢ = ?
9	Press R/S (90^{o})		R/S	SD. DIST = ?
10	Input (horiz.) DIST	12.98	R/S	Pt.# 10.0000?
11	Affirm suggested Pt. #		R/S	Hrz.∢turned=?*

*ignore this display

3.6 INVERSE ROUTINE

The following example demonstrates the calculation of inverses between pts. 1 and 9, 2 to 7, and 7 to 3 as generated in the preceding examples:

Step	Instructions	Input	Function	Display
1	Initialize with 1st inverse pt. #	1	[I]	PRT. COORDS?*
2	Answer Y for yes (If desired & printer is on)	N	R/S	INV. TO PT.?
3	Input pt. # being inversed to, R/S	9	R/S	Az=60.16 29

4	Press R/S to view distance		R/S	DIST=1240.622
5	Put in next pt. #	2	[I]	INV. TO PT.?
6	Input pt. # being inversed to, R/S	7	R/S	Az=44.2620
7	Press R/S to view distance		R/S	DIST=771.2361
8	Input next pt. #	3	R/S	Az=236.3113
9	Press R/S to view distance		R/S	DIST=758.9119

*This question will be displayed upon execution of this routine only once after the HP41C is turned on and only if a printer is plugged in. You may elect to change this status during subsequent inversing by simply setting or clearing flag 10 (See Appendix A).

Note: Step 8 above discloses the ease with which you can automatically inverse between sequential points. Had the HP#82143A peripheral printer been attached and enabled at Step 2, both pt. numbers with their associated coordinate values would have been printed, along with calculated azimuth and distance. Otherwise, had the printer been on and a "N" entered as in Step 2, only the pt. numbers being inversed between and the calculated azimuth and distance would have been printed. By inversing all perimeter points of an enclosed polygon, area is calculated by the D.M.D. method. (See next section)

3.7 INTERSECTIONS

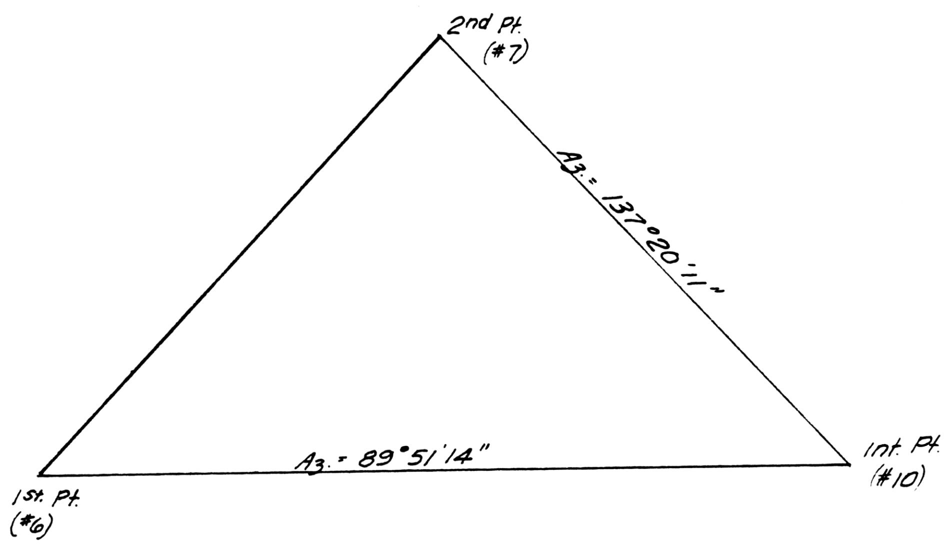
These routines will allow you to perform the following three intersections quickly and accurately: Azimuth-Azimuth, Azimuth-Distance and Distance-Distance. The routine used to store the values of the two known coordinate points allows you to try the different intersection types without re-entering the point numbers.

a. Intersection setup; before beginning, make sure you assign IS to the shifted A key (ASN, ALPHA, IS, ALPHA). In the routine which follows, points 6 and 7 as generated above will be used as the known points for these intersections:

Step	Instructions	Input	Function	Display
1	Input 1st pt., initialize	6	[A]	INPT AZ,D
2	Input 2nd pt.	7	shifted A	5625.1768

The display contains the northing values of point #7. You may now proceed with a particular intersection type.

b. Azimuth-Azimuth:



Note: This intersection is based upon the following data:

Az.1 (from pt. 6) = N89° 51'14"E

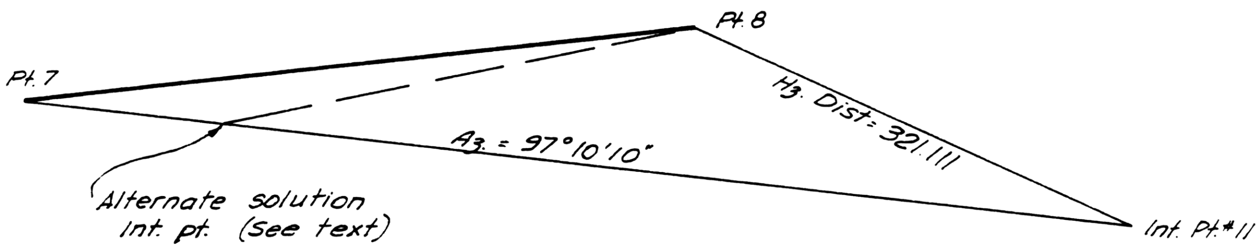
Az.2 (from pt. 7) = S42° 39'49"E

Assuming step a above has been performed, execute the following:

Step	Instructions	Input	Function	Display
1	Initialize		[F]	Az 1 =
2	Input Az. 1	89.5114	R/S	Az 2 =
3	Input Az. 2	137.2011	R/S	INT. PT=?
4	Input next available			

pt. #		10	R/S

c. Azimuth-Distance:



Note: This intersection is based upon the following data:

Az. (from pt. 7) = 97°10'10"

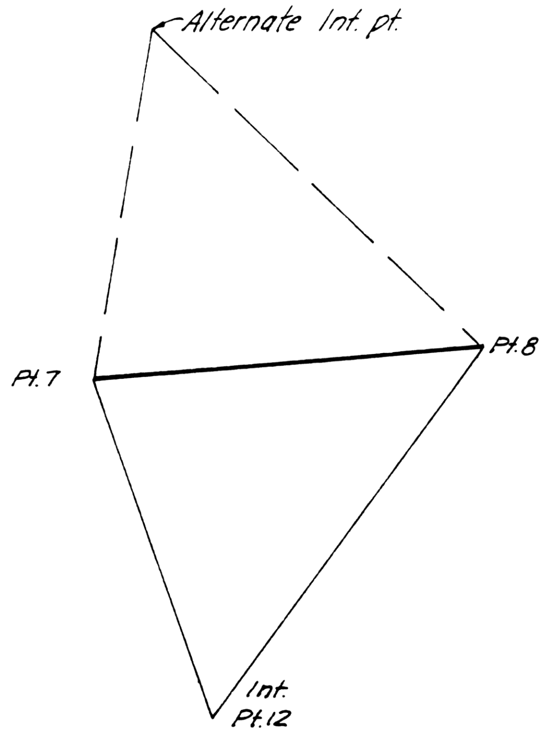
Dist. (from pt. 8) = 321.111

Assuming step a above has been performed, execute the following:

Step	Instruction	Input	Function	Display
1	Initialize		Shift [F]	Az=?
2	Input azimuth	97.1010	R/S	DIST =?
3	Input distance	321.111	R/S	INV. PT.?
4	Input next available pt. #	11	R/S	

To solve for the far solution, input the azimuth minus 180° in step 2 above.

d. Distance-Distance:



This intersection is based upon the following data:

Dist 1 (from pt. 7) = 382.112

dist 2 (from pt. 8) = 493.102

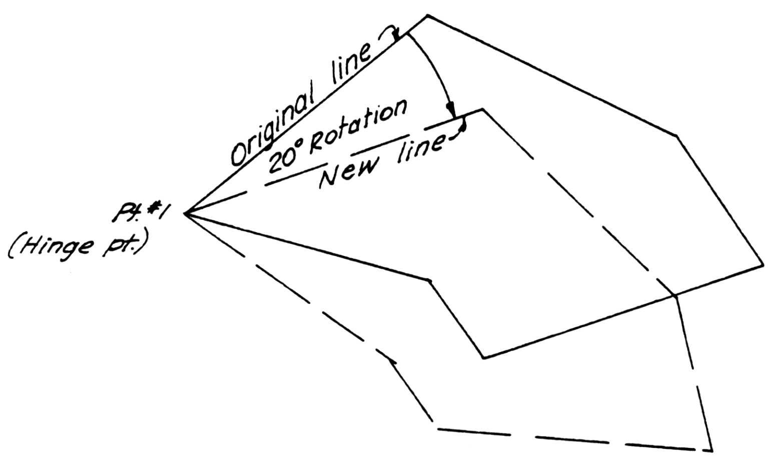
Since these points were entered as a part of the previous example, you need not perform step a above again.

Step	Instructions	Input	Function	Display
1	Initialize		Shift [G]	Dist. 1=?
2	Input distance 1	382.112	R/S	Dist. 2=?
3	Input distance 2	493.102	R/S	Int. Pt.=?
4	Input next available pt. #	12	R/S	

This routine calculates the intersection of the distances based upon a clockwise rotation of the line radiating from pt. 1 to the line radiating from pt. 2. To calculate the 2nd solution to this intersection, reverse the order of data input; treat pt. 8 as the 1st pt. and pt. 7 as the 2nd pt. Then using their respective distances the intersection will be performed as above.

3.8 ROTATION

This routine rotates an entire set of coordinates (up to the maximum available) to a known bearing or azimuth.



Note: this sketch is given as illustration only, and is not related to the example given.

Procedurally, coordinate pairs are sequentially recalled into the stack, inversed to a reference pt. (#1) and the rotation value is added and subtracted from the resultant azimuth. Next, the new azimuth is used, with the inversed distance to re-traverse to the new pt., which is assigned the original pt. #. Because the unrotated values are "written over" by this process, you should record them on magnetic cards prior to initialization (just in case an error is made along the line).

It may be desirable, from a drafting or a plotting point of view, to always use large, even integers for point #1 such as 5000 and 10000 or the like, since the values of pt. #1 do not change during the rotation routine.

In the following example we will rotate points 6,7,8,9,10,11 and 12 by 20° in a clockwise direction. As a check you may subsequently inverse between 2 or more of the points to insure the distances between them remains the same as the previously measured distances and the difference in azimuths is 20°. A clockwise rotation is entered as a plus value, and a counter-clockwise rotation is entered as a negative value, at the appropriate prompt. By setting Flag 18, you will no longer be asked to affirm the "over-write" of old coordinate values. Refer to Appendix A for further explanations of various flag settings

Step	Instructions	Input	Function	Display
1	Initialize		[G]	ROT. ∢=?
2	Input rotation < (↻=+,↺=-)	20.0000	R/S	NEW PT.=?
3	Input 1st pt. to be rotated	6	R/S	*

This routine will sequentially rotate all successive coordinate points, until you stop it by pressing R/S.

* If Flag 18 was clear, you would be requested to affirm the "writing-over" of the unrotated values at this step for this and subsequent points, by pressing R/S for each point after the "beep" tune is played. Furthermore, setting Flags 01 and 18 makes the execution of this routine <u>fully</u> automatic, with points being sequentially rotated until routine is terminated by pressing R/S.

Chapter 4

PROGRAM LISTING

```
01♦LBL "CH"
02♦LBL H
03 AON
04 "≠"
05 AOFF
06 -23
07 PASN
08 FIX 0
09 CF 00
10 CF 02
11 CF 03
12 36
13 PSIZE
14 "TO CD.?
=R/S"
15 PROMPT
16♦LBL E
17 "FIRST P
T.=?"
18 PROMPT
19 STO 33
20 STO 00
21 ST+ X
22 1
23 -
24 AON
25 "SP"
26 AOFF
27 SEEKPTA
28 "LAST PT
.=?"
29 PROMPT
30 STO 34
31 RCL 33
32 -
33 16
34 X<Y?
35 XEQ a
36 X<>Y
37 1
38 +
39 X<>Y
40 X>Y?
41 XEQ C
42♦LBL d
43 1.032
44 FIX 3
45♦LBL "Σ"
46 FS? 02
47 XEQ D
48 GETRX
49 WDTAX
50 SF 14
51 FS? 00
52 XEQ e
53 RCL 33
54 "PTS. "
55 ARCL X
56 AVIEW
57 PSE
58 RCL 34
59 "THRU "
60 ARCL X
61 "⊢ STORE
D"
62 AVIEW
63 STOP
64♦LBL e
65 RCL 35
66 16
67 +
68 XEQ c
69♦LBL a
70 SF 00
71♦LBL c
72 RCL 34
73 X>Y?
74 XEQ A
75 CF 00
76♦LBL B
77 RCL 34
78 RCL 35
79 -
80 2
81 *
82 2
83 +
84 .001
85 *
86 1
87 +
88 XEQ "Σ"
89♦LBL A
90 X<>Y
91 STO 35
92 XEQ d
93♦LBL C
94 SF 03
95 RCL 33
96 STO 35
97 XEQ B
98♦LBL D
99 RDTAX
100 SAVERX
101 FS? 00
102 XEQ e
103 "XMEM LD
D"
104 AVIEW
105 STOP
106♦LBL "≠"
107 SF 02
108 XEQ E
109 .END.
```

Card and Memory Load

Compass Adj. / Rotate

```
01♦LBL "CO"
02 CF 03
03 RCL 11
04 STO 08
05 RCL 14
06 -
07 RCL 15
08 /
09 STO 03
10 RCL 12
11 STO 05
12 RCL 13
13 -
14 RCL 15
15 /
16 STO 04
17 CF 07
18 "FRST. P
T.=?"
19 PROMPT
20 STO 16
21 GTO 08
22♦LBL "CT
23 1
24 ST+ 16
25 RCL 16
26♦LBL 08
27 PSE
28 "OLD"
29 FS? 21
30 PRA
31 XEQ "PR"
32 X<>Y
33 STO 06
34 RCL 12
35 -
36 ST+ 05
37 X<>Y
38 STO 07
39 RCL 11
40 -
41 ST+ 08
42 R-P
43 STO 09
44 RCL 03
45 *
46 ST+ 08
47 RCL 09
48 RCL 04
49 *
50 ST+ 05
51 RCL 06
52 STO 12
53 RCL 07
54 STO 11
55 RCL 05
56 RCL 08
57 RCL 10
58 PSE
59 "NEW PT.
="
60 ARCL X
61 FS? 21
62 PRA
63 SF 08
64 1
65 ST+ 10
66 RDN
67 XEQ "DE"
68♦LBL G
69 0
70 X<>F
71 CF 08
72 CF 09
73 CF 14
74 CLΣ
75 FC? 55
76 CF 10
77 FC? 55
78 CF 21
79 "ROT. ∡=
"
80 PROMPT
81 "ROT. ∡=

82 ARCL X
83 FC? 21
84 AVIEW
85 STO 10
86 FS? 21
87 ADV
88 1
89 XEQ "PR"
90 STO 18
91 STO 12
92 X<>Y
93 STO 19
94 STO 11
95 "NEW PT.
= ?"
96 PROMPT
97 STO 08
98♦LBL "NT"
99 CF 10
100 1
101 FS? 14
102 ST+ 08
103 RCL 08
104 FS? 21
105 PRX
106 XEQ "PR"
107 STO 14
108 X<>Y
109 STO 13
110 RCL 19
111 -
112 RCL 14
113 RCL 18
114 -
115 X<>Y
116 R-P
117 X<>Y
118 X>0?
119 XEQ 08
120 360
121 +
122♦LBL 08
123 RCL 10
124 HR
125 +
126 STO 03
127 X<>Y
128 P-R
129 RCL 19
130 STO 12
131 RDN
132 ST+ 12
133 X<>Y
134 RCL 18
135 STO 11
136 RDN
137 ST+ 11
138 RCL 12
139 RCL 11
140 RCL 08
141 SF 14
142 XEQ "DE"
143 END
```

Intersections

```
 01♦LBL "IS"
 02 SF 04
 03 "Σ"
 04 -21
 05 PASN
 06 RDN
 07 "%"
 08 -22
 09 PASN
 10 RDN
 11 "2ND PT.
="
 12 ARCL X
 13 XEQ "PR"
 14 STO 14
 15 RDN
 16 STO 13
 17 RTN
 18♦LBL F
 19 "AZ.-AZ.
 INT."
 20 AVIEW
 21 XEQ "IN
 22 "AZ. 1=
"
 23 PROMPT
 24 "AZ. 1=
 25 ARCL X
 26 AVIEW
 27 HR
 28 STO 15
 29 RDN
 30 "AZ. 2=?
"
 31 PROMPT
 32 "AZ. 2="
 33 ARCL X
 34 AVIEW
 35 HR
 36 STO 09
 37 -
 38 CHS
 39 SIN
 40 *
 41 RCL 09
 42 RCL 15
 43 -
 44 SIN
 45 /
 46 STO 16
 47♦LBL 07
 48 RCL 15
 49 RCL 16
 50 P-R
 51 RCL 12
 52 +
 53 X<>Y
 54 RCL 11
 55 +
 56 "INT. PT
.=?"
 57 PROMPT
 58 XEQ "DE"
 59♦LBL "Σ"
 60 "AZ.-DST
. INT:"
 61 AVIEW
 62 XEQ "IN"
 63 "AZ. =?"
 64 PROMPT
 65 "AZ. ="
 66 ARCL X
 67 AVIEW
 68 HR
 69 STO 15
 70 -
 71 CHS
 72 X<>Y
 73 P-R
 74 X<>Y
 75 X↑2
 76 "DIST. =
"
 77 PROMPT
 78 "DIST. =
"
 79 ARCL X
 80 AVIEW
 81 X↑2
 82 -
 83 CHS
 84 SQRT
 85 +
 86 STO 16
 87 GTO 07
 88♦LBL "%"
 89 "D-D INT
."
 90 FS? 21
 91 PRA
 92 "DIST. 1
=?"
 93 PROMPT
 94 STO 05
 95 "DIST. 1
="
 96 ARCL X
 97 FS? 21
 98 PRA
 99 "DIST. 2
=?"
100 PROMPT
101 STO 04
102 "DIST. 2
="
103 ARCL X
104 FS? 21
105 PRA
106 XEQ "IN"
107 X<>Y
108 STO 07
109 X↑2
110 RCL 05
111 X↑2
112 +
113 RCL 04
114 X↑2
115 -
116 2
117 /
118 RCL 07
119 RCL 05
120 *
121 /
122 ACOS
123 -
124 RCL 05
125 P-R
126 RCL 12
127 +
128 X<>Y
129 RCL 11
130 +
131 "INT. PT
.=?"
132 PROMPT
133 XEQ "DE"
134 .END.
```

```
01◆LBL "TR"
02◆LBL "DE"
03 SF 00
04◆LBL "PR"
05 CF 06
06 FS?C 00
07 SF 06
08 FS? 06
09 XEQ 02
10 STO 01
11 0
12 STO 00
13 RDN
14◆LBL 02
15 AON
16 "SP"
17 ASTO 02
18 AOFF
19 "PT. "
20 ARCL X
21 FS? 14
22 PSE
23 FS? 21
24 PRA
25 FS? 06
26 XEQ 03
27 CLA
28 AON
29 ARCL 02
30 AOFF
31 ST+ X
32 1
33 -
34 SEEKPTA
35 GETX
36 GETX
37 FS? 10
38 PRX
39 X<>Y
40 FS? 10
41 PRX
42 FS? 01
43 XEQ 11
44 RTN
45◆LBL 11
46 1
47 ST+ 01
48 RCL 01
49 XEQ 02
50◆LBL 03
51 CLA
52 ST+ X
53 1
54 -
55 AON
56 ARCL 02
57 AOFF
58 SEEKPTA
59 GETX
60 FS? 18
61 XEQ 06
62 X>0?
63 BEEP
64 X>0?
65 STOP
66◆LBL 06
67 RDN
68 SEEKPTA
69 RDN
70 SAVEX
71 RDN
72 SAVEX
73 FS? 14
74 CF 21
75 "N="
76 ARCL X
77 FS? 10
78 XEQ 12
79 FS? 21
80 PRA
81◆LBL 12
82 R↑
83 "E="
84 ARCL X
85 FS? 10
86 XEQ 23
87 FS? 21
88 PRA
89◆LBL 23
90 FS? 21
91 ADV
92 FS? 03
93 XEQ 15
94 FS? 07
95 XEQ c
96 FS?C 08
97 XEQ "CT"
98 FS?C 05
99 XEQ 04
100 FS? 14
101 XEQ "NT"
102 STOP
103◆LBL I
104 4
105 X<>F
106 RDN
107 CF 08
108 CF 10
109 CF 14
110 0
111 STO 17
112 STO 18
113 RDN
114 STO T
115 FS? 21
116 ADV
117 FS? 25
118 XEQ 01
119 AON
120 "PRT. CO
ORDS?"
121 FS? 21
122 STOP
123 AOFF
124 ASTO Y
125 "Y"
126 ASTO X
127 X=Y?
128 SF 10
129 R↑
130◆LBL 01
131 SF 25
132 "INV:"
133 FS? 21
134 PRA
135 XEQ "PR"
136 RDN
137 STO 12
138 R↑
139 STO 11
140 "INV. TO
 PT.?"
141 PROMPT
142◆LBL 18
143 XEQ "PR"
144 STO 14
145 X<>Y
146 STO 13
147 XEQ "IN"
148 HMS
149 "AZ="
150 ARCL X
151 AVIEW
152 FC? 21
153 STOP
154 X<>Y
155 "DIST="
156 ARCL X
157 AVIEW
158 STOP
159 RCL 13
160 STO 12
161 RCL 14
162 STO 11
163 RDN
164 RDN
165 XEQ 18
166◆LBL A
167◆LBL "AB"
168 020
169 PSIZE
170 16
171 X<>F
172 RDN
173 RDN
174 CF 08
175 CF 09
176 CF 10
177 CF 12
178 CF 13
179 CF 14
180 CF 15
81 CF 18
82 SF 27
83 ΣREG 11
84 DEG
85 FIX 4
86 "AB"
87 11
88 PASN
89 "DE"
90 14
91 PASN
92 "PR"
93 15
194 PASN
195 R↑
196 "TRV. FR
M "
197 FS? 21
198 PRA
199 STO 03
200◆LBL 13
201 XEQ "PR"
202 0
203 STO 17
204 STO 18
205◆LBL 16
206 FS?C 04
207 CLΣ
208 FC? 07
209 RDN
210 STO 11
211 X<>Y
212 STO 12
213 FS? 09
214 RTN
215 FS? 03
216 XEQ c
217◆LBL 04
218 1
219 ST+ 03
220 RDN
221 "INPT. A
Z, D"
222 FC? 09
223 PROMPT
224 "DIST.=?
"
```

```
225 FS? 09
226 PROMPT
227 X<>Y
228◆LBL 19
229 "AZ.= "
230 ARCL X
231 FC? 09
232 AVIEW
233 X<>Y
234 "HRZ. DST.="
235 ARCL X
236 AVIEW
237 X<>Y
238 HR
239 X<>Y
240 ST+ 15
241 P-R
242 ST+ 12
243 X<>Y
244 FC? 02
245 XEQ 10
246 FS? 02
247 ST+ 11
248 RCL 12
249 RCL 11
250 CF 22
251 "PT.  "
252 ARCL 03
253 "⊢?"
254 PROMPT
255 FC?C 22
256 RCL 03
257 STO 03
258 SF 05
259 XEQ "DE"
260◆LBL B
261 0
262 X<>F
263 CF 08
264 SF 09
265 CF 12
266 CF 14
267 CF 15
268 CF 18
269 SF 20
270 FS? 21
271 ADV
272 "INSTR. PT.=?"
273 PROMPT
274 "FLD. DTA TRV:"
275 FS? 21
276 PRA
277 XEQ 20
278◆LBL c
279 SF 09
280 "HRZ. ∡ TRND=?"
281 PROMPT
282 "HRZ. ∡ ="
283 ARCL X
284 FS? 21
285 CLA
286 HR
287 RCL 07
288 +
289 360
290 MOD
291 FS? 03
292 XEQ 14
293◆LBL 05
294 STO 07
295 180
296 ST+ 07
297 RDN
298◆LBL 14
299 HMS
300 "NEW AZ= "
301 ARCL X
302 AVIEW
303 SF 07
304 1
305 ST+ 03
306 CF 22
307 "VERT. ∡ =?"
308 PROMPT
309 FC?C 22
310 90
311 "VERT. ∡ = "
312 ARCL X
313 FS? 21
314 PRA
315 HR
316 SIN
317 "SL. DIST. =?"
318 PROMPT
319 "SL. DIST.= "
320 ARCL X
321 FS? 21
322 PRA
323 *
324 R↑
325 XEQ 19
326◆LBL b
327 8
328 X<>F
329 CF 08
330 SF 09
331 CF 12
332 CF 14
333 CF 18
334 CF 20
335 CLΣ
336 "SDSHOT FRM=?"
337 PROMPT
338 "SDSHOT FROM:"
339 FS? 21
340 PRA
341◆LBL 20
342 STO 03
343 STO 19
344 XEQ 13
345◆LBL 17
346 "BCKSTE PT.=?"
347 PROMPT
348 "BCKSITE :"
349 FS? 21
350 PRA
351 XEQ "PR"
352 STO 14
353 RDN
354 STO 13
355 XEQ "IN"
356 HMS
357 "BK. AZ. = "
358 ARCL X
359 FS? 21
360 PRA
361 HR
362 STO 07
363 RDN
364 SF 04
365 XEQ c
366◆LBL 15
367 RCL 19
368 XEQ "PR"
369 GTO 16
370◆LBL "IN"
371 RCL 14
372 RCL 11
373 -
374 STO 06
375 RCL 13
376 RCL 12
377 -
378 STO 07
379 FS? 02
380 XEQ 10
381 RCL 06
382 RCL 07
383 R-P
384 X<>Y
385 360
386 MOD
387 RTN
388◆LBL 10
389 ST+ 17
390 FC? 02
391 ST+ 11
392 2
393 /
394 RCL 17
395 -
396 *
397 ST+ 18
398 RCL 12
399 RCL 11
400 RTN
401◆LBL J
402 RCL 18
403 ABS
404 43560
405 /
406 "AREA="
407 ARCL X
408 "⊢ ac."
409 AVIEW
410 END
```

Chapter 5

REGISTERS AND FLAG STATUS

The below listed tables allow you to determine which storage registers and flags are used in the various routines. you will note heavy usage of some registers, such as 0, 2, and 11 thru 14. Conversely, registers 4, 10, and 19 are hardly used. Flag 19 is the least used flag, while flags 00. 02, 10, 14, 18, and 21 are heavily used. Any additions, amendments or deletions to this program must be implemented with these usages in mind. Furthermore, if utility or other routines are used in conjunction with the one under consideration, the impact of the added routine must be looked at also. The define and print point routines are the utility routines which are used as part of a majority of the remaining routines.

REGISTER USEAGE:

ROUTINE	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	Define Point	Print Point	Remarks
TRAVERSE				*			*	*				*	*			*		*	*		*	*	
INVERSE							*	*				*	*	*	*			*	*			*	
FLD.DATA TRAV.				*			*	*				*	*	*	*	*		*	*	*	*	*	
SIDESHOTS				*			*	*				*	*	*	*	*		*	*	*	*	*	Area not calculated
COMPASS ADJ.				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*	Inverse routine prior
INT. SETUP	*	*	*	*			*	*				*	*	*	*	*		*	*		*	*	Traverse routine prior
AZ.-AZ. INT.							*	*		*		*	*	*	*	*	*				*		
AZ.-DIST. INT.							*	*				*	*	*	*	*	*				*		
DST.-DST. INT.					*	*	*	*				*	*	*	*						*		
ROTATION				*					*		*	*	*	*	*				*	*	*	*	
AREA																			*				Actual calc. inclds TR or IN
DEFINE POINT			*																				
PRINT POINT	*	*	*																				

* Indicates register used

FLAG USEAGE:

ROUTINE	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	Define Point	Print Point	Remarks/other flags used
TRAVERSE			T	T	T_c			T		T												T			T_c = 22
INVERSE			T_1								P											T		X	T=25:T_1= on area sbroutine
FLD.DATA TRAV.			T		T_c			T		T												T	X	X	T_c = 22
SIDESHOTS			T	T	T_c			T		T												T	X	X	T_c = 22
COMPASS ADJ.																						T	X	X	
INT. SETUP																								X	TR prtn. used prior
AZ.-AZ. INT.																							X		Prt. Pt. used in setup
AZ.-DIST. INT.																							X		" " " " "
DST.-DST. INT.																						T	X		" " " " "
ROTATION											T				T							~~P~~T	X	X	T = 55
AREA																									Calc'd in TR or IN
DEFINE POINT	T			T		T_c			T_c	T	T				T				T			P			
PRINT POINT	T	T					T				T				T							T			

SYMBOLS: T = Tested

T_c= Tested & Cleared

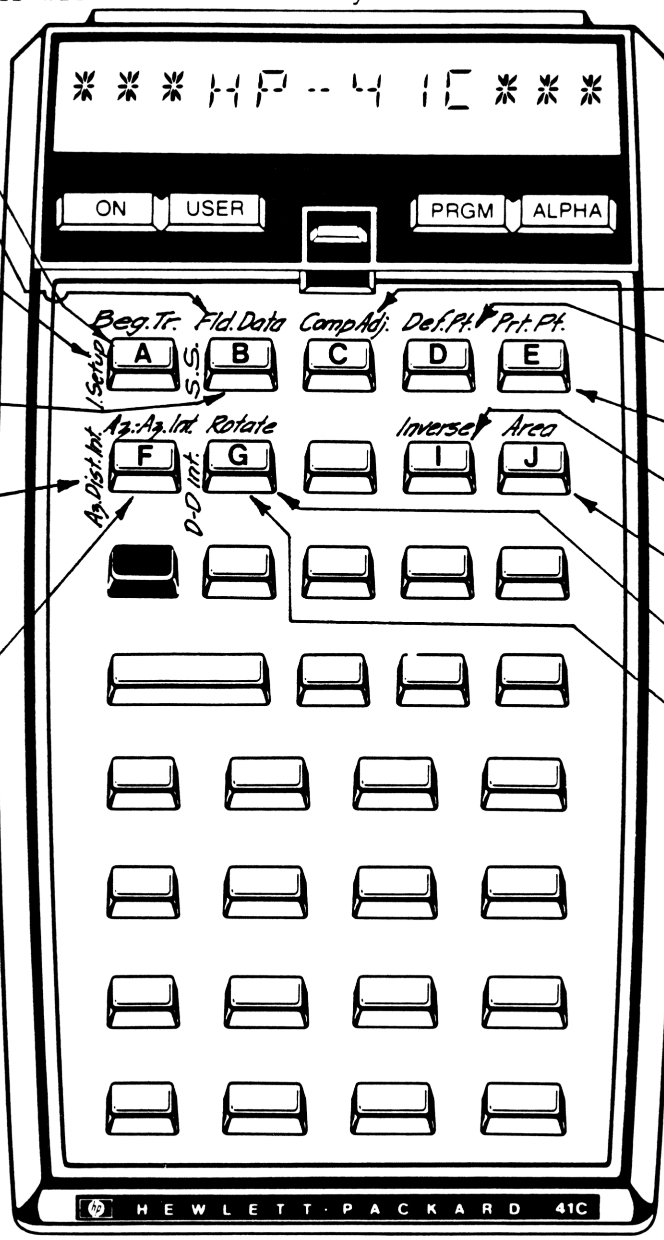
P = Possible Status Change

X = Utility Routine used

Note: At the beginning of some routines a flag status "setup" is executed to insure proper routine interaction. In addition, the status of all flags at the time the HP41C is turned on can be determined in accordance with Section 14 of your Handbook & Programming Guide.

KEYBOARD CARD LABELING

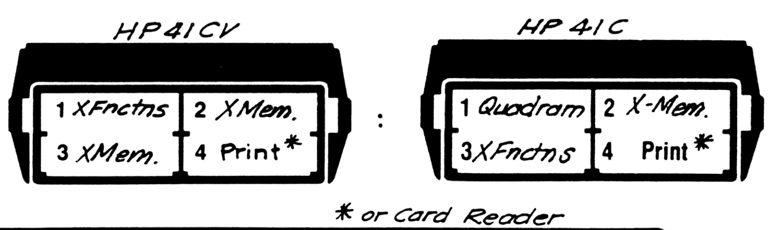
Note: with all calculation programs loaded, the following keyboard assignments will be automatically made:



KEYBOARD

You may wish to move these around to fit your particular system. (see text)

SYSTEM CONFIGURATION



CARD

Chapter 7

PERIPHERALS

A. Card reader: The HP82104 Card Reader provides a quick and easy method of transferring programs or data between magnetic cards and the HP41C. Refer to Section 2.2 herein for more information.

B. Printer: The HP82143A Printer permits you to have a printed record of coordinates and calculations. This program has accordingly been designed with the printer usage in mind. If one is hooked up, you must decide whether or not to print coordinates or data.

C. Cassette: Because this is one peripheral not considered by these programs, you should take a close look at flag usage herein to insure no conflicts.

Chapter 8

COMPATIBILITY WITH HP#00041-15005 SURVEY ROM

The following HP Survey ROM routines will clear any coordinate or calculation values you may have stored in main memory: TRAVERSE, INVERSE AND SIDESHOTS, COMPASS RULE ADJUSTMENT, TRANSIT RULE ADJUSTMENT, VOLUME BY AVERAGE END AREA, and VOLUME BY BORROW PIT. The extended memory containing your coordinates will not be destroyed by anything short of a master clear.

Remaining routines in the ROM such as the various curve calculations, resection, predetermined area, coordinate transformation and offset from a point to a line (in the intersections routine) are not aviable in our program. Coordinate pairs generated within the aforementioned routines can be defined arbitrary point numbers by "jumping" from ROM to this program: (assumes a ROM routine has calculated the coordinate values)

Step	Instructions	Input	Function	Display
1	With easting displayed, enter next pt. #	n		n
2	Execute TR		XEQ TRa	n

If you have the survey ROM in your system, you can modify our programs to make use of the BRG and AZ routines, but you will sacrifice considerable time because the program must "jump" into ROM and back out as a part of program execution. This could for example allow you to print out all final bearings during the point to point inverse routine.

Chapter 9

EQUATIONS

COMPASS (BOWDITCH) RULE ADJ.

Equations:

$$C_L = \frac{(\Delta N)\,(Dist)}{\Sigma Dist}$$

$$C_D = \frac{(\Delta E)\,(Dist)}{\Sigma Dist}$$

Where:

C_L = Correction to latitude of a course

C_D = Correction to departure of a course

ΔN = Closing latitude

ΔE = " departure

Dist = length of course to be corrected

ΣDist = total length of traverse

ROTATION

Because this routine makes heavy use of the Inverse and Traverse routines, their equations will not be repeated here. The primary difference is the addition of the stored rotation angle to each Az. inversed automatically, and subsequent retraverse to the rotated point. Close study of this program section will help clarify the actual procedures used.

FIELD ANGLE OR BEARING TRAVERSE AND SIDESHOTS

Equations:

$$N_{i+1} = N_1 + \text{HDist} \cos \text{AZ}$$

$$E_{i+1} = E_1 + \text{HDist} \sin \text{AZ}$$

$$\text{Area} = \sum_{k=1}^{n} \text{LAT}_k \left(\tfrac{1}{2}\text{DEP}_k + \sum_{j=1}^{k-1} \text{DEP}_j \right)$$

where:

$\text{DEP}_k = E_{k+1} - E_k$ and $\text{LAT}_k = N_{k+1} - N_k$

Note: Area calculations are performed upon execution of either type of Traverse and Inverse. However, any sideshots calculated during or after a traverse routine will cause invalid area calculations.

INVERSE

Equations:

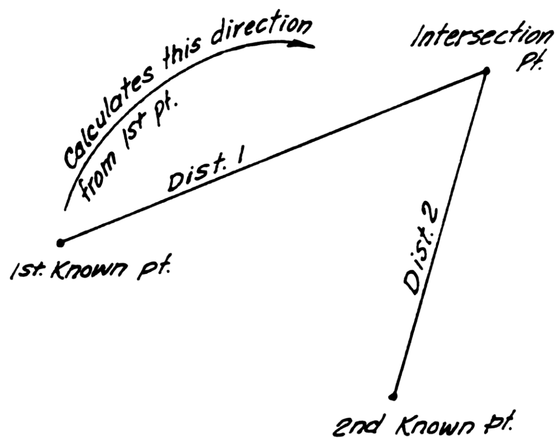
$$\text{HD} = (N_1 - N_{i-1})^2 + (E_1 - E_{i-1})^2$$

$$\text{AZ} = \tan^{-1} \frac{E_i - E_{i-1}}{N_i - N_{i-1}}$$

Area: same as above

DISTANCE - DISTANCE INTERSECTION

Given two lines, each of known length and originating from two known points, this program computes the intersection coordinates. There are two possible solutions; this program calculates the one found by proceeding in a clockwise direction from the first known point to the second known point. The other solution is found by reversing the entry of the point numbers.



EQUATIONS:

$$\phi = \cos^{-1} \frac{Dist_{12}^2 + Dist_1^2 - Dist_2^2}{2(dist_1)\,(dist_{12})}$$

$$AZ = \tan^{-1} \frac{E_2 - E_1}{N_2 - N_1}$$

$$N = N_1 + Dist_1 \cos\,(AZ - \phi)$$

$$E = E_1 + Dist_1 \sin\,(AZ - \phi)$$

where:

ϕ = Angle between line 1 & line 1-2

$Dist_{12}$ = Distance from 1st pt. to 2nd pt.

$Dist_1$ = Known distance along line 1

$Dist_2$ = " " " " 2

N_1E_1 = Northing, Easting of 1st pt

N,E = Northing, Easting of intersection point

AZ = Azimuth of line from 1st pt. to 2nd pt.

AZIMUTH - DISTANCE AND
AZIMUTH - AZIMUTH INTERSECTIONS

This program computes the coordinates of the point of intersection of two lines:

1) One of known azimuth through known coordinates and the other of known length from a point of known coordinates; or

2) When the azimuth of each line is known and the coordinates of a point on each line are lnown.

For the first case, both solutions may be computed.

Equations:

Azimuth - Distance

$$Az_{12} = \tan^{-1} \frac{E_2 - E_1}{N_2 - N_1}$$

$$h = Dist_{12} \sin\phi$$

$$b = \frac{\qquad}{Dist_2^2 - h^2}$$

$$N = N_1 + ((Dist_{12} \cos\phi) + b) \cos(Az_1)$$

$$E = E_1 + ((Dist_{12} \cos\phi) + b) \sin(Az_1)$$

where:

Az_{12} = Azimuth of line from 1st pt. to 2nd pt.

Az_1 = Azimuth 1

ϕ = Angle between line 1 and line from 1st pt. to 2nd pt.

h = Perpendicular distance from 2nd pt. to line 1

b = Distance from pt. of intersection to the point where the perpendicular (h) intersects line 1

$Dist_2$ = Known Dist.(line 2)

N_1E_1 = Northing, Easting of 1st pt.

N_2E_2 = Northing, Easting of 2nd pt.

$Dist_{12}$ = Distance from 1st pt. to 2nd pt.

Azimuth - Azimuth

$$N = N_1 + Dist(\cos AZ_1)$$

$$E = E_1 + Dist(\sin AZ_1)$$

$$Dist = \frac{Dist_{12} \sin(AZ_2 - AZ_{12})}{\sin(AZ_2 - AZ_1)}$$

where:

AZ_{12} = Azimuth of line from 1st pt. to 2nd pt.

AZ_1 = Azimuth of line 1

AZ_2 = " " " 2

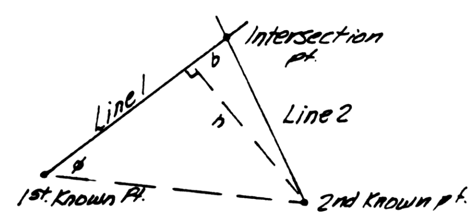
N_1E_1 = Northing, Easting of 1st pt.

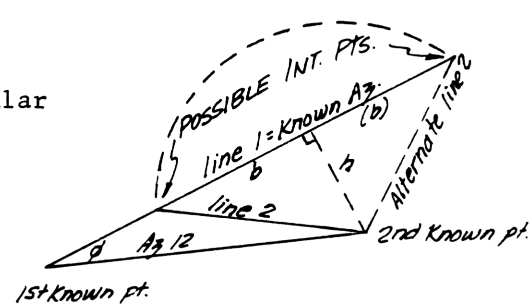
N_2E_2 = " " " 2nd pt.

N,E = Northing, Easting of intersection point

$Dist$ = Dist from 1st pt. to intersection

$Dist_{12}$ = Distance from 1st pt. to 2nd pt.





Chapter 10

SIZE REQUIREMENTS

The TR program uses the PSize function to set size 20 for all survey calculations. The same function is used to set size 299 when coordinate values are either being stored into extended memory or transferred to magnetic cards. With only the TR program loaded in, you will have used up 968 Bytes of memory. The intersections program adds another 366 Bytes, and inclusion of the Compass Adjust/Rotation Program adds another 269 Bytes, for a total of 1603 Bytes of memory. Ignoring the effect of other key assignments, you are left with 630 Bytes of main memory, which equals a maximum of 90 program lines. You may wish to keep in mind the fact that 7 one byte program lines equals 1 data register, and you began with a system user total of 2233 bytes.

Appendix A

FLAG USAGE

This program uses flags 00 thru 21 & 55, the latter being the only system flag tested. Three of the user flags can be controlled by you, allowing different printer and program routines to be enabled or ignored. The remaining 18 user flags are normally addressed only by the program itself. Their program usage is explained below, on a routine by routine basis.

Flag 01: Continuous Point Printout: If set, this flag allows the printer to print out the coordinate points (and their associated values, if flag 10 is set) sequentially. A continuous loop is enabled if flag 01 is set, incrementing a counter register by one during each loop iteration. If flag 01 is cleared, only the point number input or being generated is displayed or printed out.

Flag 10: Print Suppression: By setting this flag, coordinate points, field measurements and calculated data will be printed. Program line 128 sets this flag in accordance with you response to a question regarding data printout, within the inverse routine. However, two other routines test flag 10 for determination of data printout. Should you encounter a NONEXISTENT message during execution, you might first want to see if the printer is plugged in.

Flag 18: If you wish not to be appraised of the fact that coordinates reside in the storage register addressed in the defined point routine, set Flag 18. A test at line 060 causes the programs to execute routine 22, bypassing a short portion of the test routine.

A.1 TRAVERSE

Flag 03: Cleared at initialization and is tested at line 213.

If set, you are branched to the sideshot routine. If clear, program execution continues.

Flag 04: Set at initialization and is tested and cleared at line 203. If set, the registers are cleared as part of the set up procedure in the traverse routine. Regardless of status, program execution continues.

Flag 07: Is cleared at initialization and is tested at line 205. If set, a R statement will not be executed, as part of the sideshot routine. If clear, the point # in the X register is rolled down. In either case, program execution continues.

Flag 09: Is cleared at initialization and is tested at lines 210,219,222 and 228. The status of Flag 09 at these lines determines whether or not to return to another calling routine in the former test, and whether or not to display 3 different alpha messages or requests in the latter tests.

Flag 21: This is the printer enable flag. This Flag matches the status of flag 55 each time the HP41C is turned on. If a printer is plugged in, but not turned on, Flag 21 will still be set unless you change its status, and the "Printer Off" message will be displayed. Program execution stops.

Flag 22: Is the numeric input flag, and is tested and cleared at line 247. If set, user has entered numeric data (point #) which will be stored in reg. 03. If clear, the number residing in reg. 03 was affirmed by the user upon pressing R/S with no numeric input.

A.2 INVERSE

Flag 10: Is cleared at initialization and will (conditionally) be set at line 122, if an alpha Y was input following line 116. This is the answer input if you wanted the coordinate values printed during the inverse routine. Flag 10 is subsequently tested in the print point routine to determine if printing of coordinates was desired.

Flag 21: Printer enable flag. Same usage as in the traverse routine. (see above)

A.3 FIELD DATA TRAVERSE

Flag 04: Is cleared at initialization and is tested and cleared at line 203. If set, you are in traverse routine and the registers are cleared. If clear, these registers are not cleared. In either case, program execution continues.

Flag 07: Is cleared at initialization, is not set within this routine. It is tested at line 206. If set, the zero in the X register is not rolled down. If clear, a R↓ statement is executed, and program execution continues.

Flag 09: Is set at initialization and is tested at lines 210, 219, 222 and 228. Results of these tests will be exactly the opposite of those described for this flag in the traverse routine, with accordingly opposite actions or displays taking place.

Flag 21: Printer enable flag. Same usage as in the traverse routine. (see above)

Flag 22: Is tested at line 306 for numeric input, in the same manner as in the traverse routine.

A.4 SIDESHOTS ROUTINE

Flag 03: Is set at initialization and is tested at the end of the define point routine, line 92 for a return to the calling routine if set. It is also tested at line 288. If set, a portion of the field data traverse is skipped. If clear, the azimuth under consideration is added to 180 and stored. In either case program execution continues.

Flag 04: Is cleared at initialization and tested and cleared at line 203. Results of the test are the same as in the field data traverse routine.(see above)

Flag 07; Is cleared at initialization, set at line 300 and tested at line 206. If set, the point # recalled from register 19 at line 364 is rolled down: if clear, the traverse routine is in use. In either case program execution continues.

Flag 09: Is set at initialization and has the identical status, purpose and tests as it does in the field data traverse routine.(see above)

Flag 21: Printer enable flag. Same status and usage as in the traverse routine. (see above)

Flag 22: Is tested and cleared at line 290 for numeric input, in the same manner as in the traverse routine.

A.5 ROTATION

Flag 10: Is either set or cleared at initialization, depending upon other previously used routines. If coordinates were printed out in the inverse routine, they will also be printed out using this routine. If Flag 10 is set, printing will occur: if not, program execution by-passes the PRX or PRA command. If a printer is not plugged in, a test of Flag 55, printer existence flag at line 074 will accordingly clear Flag 10.

Flag 14: Is cleared upon initialization and is subsequently set at line 139. A test is encountered near the end of the define point routine line 101: if set, Flag 14 instructs execution of routine labeled NT, in the rotation program. If clear, program execution stops.

Flag 21: Printer enable flag will be automatically cleared at line 076 if Flag 55, the printer existence flag is clear.

A.6 DEFINE POINT

Flag 00: Is set at line 003 and is tested at line 006. If set, flag 06 is set. If clear, the print point routine continues.

Flag 03: Is either set or clear dependent upon previous routine execution. A test at line 092 sends program execution to line 363, LBL 15, if set. If clear, program execution continues.

(Sideshot routine)

Flag 05: Is either set or clear dependent upon previous routine execution. A test at line 098 branches program execution to line 214, LBL TH, if set. If clear, program execution continues. Flag 05 is cleared automatically following the test. (Traverse routine)

Flag 06: Is cleared in line 005 of the print point routine, and is set within the define point routine, dependent upon the status of flag 02, at line 07. It is then tested at lines 008 and 025.

Flag 07: Is either set or clear dependent upon prior routine execution. (Sideshots or field data traverse) A test at line 094 sends program execution to line 275, if set. If clear, program execution continues within this routine.

Flag 08: Is either set or clear dependent upon previous routine execution. A test at line 096 branches program execution to line 022 of the CO program, if set. If clear, program execution continues. Flag 08 is cleared automatically, following the test. (Compass Rule Adjust routine)

Flag 18: Is not set within the program. If set, the user has decided to have this routine disregard the test or storage registers to see if coordinate values are resident. The test at line 060 bypasses 4 steps of this routine, lines 062 thru 065, if set.

Flag 21: Is set or clear dependent upon previously executed routines, or printer presence. Furthermore, a test at line 074 of Flag 14 will clear this flag, if flag 14 was set. Flag 14 is only set within the rotation routine. Three subsequent tests at lines 080, 087, and 090 command 2 "PRA"s and an "ADV" statement if Flag 21 is set.

A.7 PRINT POINT

Flag 00: Is tested and cleared at line 006. If set, flag 06 is set and the define point routine is activated.

Flag 01: Is set by the user to continuously print out a "BLOCK" of coordinate pairs. A test at line 042 will cause program execution to branch to line 045, if Flag 01 is set. If clear, the print point routine is terminated and returned to the calling routine.

Flag 10: Is either set or clear based upon prior routine execution or user command. A test at lines 037 and 040 will cause execution of a "PRX" command, if Flag 10 is set. If clear, the "PRX" command is jumped over and in either case, program execution continues.

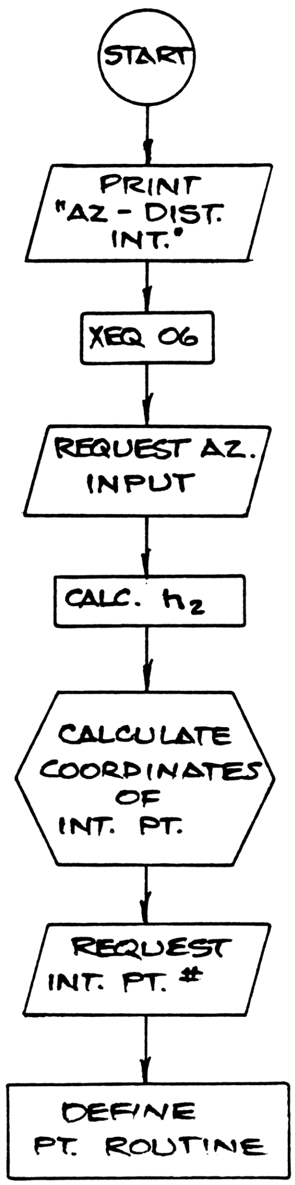
Flag 14: Is either set or clear based upon prior routine execution. (Rotation) If set, the PSE command at line 022 is executed following the test at line 021. If clear, this pause is not executed and program execution continues.

Flag 21: Is set or clear based upon prior routine execution, or printer presence. A test a line 024 will result in execution of a "PRA" command, if this flag is set. If clear, this command is jumped; in either case program execution continues.

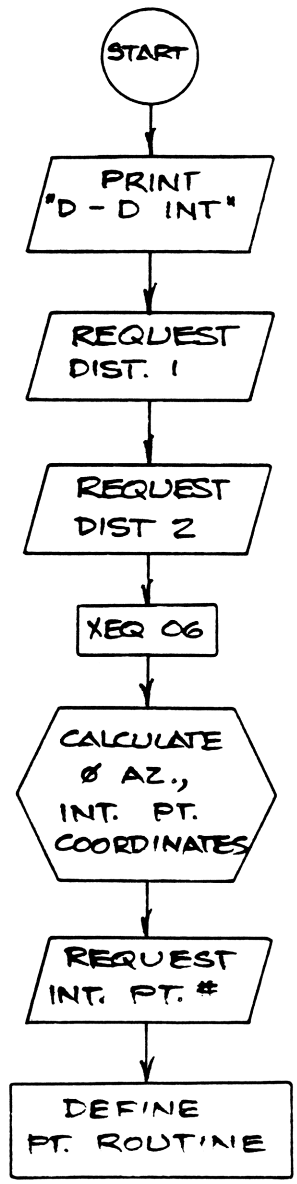
Appendix B

FLOW CHARTS

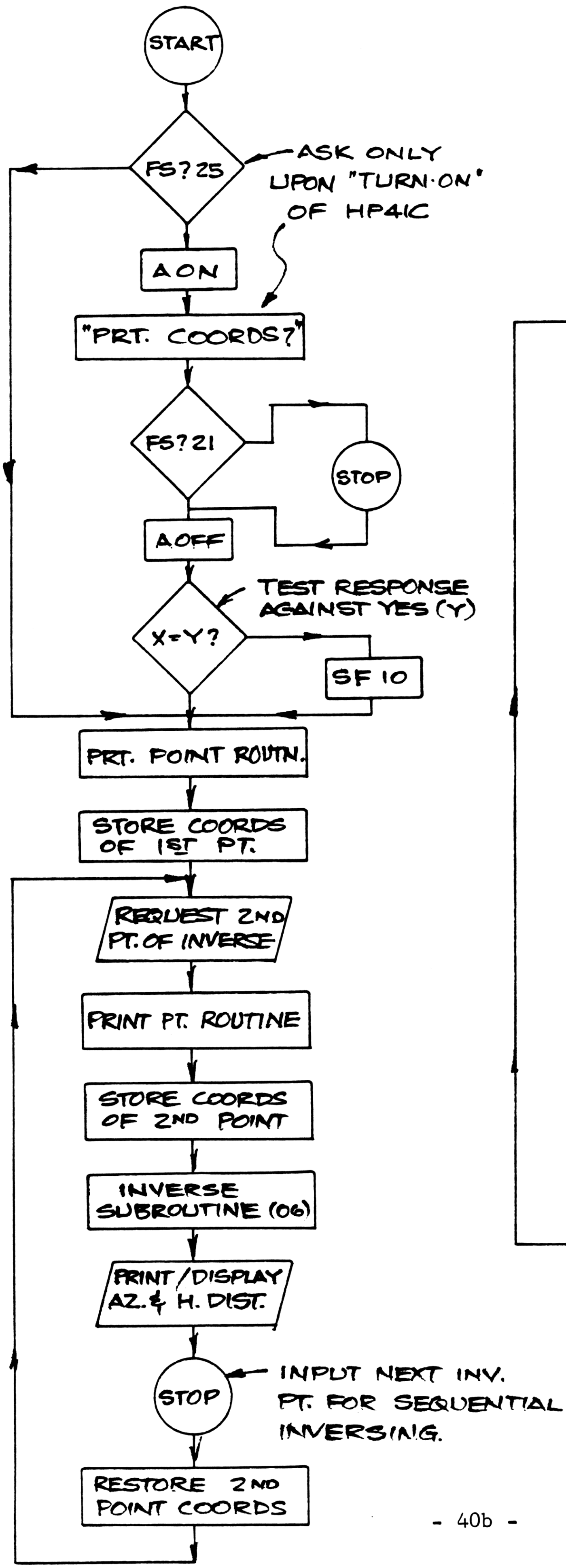
AZ.-DIST. INT.



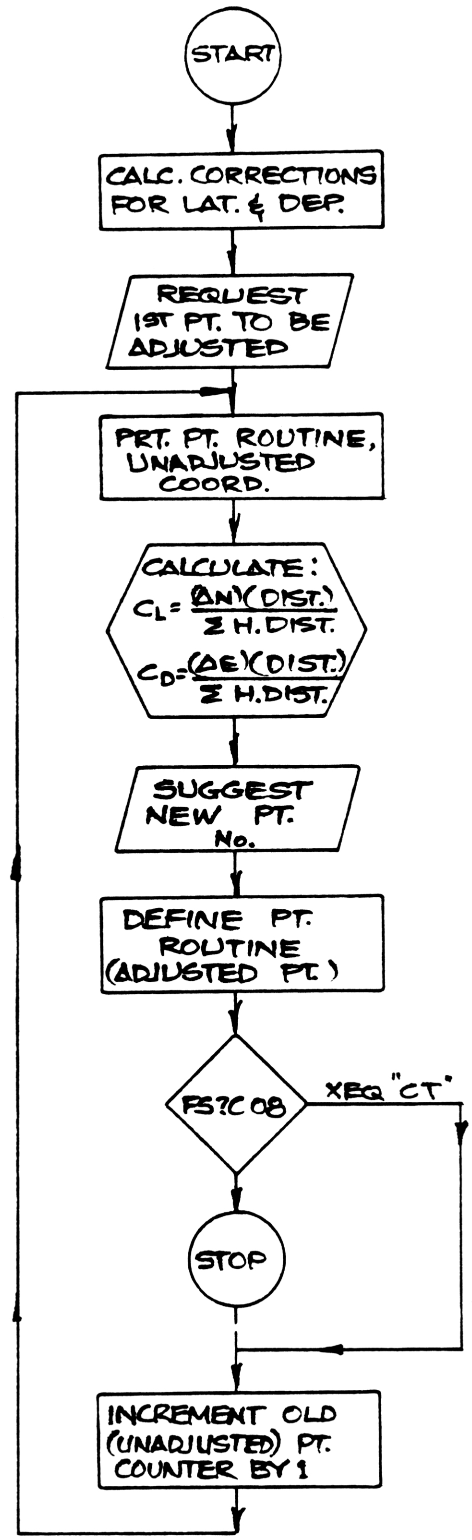
DIST.-DIST. INT.



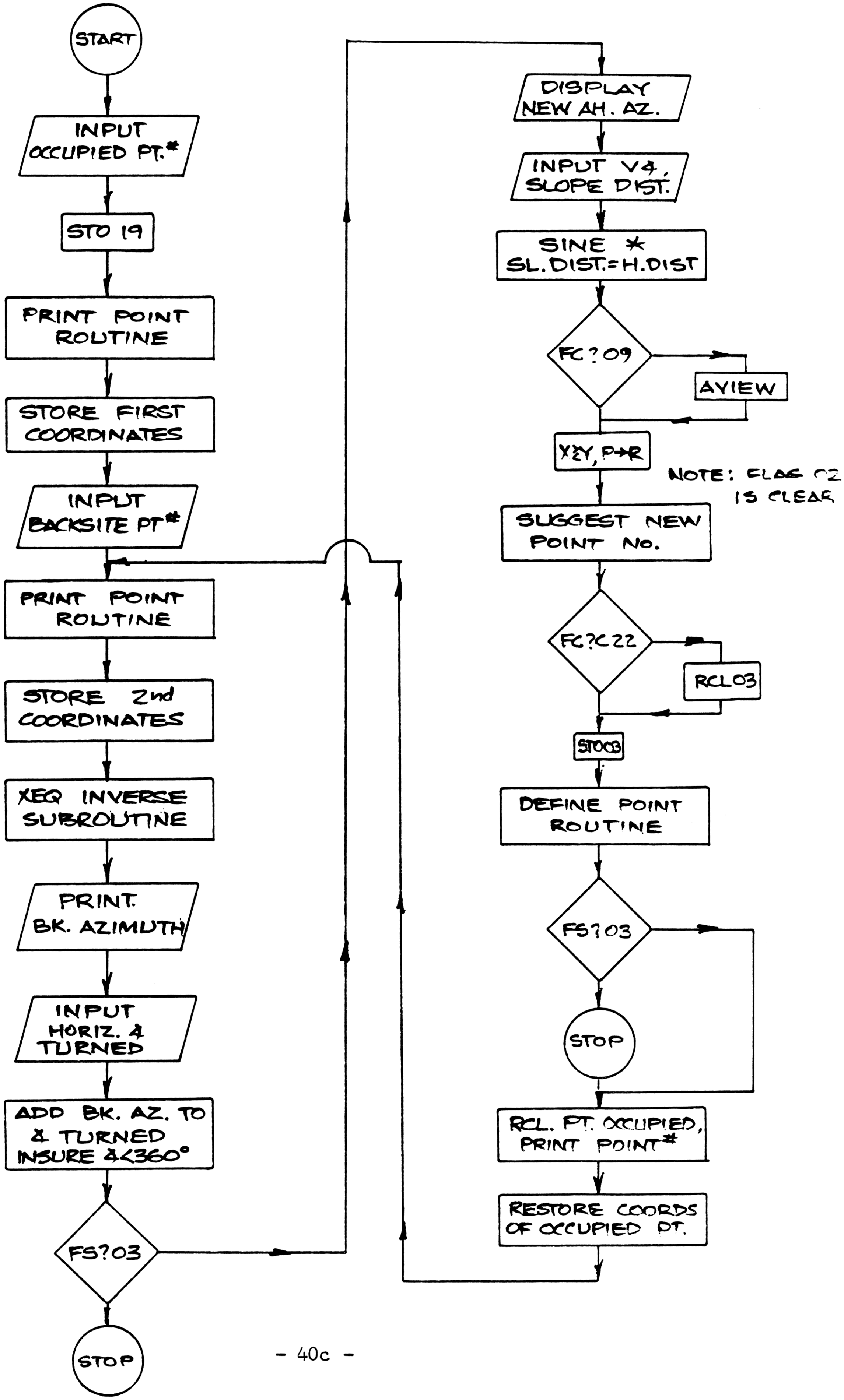
INVERSE



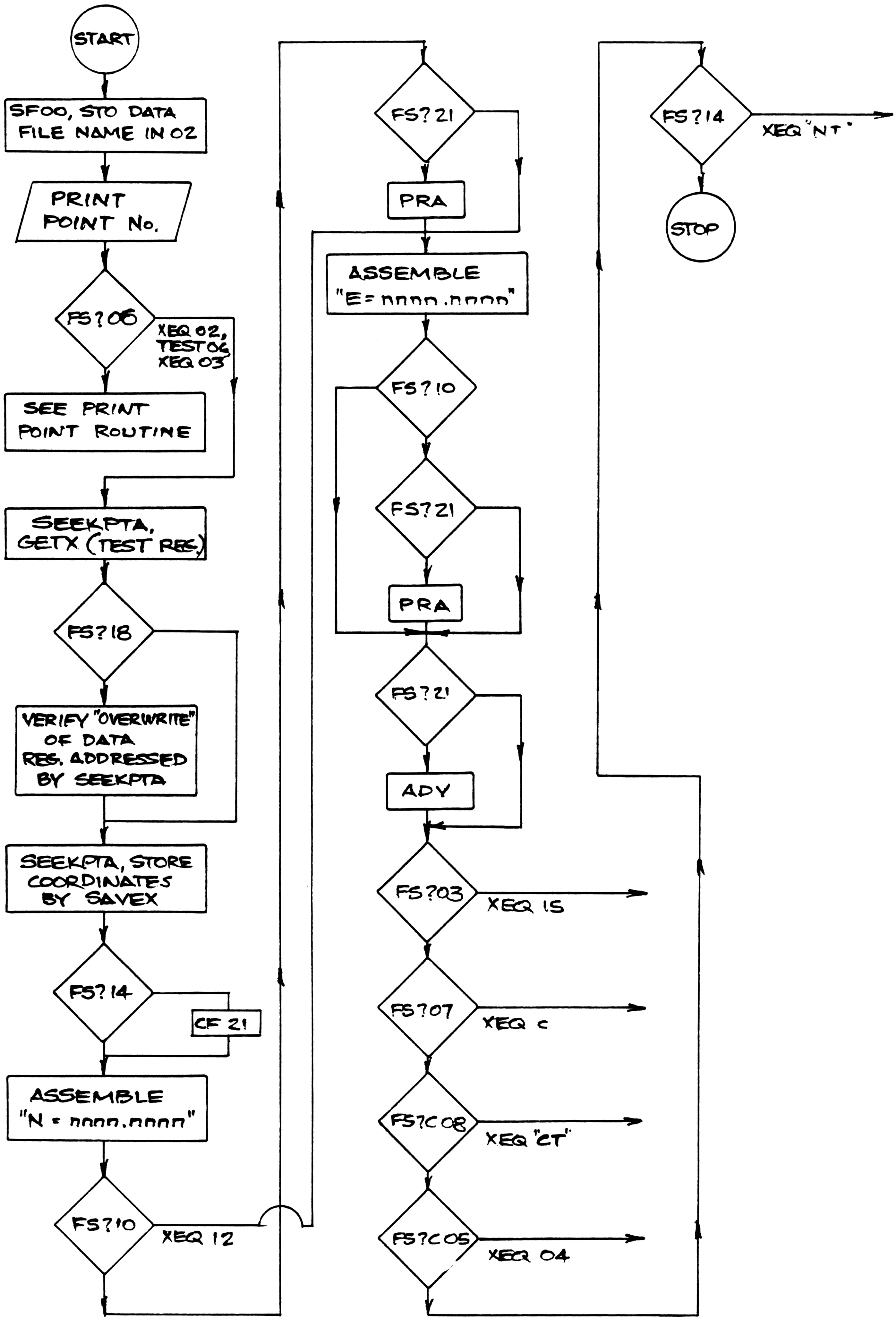
COMPASS ADJ.



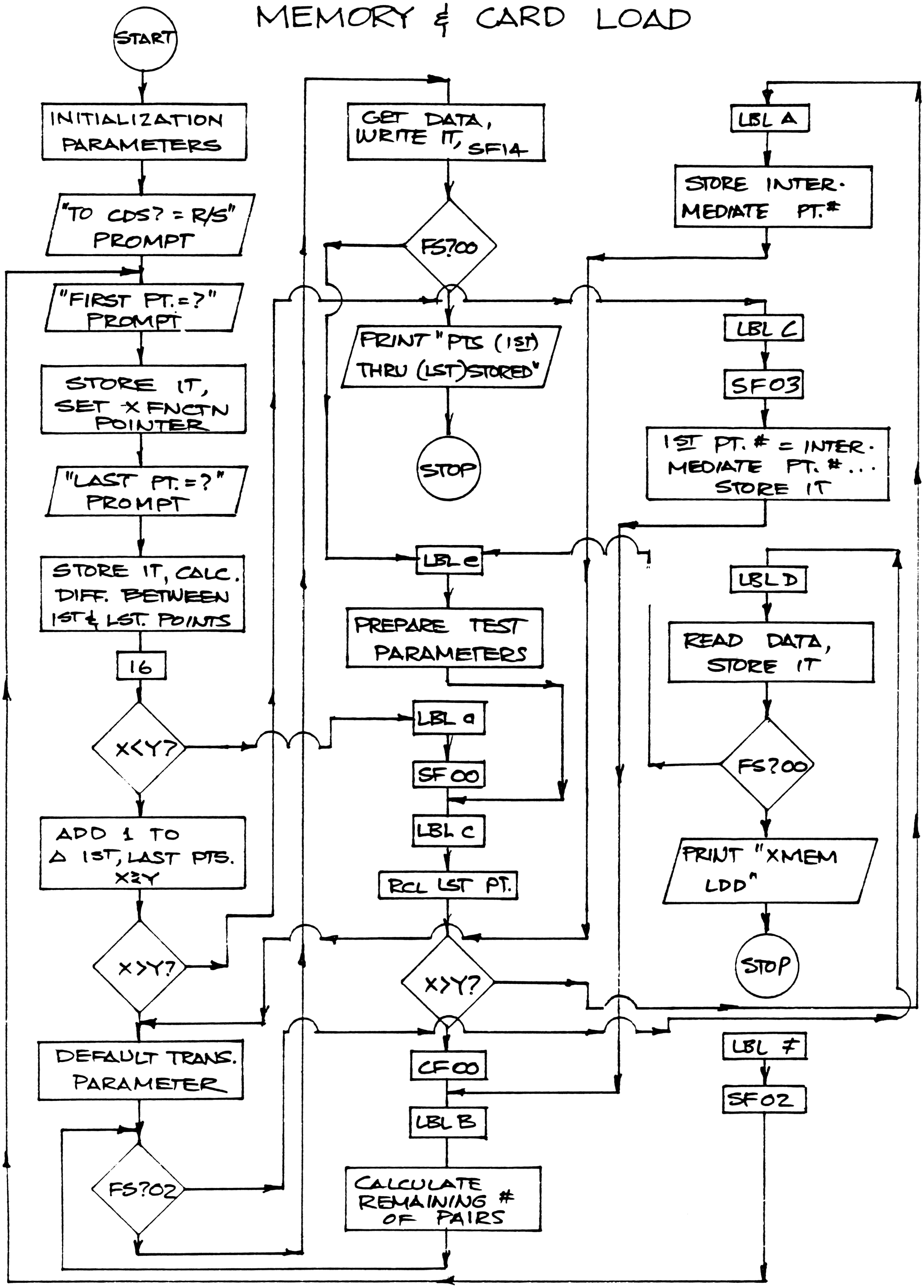
SIDESHOTS

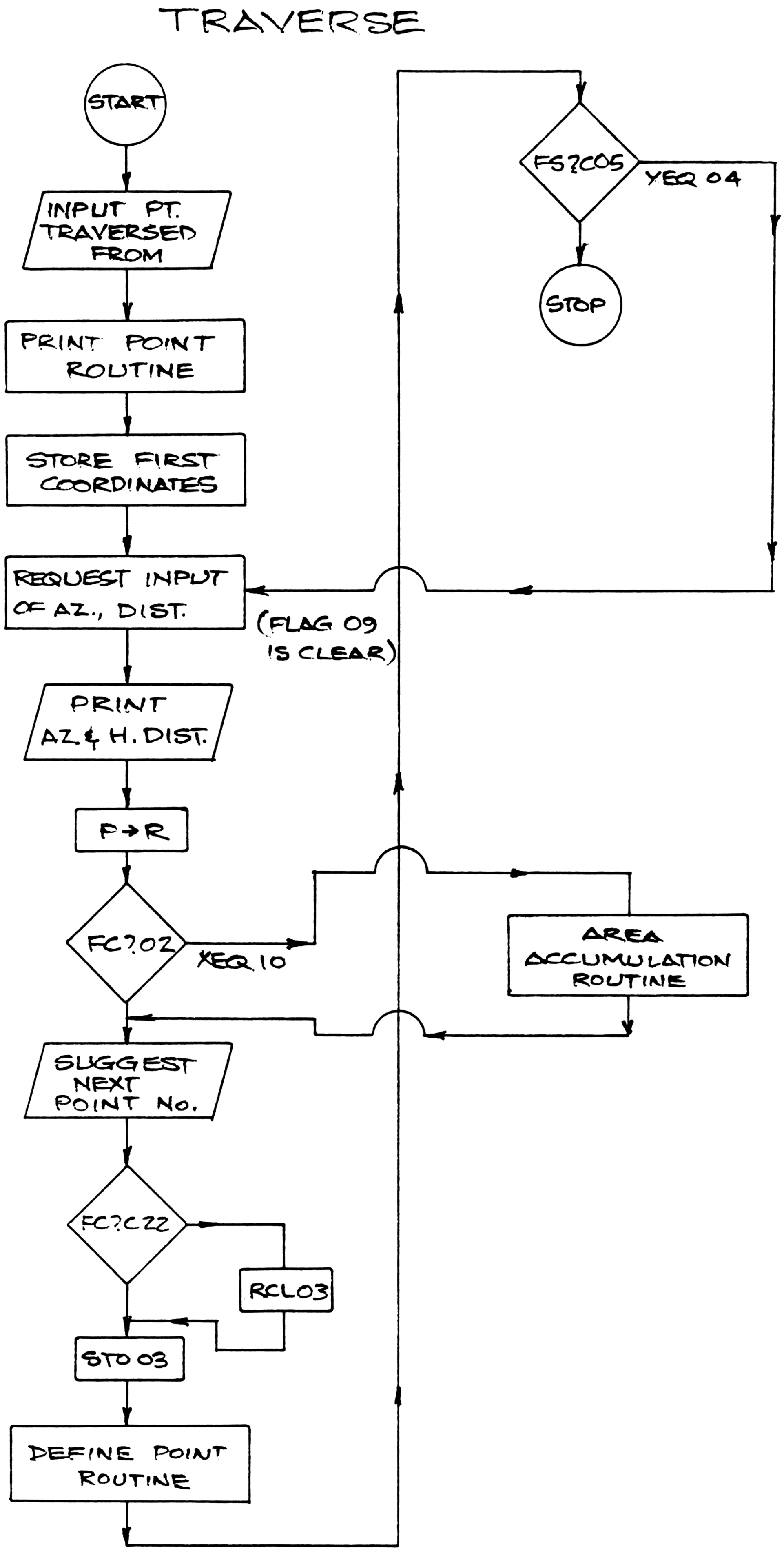

START
INPUT OCCUPIED PT.#
STO 19
PRINT POINT ROUTINE
STORE FIRST COORDINATES
INPUT BACKSITE PT#
PRINT POINT ROUTINE
STORE 2nd COORDINATES
XEQ INVERSE SUBROUTINE
PRINT. BK. AZIMUTH
INPUT HORIZ. ∢ TURNED
ADD BK. AZ. TO ∢ TURNED INSURE ∢<360°
FS?03
STOP
DISPLAY NEW AH. AZ.
INPUT V∢, SLOPE DIST.
SINE * SL. DIST.=H. DIST
FC?09
AVIEW
X≷Y, P→R
NOTE: FLAG 02 IS CLEAR
SUGGEST NEW POINT No.
FC?C22
RCL03
STO03
DEFINE POINT ROUTINE
FS?03
STOP
RCL. PT. OCCUPIED, PRINT POINT#
RESTORE COORDS OF OCCUPIED PT.

DEFINE POINT

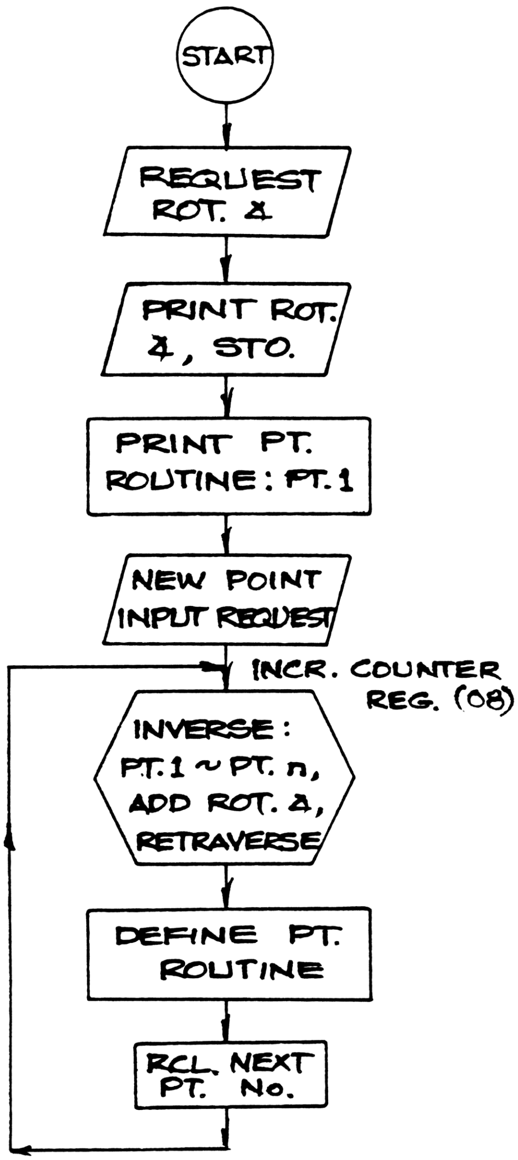


MEMORY & CARD LOAD

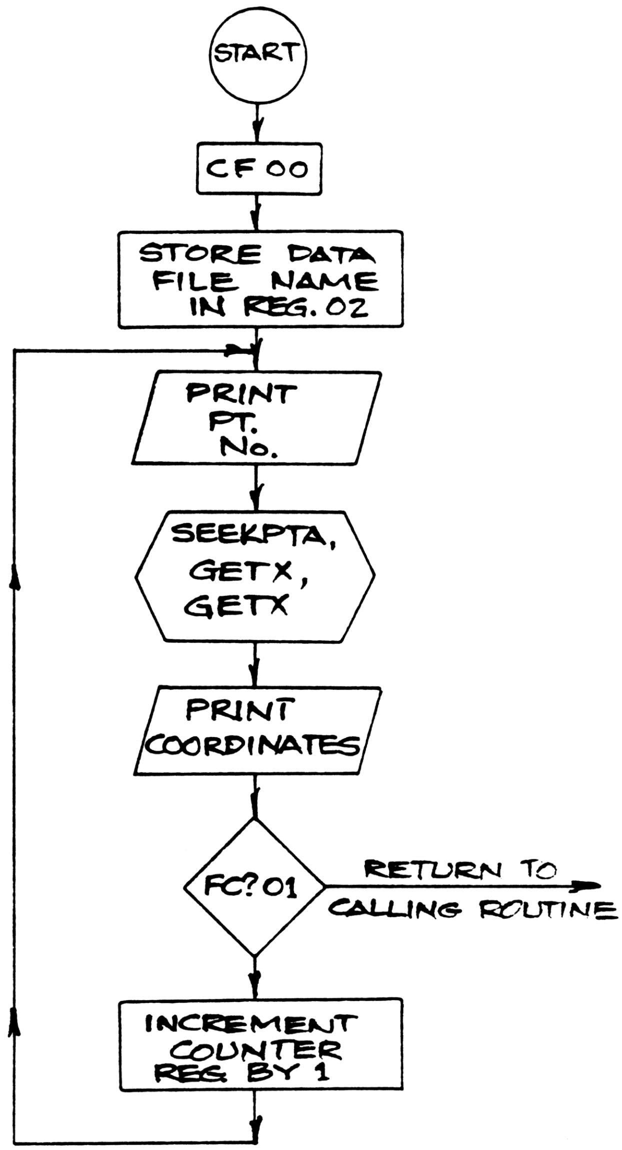



TRAVERSE
START
INPUT PT. TRAVERSED FROM
PRINT POINT ROUTINE
STORE FIRST COORDINATES
REQUEST INPUT OF AZ., DIST.
(FLAG 09 IS CLEAR)
PRINT AZ & H. DIST.
P→R
FC?02
XEQ 10
AREA ACCUMULATION ROUTINE
SUGGEST NEXT POINT No.
FC?C22
RCL03
STO 03
DEFINE POINT ROUTINE
FS?C05
XEQ 04
STOP

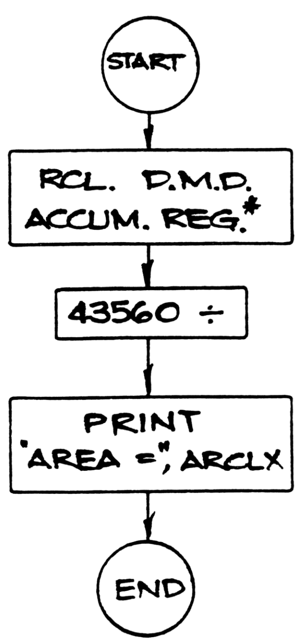
ROTATION



PRINT PT. ROUTINE

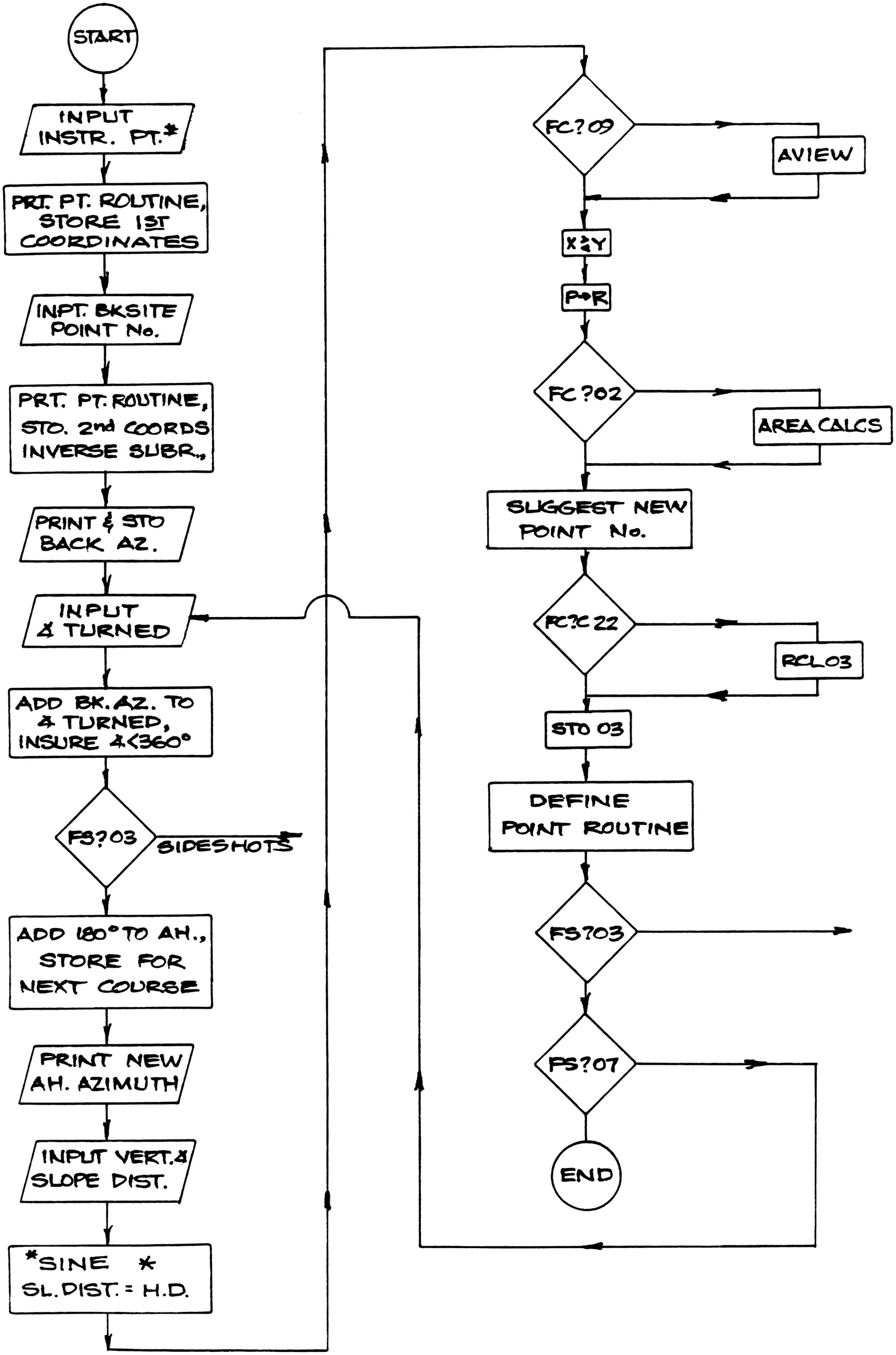


AREA



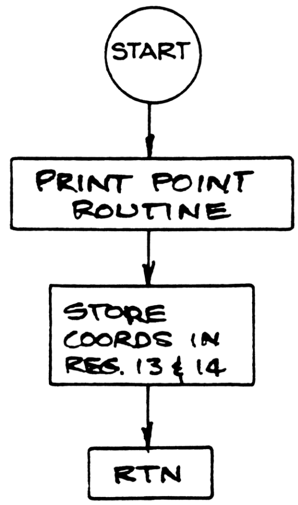
*AREA WAS ACTUALLY CALCULATED IN EITHER TRAVERSE OR INVERSE ROUTINE.

TRAVERSE BY FLD. DATA INPUT

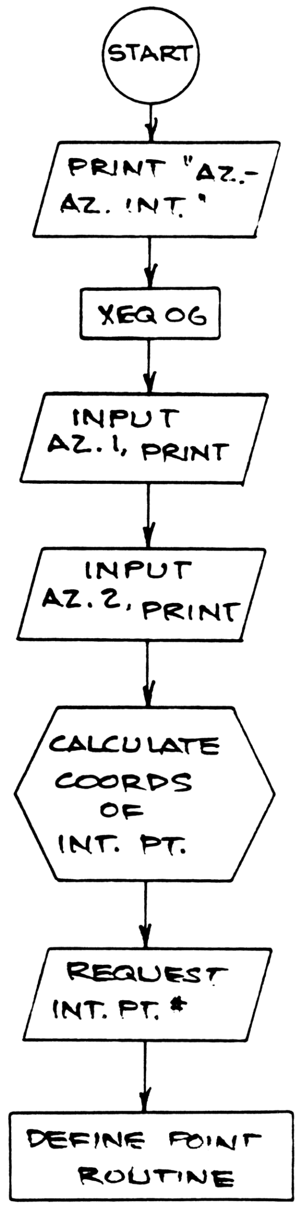


* DELETE & REPLACE W/COS. IF GUN USED CALLS HORIZ.

INTERSECTION SETUP



AZ.-AZ. INT.



Appendix C

ERROR MESSAGES

The following is a list of possible explanations for applicable error messages:

ALPHA DATA: Attempt was made by the HP41C to perform a numeric operation upon non-numeric data. This would normally occur upon program changes, other resident programs or ROM interaction with register 02, which contains the X function working file name, "SP"

DATA ERROR: Refer to Appendix E of your owners manual.

NONEXISTENT: Normally occurs at attempted execution or making use of a nonexistent (or defaulted) peripheral, such as the printer or X function module. Check Flags 10 and 21, if a printer is not enabled.

CHECKSUM ERROR: Usually caused by attempted loading of dirty card (or a card recorded following program changes or additions, without a proper "packing") into the card reader.

CARD ERR: Usually caused by mixing types (program and data) of multiple track cards during the card reading process.

PACKING AND TRY AGAIN: Normally caused by an insufficient allocation of program memory, and will be displayed either during the loading or amendment of a program. Note: while you may be able to load the TR program into the HP41C, the programmed size statement (020) may cause this display if another resident program is too large to allow this additional program.

FL NOT FOUND: This will be displayed if either the working file has not been created (page 5) or the contents of register 02 have been changed within the define or print point routines.

PRINTER OFF: No explanation required for this message: either enable this peripheral or unplug it to continue program execution. In most instances a R/S will allow proper program continuance, after the printer is enabled.

Appendix D

LABELS

The table below lists the local, Alpha and Global labels used in these programs: You will need to keep aware of them if you alter the programs or add new ones. A conflict may arise, for example, if you duplicate one of the labels listed below in your own program. Don't be afraid to experiment: We don't claim our program is perfect, but it serves us well.

PROGRAM	ALPHA	LOCAL	GLOBAL
TR	I,A,b,J,B,c	01,02,03,04, 05,06,10, 11,12,13,14, 15,16,17,18, 19,20,23	TR,DE,PR,IN,AB
IS	F,Σ,%	07	IS
CO	G	08	CO,CT,NT
*ML	A,B,C		ML
*CL	a,b,c		CL

* No conflict should arise here, because all other resident programs should have been cleared prior to execution.

Appendix E

USER TIPS

A. If you know you are going to be using CL and have room for it upon listing CAT 1. This way you can clear all following programs by using the PCLPS (from the keyboard) in one fell swoop. When you are done calculating simply go to the first label following the CL program, as listed by CAT 1; execute PCLPSa, and all the programs following it will be gone. This saves you time in clearing out those programs individually.

B. With a quad RAM in a HP41C, you will be able to generate 300 coordinate pairs, but you will not be able to print or transfer them to permanent storage. This is because all four ports are being used and the quad RAM is required for data transfer. If this is your situation, you can modify the TR program by deleting everything except the print point routine (labelled PR) and pulling out the quad RAM module, vacating one port, facilitating at least a print out of all the coordinates. Everything following label TW can be deleted without effecting the print point routine. Here's how:

1. Insure all other programs are cleared out of the HP41C except TR.

2. GTO .049

3. Program mode ON: Insert phony label "AC"; press label, alpha, AC, alpha.

4. Backstep to line 049.

5. Insert "END" statement: execute alpha, END, alpha.

6. Program mode off.

7. Execute alpha, CLP, alpha, alpha, AC, alpha.

Now, by setting flag 01, you can begin printing the point numbers and coordinates out in the normal manner.

C. Label c could include multiple angle tests against rejection limits. This could be handy if you want to double check your field angle rejection limits.

D. A short routine using "IN" could give you angle rights or angle lefts and horizontal distances to set monuments. "IN" is the inverse subroutine at line 372 of TR.

E. It is possible to do a quick swap of extended function/memory module between two HP41Cs. You should pay attention to the order of installation discussed in the owners manual. Normally you will have between 20 seconds and 2 minutes to perform this exchange.

F Should you wish to print out the newly adjusted coordinate pairs as they are being generated during the compass adjust routine, delete the following lines from program memory:

077 FS? 10

078 XEQ 12

085 FS? 10

086 XEQ 23

Bear in mind however, that this step will "force the calculator to print out newly defined coordinate values, in all routines.