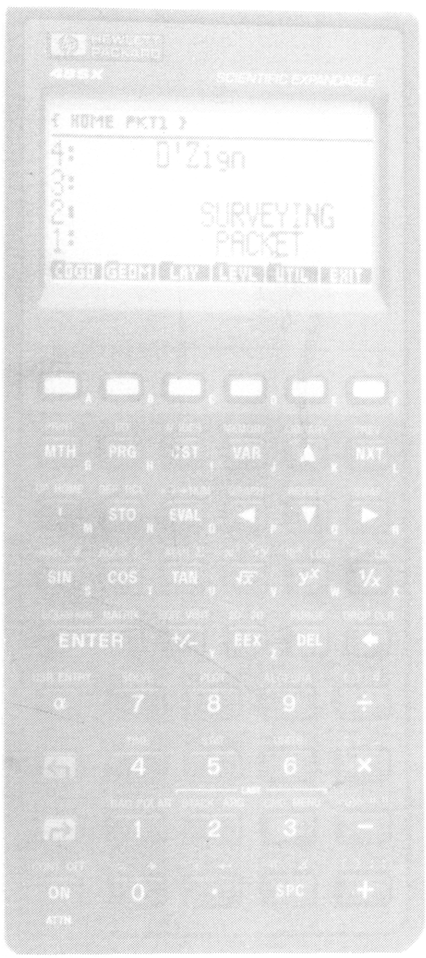
HP48SX SURVEYING PACKET

1

VERSION 3.0



Owner's Manual

HP48SX SURVEYING PACKET

1

VERSION 3.0

Owner's Manual

Library of Congress Catalog Card No. 90-86308
ISBN 0-944889-15-8

Manufactured in the United States of America

1st printing December 1990 **Ver 1.0**
2nd printing August 1991 **Ver 2.0**
3rd printing January 1992 **Ver 3.0**

published by D'Zign
Fresno, California, U. S. A. 1990

ISBN 0-944889-18-2

TECHNICAL ASSISTANCE

The program material, instructions and procedures contained in this book assume that the user has a working knowledge of both surveying and the general operation of the Hewlett-Packard 48SX calculator.

Technical assistance is limited to verification of the results shown in the various examples used in the Owner's Manual for this product, and is available to those with registered manual numbers.

User Support staff are available between the hours of 3:00 A.M. and 12:00 Noon (pacific time), Monday through Friday. For assistance, call (209) 276-3460.

D'Zign

P. O. Box 9790

Fresno, CA 93794 - 9790

TABLE OF CONTENTS
and
LIBRARY GUIDE

The following guide shows the contents and location, by section and page number, of the programs in each of the libraries of Version 3.0 of Surveying Packet #1.

-------- LIBRARY 787 --------

There are a number of things to be aware of, before you begin to use the programs contained in this surveying packet. Please resist the urge to install the programs and start traversing, and read this introduction.

Your reaction to the 48 will depend on what calculator you've been using. If you have been using a 41 with a survey pac, some of what the 48 can do will seem just short of fantastic . . . on the other hand, if you've been using some of our HP42S software, you'll find yourself thinking that the 48 is a little slow.

When we began writing the programs for this survey packet, we found that there are two things that can work very slowly in the 48. Graphics and displays. We found a way to make the displays react in a reasonable manner, and got around the graphics speed by not using graphics.

Most field surveyors don't have the time to stand around and watch their calculator draw pictures, and you can (literally) do that if you try to plot out your subdivision or topo on the screen.

If this is the book that you received with your copy of Surveying Packet #1, you have version 1.00. This, as with all programming, hints at newer, or better, versions in the future. This doesn't mean that you bought your copy too soon, because UPDATES are FREE. All you have to do is fill out and return the registration card.

product upgrade plan

Your D'Zign software has been designed to give you flexibility in your calculations as well as rapid solutions to your field problems. We are continually attempting to improve our products to make them better and easier to use.

Your input is also needed. We need to know what our programming **doesn't** do that you may need in your particular type of work.

how it works

As part of the upgrade plan, free copies of each new version will be sent out as it is produced. The registered purchasers of Surveying Packet #1 will receive a new program disk and any necessary addendum pages. Registered manual owners will receive the addendum pages, and can upgrade their programming through the original source of their program. This manual has been done in sections, to allow for future additions.

user support

We want you to get maximum performance from your D'Zign product. If you have questions or if you should encounter any difficulties, we will be glad to help! Most of the time you will find the answers right in your Owner's Manual, so check there first. If you still have questions, call for help. Our Support Staff is available at (209) 276-3460 between 3:00 A.M. and 12:00 noon PACIFIC time.

product replacement plan

In spite of our rigorous testing and high quality control standards, even D'Zign Products sometimes need replacement. In the event your disk proves to be defective, it will be replaced at no charge within 90 days of purchase or for a $20.00 service fee thereafter.

If you think you have a defective product, call our Support Line. Upon confirmation that a problem exists, mail the defective product, your proof of purchase and a description of the problem you are experiencing **to D'Zign**. A product returned without proof of purchase is not eligible for warranty replacement.

NOTICE

Note: additional manuals are $40 each

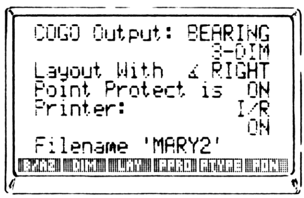
READ THE REMAINDER OF THIS SECTION BEFORE CONTINUING

parameters

The output generated by your programs depends on the settings you have extablished, and the settings are made by using the PARAMETER file. When your main menu,

is showing, stroke NXT to access the program.

A screen like the one shown to the right will appear, and each of the settings can be changed by using the softkey menu.



The **current** filename is displayed at the bottom of the screen first, followed by the rest of the **current** parameters.

The first 2 lines of the display set the output for programs like Traverse, Intersection, etc. The first line options are **bearing** output or **azimuth** output. Change the output by stroking

Line 2 has options for working in 2-, 3- or 4-dimensional coordinate mode. Three-dimensional traversing, as an example, will require input of an H.I. elevation each course; if you aren't actually in the field doing the traverse, the 2-DIM setting would be more appropriate. If the stored coordinates include an elevation, it will be displayed as part of the output anyhow. Toggle the second line with

point protection

There are always mixed feelings about point protection. If you are using COGO to just generate some needed coordinates and don't need intermediate results, you can just reuse 'throwaway' point numbers, and "used point" prompts are a pain in the neck.

On the other hand, if you accidentally overwrite a point in a serious file of coordinates you'll wish you had point protection. Because most of us work both ways, we've set it up so that you may turn it on or off. Toggle by stroking

The output of the layout programs may be either azimuth or angle-right. Select or change that parameter with LAY

printer parameters

The printer **type** options allow for using either the InfraRed or a serial printer, and further, either a 40 or 80 character output for the serial printer.

If you are going to use a serial printer you should read pages 610 and 611, of the HP48 Owner's Manual, as well as the section on setup (page 617) of the I/O parameters. The necessary flag settings in the HP48 are automatically changed when you select the printer type with PTYPE

Whether or not the printer is being used affects how some of the programs work, so that is also selectable, using PON

additional parameters

Stroking NXT CONT will bring up additional softkeys and their current settings.

The CONV key sets the output parameter for the programs, CONVERSIONS and ACCUMULATE.

South azimuth is set with the AZM key. Default is north azimuth, but once set to south azimuth, the programs will all use south azimuth until re-set.

code

This is a timesaving function included in several of the functions, and is also used to define a 'block' of coordinates. The form used by Ver. 3.0 is **1st number** SPC **2nd number**. When the input calls for more than one number, the items are always separated by spaces (SPC).

exiting

It is extremely important that you **EXIT** each program when you have finished with it. In many cases the keystroke, EXIT, will be in the menu prompt bar, but if it isn't, stroking VAR brings up a menu that contains the softkey.

installation

Before you start to transfer this packet to your HP48SX you should read chapter 33 of the Owner's Manuals for the calculator. It begins on page 612, in Volume II.

Your calculator should contain a 128K RAM card, with the memory **merged** to main memory. Because it is merged, the programming is installed into PORT 0, regardless of which port the RAM card is in.

kermit

Kermit protocol was developed at Columbia University Center for Computing Activities. Contrary to common belief, the Kermit programs are not necesarily in the public domain. The version included on the diskette furnished with your surveying packet is actually copyrighted by Hewlett-Packard, as part of their Serial Interface Kit, and reproduced with their permission.

Kermit programs are provided "as-is", with no warranty of any kind. Columbia University, the individual programmers, and the contributing institutions make no claim as to their correct operation or the accuracy of the documentation . . . Kermit is not a commercial venture.

port

Our instructions tell you to SET PORT 1. If you have accessories on your PC (mouse?) the correct port for use <u>may</u> be PORT 2 instead.

file transfer

Additional programming is included on your diskette for transforming files from the PC format to the HP48 format we use, and back again. This software is also provided "as-is" with no express or implied warranties.

getting started

One of the best 'first moves' is to make a backup copy of the diskette, and print out the text files, which contain the specific instructions for each of the programs. D'Zign's License Agreement with you allows you to make one copy, for backup purposes, and it must show our copyright on its label. Once you are sure you understand the process, you are ready to hook your calculator up to the PC and down-load the required library (or libraries) to your HP48SX, as follows:

The first thing to do is to match the settings on the 48 and PC. They should look something like the information shown to the right, and note that all of the **programming** is downloaded from the PC in **binary** code.

```
    I/O setup menu
IR/wire:        wire
ASCII/binary:   binary
baud:           9600
parity:         none 0
checksum type:  3
translate code:3
```

On the calculator, you would stroke ⬑ PRG to bring up the I/O menu bar. Stroke the ▭ key under SETUP to set the parameters for the transfer. To leave the setup area after you have set the parameters, stroke ⬑ PRG again.

Regardless of which optional libraries you may want to add, the first library to down-load is **PKT1L003**, which has a Library Number of **787**. For other libraries, follow the same steps exactly, with the exception that you add the letter suffix to the name where the † is.

on the computer

Initiate Kermit by typing

KERMIT ↵

1. **prompt:** KERMIT-MS>
   keystrokes: SET PORT 1 ↵
2. **prompt:** KERMIT-MS>
   keystrokes: SET BAUD 9600 ↵
3. **prompt:** KERMIT-MS>
   keystrokes: SEND PKT1L003† ↵

At the same time (approximately) that you stroke the last key on the computer, stroke the ▭ key under RECV in the 48's menu bar.

4. When the computer tells you that the transfer is complete, stroke [VAR] and you should see PKT1 in the menu bar. Stroke the ['] key, then the [ ] key under PKT1 and then stroke [→] [STO]. This should recall the library to Level 1 in the display.

Again stroke ['] and the [ ] key under PKT1. Now purge it by stroking [←] [DEL]. Next, follow the keystrokes below:

[→] [+] [0] [▶] [7] [8] [7]
[ENTER] [STO] [→] [ON]

Turn the calculator back on, and it should sort of 'blink'. Do these keystrokes:
[7] [8] [7] [←] [VAR]

Stroke [NXT] and stroke the [ ] key under ATTAC.

Now, select your options. The table of contents indicates which programs are in each of the optional libraries, and the programs which you opt to include require that you down-load the library that includes any program that you want to have in your calculator.

If you haven't left kermit on the PC, it isn't necessary to reset the baud rate or port, so starting at step 3, repeat the procedure with the next library. Substitute the correct Library Number in place of **787** when you attach the library. Repeat steps 3 and 4 until all of your options are loaded and attached to the home directory.

With that done, the directory can be down-loaded. Select the PKT1 directory with the correct suffix for the libraries you have installed from the chart below.

LIBRARIES USED	USE	LIBRARIES USED	USE
787	PKT1A	787 +792 +793	PKT1E
787 +791	PKT1B	787 +791 +793	PKT1F
787 +791 +792	PKT1C	787 +792	PKT1G
787 +791 +792 +793	PKT1D	787 +793	PKT1H

on the computer

prompt: KERMIT-MS>

keystrokes:

SEND PKT1 † [↵]

on the HP48SX

Stroke the ▭ key under RECV. This transfer goes pretty fast, and when it is completed you have finsished the downloading.

Stroke the VAR key to return to the main menu bar and you should see PKT1. Stroke the ▭ key under it to go into the directory (your top path line should now read "HOME PKT1†") and your menu line should show

COGO GEOM LAY LEVL UTIL EXIT

interfacing

Files that are generated by the programs in Version 3.0 may be down-loaded to your PC and transformed with the (included) program, **482PC.EXE**, after which they are in the form of an ASCII file, compatible with most PC software used for surveying. Your PC software, in turn, probably contains a program for transforming the ASCII file to DXF.

file transfer

To transfer a file from the 48 to the PC, your I/O parameters should be set to ASCII instead of binary.

With the filename in Level 1, use kermit to send the file to the PC. If you were to view the file, it would appear on your screen in the form shown to the right.

```
%%HP: T(3)A(D)F(.);
{ { 16 } {
[ 5030.0668 5036.48875 274.96 ]
"0+00 HEIDI DR" } {
[ 4360.2268 5030.44623 273.44 ]
"6+80 HEIDI DR" } {
[ 4360.2256 5052.3449 277 ]
"H2O METER" } {
[ 4360 4579.414 0 ]
"45 ELL" } {
[ 4392.9492 4677.9992 277.57 ]
"COR BB COURT GB" }
{
[ 4182.9508 4677.9992 276.55 ]
"COR BB COURT GB" }
{
```

transforming

Execute the program, **482PC.EXE**, for the screen prompt:

```
THIS PROGRAM IS FOR CONVERTING D'Zign COORDINATE FILES TO THE ASCII FORMAT.

SOURCE FILE NAME MUST INCLUDE DRIVE, DIRECTORY, FILE NAME AND EXT.
SOURCE FILE NAME
```

If all of your work is being done in the same directory, the drive and directory need not be input. Type in the filename and stroke ↵

```
OUTPUT FILE WILL BE STORED IN THE CURRENT UNLESS A PATH IS ENTERED .
DO NOT ADD AN EXTENTION, THE EXTENTION .ASC, WILL BE ADDED BY THE PROGRAM.
OUTPUT FILE NAME
```

Respond to this prompt by input of a filename for the output ASCII file and stroke

The file has been transformed to a PC ASCII file, and now, if viewed, looks like

```
1,5030.0668,5036.48875,274.96,0+00 HEIDI DR
2,4360.2268,5030.44623,273.44,6+80 HEIDI DR
3,4360.2256,5052.3449,277,H20 METER
4,4360,4579.414,0,45 ELL
5,4392.9492,4677.9992,277.57,COR BB COURT GB
```

Going the other way (ASCII file to HP48 form), use **PC248.EXE** for the transformation. This program will also prompt for the SIZE of the HP file.

Before transfer of the new file to the 48, use the Version 3.0 file system to create a new file with the filename you are going to use for the file. This is not necessary if the filename already exists in the HP48.

To convert an ASCII file in the computer to the form used by the programming in the 48, execute the program, **PC248.EXE.**

```
THIS PROGRAM IS FOR CONVERTING ASCII COORDINATE FILES WITH THE FORMAT
POINT NUMBER, NORTHING, EASTING, ELEVATION, DESCRIPTION, TO THE HP
48 SX FORMAT, COMPATABLE WITH D'Zign's PKT1 DATA FILES. SOURCE FILES
MAY BE EITHER COMMA OR SPACE DELIMITED.

SOURCE FILE NAME MUST INCLUDE DRIVE, DIRECTORY, FILE NAME AND EXT.
SOURCE FILE NAME
```

Again, if all of the work is being done in the same directory, it is not necessary to input the directory or drive. Input the filename, including the extention, if there is one. Stroke

```
HP FILE SIZE
```

Input the size of the file, stroke

```
OUTPUT FILE WILL BE STORED IN THE CURRENT DIRECTORY UNLESS A PATH
IS ENTERED. DO NOT ADD AN EXTENTION TO THE FILE NAME.
OUTPUT FILE NAME
```

Type in the 48 filename that matches the filename in the 48 file system and stroke

↵

Transfer the file to the 48, with kermit. When the transfer is completed, stroke VAR to return to the main menu. The new filename will appear at the left side of the menu bar (note: because this is an existing file, the transfer will add a 1 to the end of the filename. This is normal.).

Stroke ', then the ▭ key under the menu listing and

ENTER ENTER ↱ STO

Type in the filename (α filename ENTER) and stroke

STO ↰ DEL

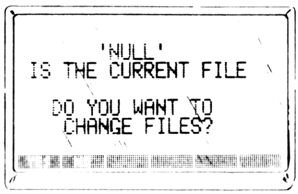
The file is now in the 48 and may be used with any of the programming.

general note:

The above system may be used for any of the data files containing coordinates of topo pickup (DATA). Storage files used with the Levels and/or EDM Slope Staking programs use a different system.

Leveling or Slope Staking files are designed to be output to a printer for permanent record. If the printer is OFF and the programs are output, they will single-step through the output so that the results may be written down. When the printer is ON, the total file is 'dumped' to the printer continuously.

All of the programs in the **CO**ordinate **G**e**O**metry directory store coordinates into a file. The file into which they are stored is selected by the user, and is the **current** file.



When you first enter the COGO directory, the screen display will remind you which file you are using, as shown to the left.

Note that the current file is a file named 'NULL'. We have included this file with the programming so that there is a file to begin with. It is sized at 100.
WARNING: never purge all of the files at the same time.

coordinate storage

Because different programs have different requirements, input of the coordinates varies slightly from program to program. More specifically, when you use the option, **NEW**, to answer the **POINT NUMBER?** prompt.

This prompt expects a point number and then selection of either **EXIST** or **NEW** in response. Where appropriate, a third option, [illegible] is also given.

POINT NUMBER?

Input the point number you want to use, and then stroke the key corresponding to the option:

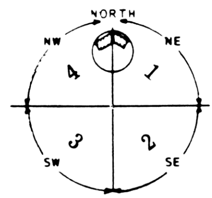
If the point number is that of coordinates that are already stored, stroke **EXIST**

If the coordinates have not yet been stored, stroke **NEW**.
One of the input menus will be shown:
1 | | | | | **CONT** **2** **PT#** | **N** | **E** | **ELEV** | | **CONT**
For the first type, input the north coordinate, stroke **SPC** (or **ENTER**), input the east coordinate and stroke **CONT**

The second type of menu requires that you **stroke** **PT#**, **and** use the **N** and **E** keys to input the coordinates. If you want to input the elevation, input the value and then stroke **ELEV**. You may also enter a descriptor at this point: stroke ↱ **−** **α**, type in the description, and stroke **ENTER**. When input is completed, stroke **CONT**

These programs do not make any distinction between "bearing-azimuth" and "field angle" traversing. Once you are in the TRAVERSE program, you may use any method of traversing which you want to use.

Bearings are input with quadrant codes and the quadrants are numbered with the same system that has been used by Hewlett-Packard since the first surveying programs for handheld HP's came out.



quadrant codes

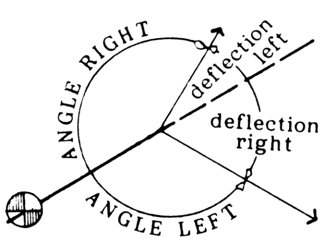
The bearing and quadrant code are entered, separated by either SPC or ENTER, and then a BRNG key is stroked.

And, some CAUTIONS: These programs do not recognize sideshots as different from traverse shots. If you use the angle-adjustment, or either the compass or transit adjustment programs, you will get incorrect answers if the traverse contains sideshots. There are ways to keep this from being a problem.

1. Use point numbers for the sideshots that are high enough to be outside the point numbers used for the traverse itself. If you are doing a traverse that will use point numbers 1 through 7, use numbers 11 and up for the sideshots.

2. Calculate the traverse without the sideshots, adjust it, and then do an INVERSE traverse from point to point to point, setting the sideshots as you go. This is quick and simple, because the coordinates are stored by point number in all of these programs.

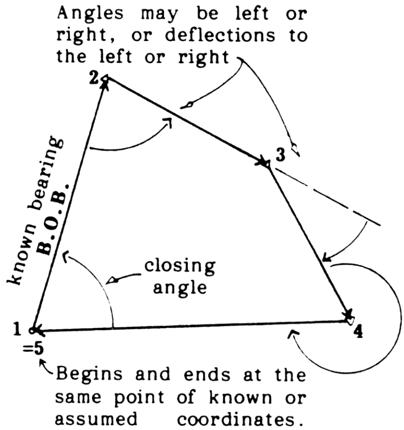
conventions



You may traverse from one point to another by input of the bearing and quadrant code (or azimuth), or you may turn an angle to the next point.

The illustration to the left shows the relationship of these angles, referenced to the backsight and foresight.

Traverses may be thought of as either "**closed**" or "**open**" traverses. For use with this program, the **CLOSED TRAVERSE** may be either of two types. What we will call **Type A** is one similar to the one shown to the right.

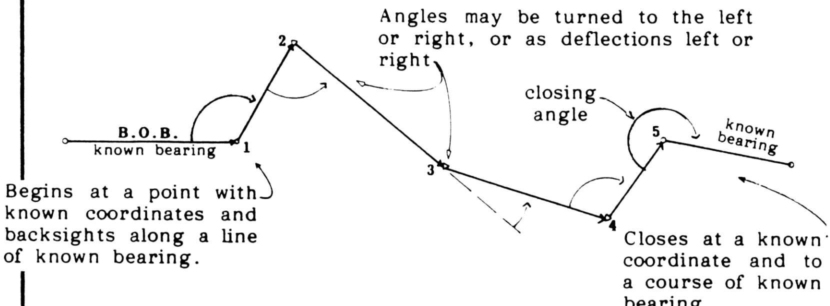


In this type of traverse, the line from **1** to **2** is usually a known line which is included in the traverse.

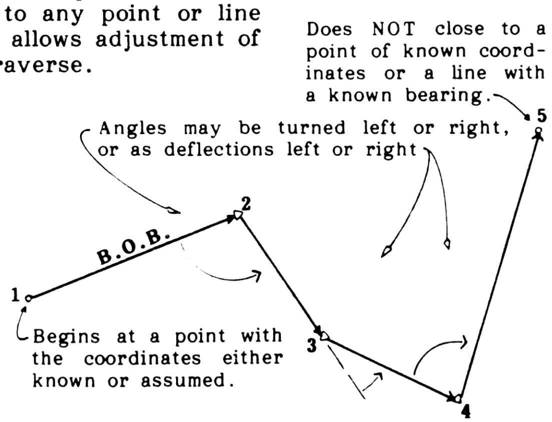
The two points used would be part of a property or monument line, and the basis of bearings would be the bearing of the line.

This type of traverse also closes back to the original point of beginning, and allows the turning of a closing angle, which is turned at the first (and last) point, foresighting the second point.

What we will consider to be a **Type B** closed traverse is one which begins at one known point and ends at another known point. For this type (below) the basis of bearings is usually obtained by backsighting another known point.



An **OPEN TRAVERSE** is one which, while it may begin at a known point, does not close to any point or line which allows adjustment of the traverse.



An OPEN traverse may also be considered as being an 'unfinished' traverse, in that it could later be used as a portion of a traverse which will be closed.

More often, a traverse of this type is run as part of a topographic survey, where the traverse is considered accurate enough without correction.

For the CLOSED type of traverse, the angular error is usually distributed equally among the angles, prior to adjustment of the traverse.

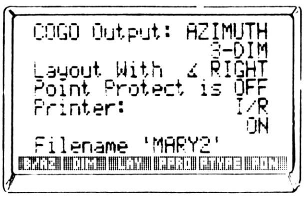
If the surveyor wanted to distribute some of the angular error to specific points (i.e.: short backsight, tall grass, tree limbs, etc) it could be done at this time, and the traverse then calculated again. The remaining angular error can then be taken out automatically with the angle adjustment routine.

After angle adjustment, the final adjustment may be made with either COMPASS or TRANSIT method, both of which are included in the programming.

Note: The instructions which follow are for traversing in 2-Dimensional coordinates. Instructions for traversing with the parameters set for 3-DIM begin on page 3-45.

selecting the output

Before you enter the traverse program, use [illegible] to select the output, either bearings or azimuths (see pages 1-**3** and 1-**4**) toggle with [illegible].



In addition to the selection of azimuth or bearing, check the setting on DIM. Unless you are in the field, running a traverse in 3-dimensions, the best setting is 2-DIM.

When the existing coordinates are stored as three dimensional, the elevation is displayed when the coordinates are output, regardless of the current DIM setting.

This program uses two 'traversing' menus:
BRNG AZIM CODE INV CLOSE EXIT & H.DIS S.DIS SIDE CURVE ∠ RT. DEF∠
If the next function you want to use is not in the menu that is currently displayed, the alternate menu will be displayed if you stroke NXT .

starting coordinates

With the correct file selected and the output type chosen, the calculator is ready for input of a starting point, and prompts

POINT NUMBER?
EXIST NEW RE #

Input the point number you want to use, and then stroke the key corresponding to the option:

If the point number is that of an already stored pair of coordinates, stroke EXIST

If the coordinates have not yet been stored, stroke NEW

INPUT COORDINATES
CONT

Input the north coordinate and stroke SPC

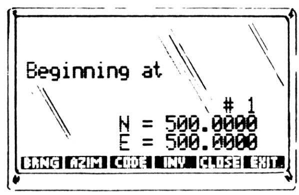
Input the east coordinate and stroke CONT

3. If this is an existing pair of coordinates, but you want to duplicate them under a new point number for this traverse, stroke [illegible]

NEW POINT NUMBER?

3a. Input the new point number you want to assign, and stroke CONT

The output will be similar to the screen shown to the right.



The last pair of coordinates calculated will always appear toward the bottom of the display. If a printer is in use the same information is printed out.

bearing input

Bearings are input in two steps, the bearing angle is entered, then the quadrant code is input.

1. Input the bearing, stroke SPC

2. Input the quadrant code, and stroke BRNG

azimuth input

Courses may be entered as asimuths instead of bearings (if the bearing is northeast, it saves keystrokes).

1. Input the azimuth and stroke AZIM

input by point code*

The bearing or azimuth between any two **stored** coordinate pairs may be called up automatically by using "**CODE**". This 'code' consists of the two point numbers, entered in the form, **AAA** SPC **bbb**, where **AAA** is the first point

* Additional features of the CODE key are described in detail on page 3-**44**. The key has shifted functions for auto-right angle and any angle by code input.

number, and **bbb** is the second.

1. Input the code for the two points, in the form noted above, and stroke

With this method of input the 'whole' bearing is used, rather than a rounded off re-input number.

setting the direction

This initial direction may be the azimuth or bearing toward the point you want to set. That is, it may be the first direction of the traverse.

It may also be a BASIS OF BEARINGS from which an angle or deflection angle is then turned, to set the direction of the first course of the traverse. If the latter, the direction of the basis of bearings should be the direction **toward** the first point.

direction by inverse

You can also establish the direction of the first course by inversing to the next point.

1. Input the point number of the coordinates you wish to inverse to and stroke

You can begin a traverse from existing points (in storage) by inversing from the backsight to the instrument point, to begin the traverse.

The direction of the course and the menu will be displayed, and you can choose your next move.

distance input

Distances may be input as horizontal distance or slope distance, using the appropriate keystroke.

1. Input the horizontal distance, stroke

or

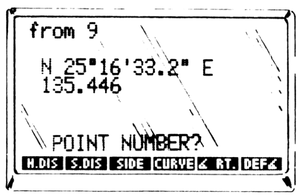
2. Input the slope distance, stroke

ANGLE?

2a. Input the vertical or zenith angle, and then stroke

The horizontal distance is printed out and added to the display.

At this point, the display will be similar to the one shown to the right, including the prompt for the next point number.



Except for inversing, you will have the **POINT NUMBER?** prompt just prior to the output of coordinates, each time. You may assign any point number you want, or just stroke ▶ to get the next consecutive point number.

<u>CAUTION!!</u> If you have used "**CODE**" for a direction, you must input a point number at the **PT NO?** prompt, or the point number used will be one higher than the bbb portion of the code, by default.

angle input

Angles may be turned as field angles or as deflection angles. For an angle left, stroke the +/- key before stroking the ∡ RT. or DEF∡ key.

sideshots

The procedure for a sideshot is the same as for any other course, with the exception that you tell the calculator that you are wanting a sideshot.

1. <u>Signal for a sideshot</u> by stroking

2. Input a deflection angle and stroke

or

3. Input a field angle and stroke

or

4a. Input a bearing, stroke [SPC]

4b. Input the quadrant code, stroke [BRNG]

or

5. Input an azimuth, stroke [AZM]

6. Input a horizontal distance, stroke [HD]

or

7a. Input a slope distance, stroke [SD]

ANGLE?

7b. Input the vertical or zenith angle and then stroke

POINT NUMBER? . . . . .etc.

curved sides

It is easy to include a curved side in the traverse. The Δ angle is input as positive if the curve is to the right, and negative if the curve is to the left.

The sign of the **radius** will determine whether or not the area of the curve segment is included in the traverse area. The radius is input as a POSITIVE number to INCLUDE the area, and as a NEGATIVE number to EXCLUDE the area.

Assume that you are at the beginning of the curve, about to do the curve portion. First, establish the direction of the chord

1. Input Δ/2 and deflect (change sign if the curve is to the left) and stroke [DEF]

or

2a. Input the bearing of the chord, and stroke [SPC]

2b. Input the quadrant code, stroke [BRNG]

or

3. Input the azimuth for the curve's chord, and then stroke

THEN

4. Signal that you are traversing a curve by stroking

CENTRAL ANGLE?

5. If the curve is to the left, Δ must be input as negative (stroke +/-) in order to signal the direction of the curve. The default is a curve to the right.

RADIUS?

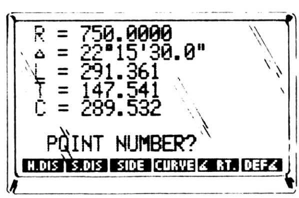
6. Input the radius. If the segment area is to be excluded, stroke +/-. Default radius is positive, and the segment area is automatically included. Stroke

RADIUS POINT
POINT NUMBER?

7. Input the point number you want to assign to the radius point. It should be a number <u>higher</u> than any of the traverse point numbers. Stroke

At this point in the output, the curve data for the curve will be displayed as shown to the right.

Note that the screen also displays the next input prompt, for input of the curve end point number.



POINT NUMBER?

8. The next point set will be the E.C. (end of curve) coordinates. Input the point number you wish assigned if you do not want the next consecutive number, and stroke

starting tip

You can use the traverse program to check your angular closure **before** you start doing the calculation of the traverse. This is often a time saver.

Start the traverse in the normal way, through the input of the bearing (or azimuth, sometimes easier to work with) of the first course. Then use the ∢ RT. and DEF∢ keys to turn the angles. The difference in the closing bearing (or azimuth), from the first bearing, is the angular error.

If the angular error is within acceptable limits (total error divided by the number of turns), you can "Judgement" adjust your angles before calculating the traverse.

Doing it this way, it doesn't matter if the angles are sometimes angle left, deflection right, etc. There is no need to first turn all of the angles into 'interior' angles to check the closure.

closing

When you have calculated all of the points in the traverse and want to close it, stroke

CLOSE

This brings up the closure menu

OPEN | | A | | B |

Stroke the key that corresponds to the current traverse's TYPE. The output will depend on the type of traverse.

If the traverse is "open", you're done. The output will be the sum of the horizontal distances traversed.

For a closed traverse, you will be prompted for the closing point. If it's a type A, this will be the beginning point.

Input the beginning point number and stroke

ENTER

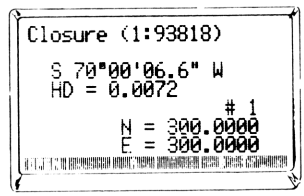
If it's a type B, input the point number for where it should have closed (the known coordinate of the closing point) and stroke either

if the coordinates are stored, [EXIST]

or

if they aren't stored, [NEW]

A typical closing screen should look like the one to the right.

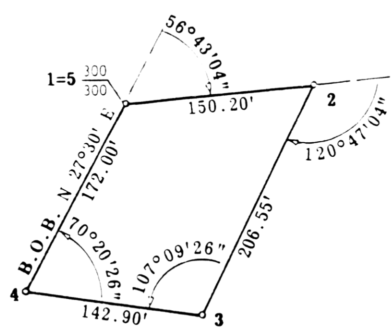


The number in parenthesis is the precision ratio of the traverse. The bearing and distance are **from** the measured point **to** the known point.

The coordinates output are the points of the known closing point. If your traverse is a TYPE 'A', stroking [CONT] will display the area enclosed by the traverse.

keystroke example

We can try a keystroke example using the little traverse shown to the right.



The basis of bearings will be the course **4-1**, and we assume that these are (found) existing points which can be occupied.

The first two angles will be turned as deflection angles to the right, the last two as angles left.

Note that this is an example using a 'basis of bearings'. The course with the known bearing could also have been used directly, that is, we could have started from #4 and gone to #1 as the **first** leg of the traverse.

Stroke

COGO

screen:

```
'NULL'
IS THE CURRENT FILE

DO YOU WANT TO
CHANGE FILES?
```

keystroke:

NO

The following example assumes that the parameters are set for bearing output.

MENU:
TRAV ADJ INTR AREA CRDT EXIT

keystroke:

TRAV

prompt:
POINT NUMBER?

EXIST | NEW | RE #

keystrokes:

1 NEW

prompt:
INPUT COORDINATES

CONT

keystrokes:

3 0 0 SPC

3 0 0 CONT

screen:

```
Beginning at
                # 1
     N = 300.0000
     E = 300.0000
H.DIS S.DIS SIDE CURVE ∠RT. DEF∠
```

keystrokes:

2 7 . 3 ENTER

1 BRNG

added to screen:

N 27°30'00.0" E

keystrokes:

5 6 .

4 3 0 4 DEG

replaced on screen:

N 84°13'04.0" E

keystrokes:

1 5 0 . 2 HDIS

added to screen:

150.2000

prompt: **POINT NUMBER?**

keystrokes:

2 CONT

screen:

```
from 1

N 84°13'04.0" E
150.2000
                # 2
     N = 315.1323
     E = 449.4358
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

[1] [2] [0] [.]
[4] [7] [0] [4] [DEGS]

added to screen:

S 25°00'08.0" W

keystrokes:

[2] [0] [6] [.] [5] [5]
[HDIS]

added to screen:

206.5500

prompt: **POINT NUMBER?**

keystrokes:

[3] [CONT]

screen:

```
from 2
 S 25°00'08.0" W
 206.5500
                 # 3
           N = 127.9378
           E = 362.1367
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

[1] [0] [7] [.]
[0] [9] [2] [6] [+/-] [ENTR]

added to screen:

N 82°09'18.0" W

keystrokes:

[1] [4] [2] [.] [9] [HDIS]

prompt: **POINT NUMBER?**

keystrokes:

[4] [CONT]

screen:

```
from 3
 N 82°09'18.0" W
 142.9000
                 # 4
           N = 147.4427
           E = 220.5741
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

[7] [0] [.]
[2] [0] [2] [6] [+/-] [ENTR]

added to screen:

N 27°30'16.0" E

keystrokes:

[1] [7] [2] [HDIS]

prompt: **POINT NUMBER?**

keystrokes:

[5] [CONT]

screen:

```
from 4
 N 27°30'16.0" E
 172.0000
                 # 5
           N = 300.0024
           E = 300.0067
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

[CLOSE]

screen:

```
 Σ HD = 671.6500


 TYPE?
```

keystroke:

[illegible]

prompt: **CLOSING POINT #?**

EXIST | NEW | RE #

keystrokes:

1 EXIST

screen:

```
Closure (1:93818)
  S 70°00'06.6" W
  HD = 0.0072
               # 1
        N = 300.0000
        E = 300.0000
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

CONT

screen:

```
AREA =
24898.6429 sq ft
= 0.5716 acres
BRNG AZIM CODE INV CLOSE EXIT
```

keystroke:

EXIT

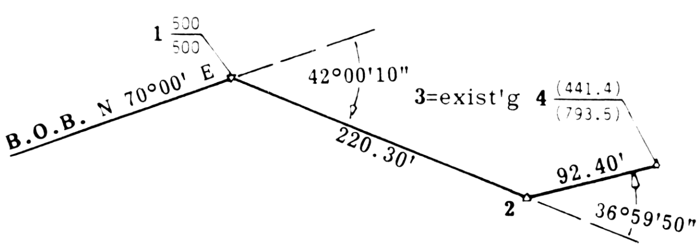
adjustments

In this particular instance, the angular error (difference between the closing bearing and original bearing for the course is 16 seconds. With four angle points, this is 4"/turn, and can be automatically adjusted out with the angle adjust routine prior to any other adjustments being made.

keystroke example 2

To save space, the next example will simplify somewhat, and only show the end output in each case.

This Type 'B' traverse starts by backsighting down a known course from a point with known coordinates (#1, N 500/E 500). In the example, traverse point #3 closes at a point with known coordinates (#4, N 441.4/E 793.5).



We'll begin this example at a point where we have gone into COGO, answered the 'current file' prompt and have the parameters set for bearing output. We also assume that the coordinates N500/E500 for point #1 were input as NEW.

screen:

```
Beginning at

                # 1
          N = 500.0000
          E = 500.0000
BRNG AZIM CODE INV CLOSE EXIT
```

keystrokes:

[7] [0] [SPC]

[1] [BRNG]

[4] [2] [.] [0] [0] [1]

[DEF∠]

[2] [2] [0] [.] [3] [H.DIS]

prompt: **POINT NUMBER?**

keystrokes:

[2] [CONT]

screen:

```
from 1

 S 67°59'50.0" E
 220.3000
                # 2
          N = 417.4643
          E = 704.2546
H.DIS S.DIS SIDE CURVE ∠ RT. DEF∠
```

keystrokes:

[3] [6] [.] [5] [9] [5]

[+/-] [DEF∠]

[9] [2] [.] [4] [H.DIS]

prompt: **POINT NUMBER?**

keystrokes:

[3] [CONT]

screen:

```
from 2

 N 75°00'20.0" E
 92.4000
                # 3
          N = 441.3705
          E = 793.5085
BRNG AZIM CODE INV CLOSE EXIT
```

keystroke:

[CLOSE]

screen:

```
 Σ HD = 312.7000



TYPE?
```

keystroke: [illegible]

prompt: **CLOSING POINT #?**

keystrokes:

[4] [NEW]

prompt: **NORTHING?**

keystrokes:

[4] [4] [1] [.] [4] [CONT]

prompt: **EASTING?**

keystrokes:

[7] [9] [3] [.] [5] [CONT]

screen:

```
Closure (1:10186)

 N 16°00'32.3" W
 HD = 0.0307
                # 4
          N = 441.4000
          E = 793.5000
BRNG AZIM CODE INV CLOSE EXIT
```

[EXIT]

automatic angle adjustment

After a traverse is closed, but before adjustment by Compass or Transit method, the angles should be balanced.

```
 #1  N= 300.0000
     E= 300.0000
N 27°30'00.0" E
N 84°13'04.0" E
HD = 150.2000
 #2  N= 315.1323
     E= 449.4358
S 25°00'08.0" W
HD = 206.5500
 #3  N= 127.9378
     E= 362.1367
N 82°09'18.0" W
HD = 142.9000
 #4  N= 147.4427
     E= 220.5741
N 27°30'16.0" E
HD = 172.0000
 #5  N= 300.0024
     E= 300.0067
```

If you will refer back to the traverse example on page 3-**12** (data shown to the left), you'll note that the original basis of bearings was N 27°30'00" E.

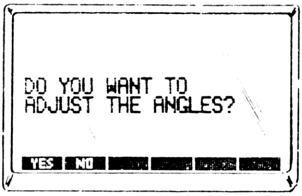
After turning the angles through the traverse, we ended up along the same course, but it now shows the bearing as N 27°30'16" E, indicating that we have 16" too much angle in the traverse.

When you divide the 16" by the four traverse points, you get an **angular error** equal to 4" per turn (you input this as 4).

While this is an acceptable amount of angular error, it still needs to be adjusted out.

The program is accessed by stroking COGO ADJ .

A prompt screen will appear, as shown to the right, asking for a yes or no answer on angle adjustment. In this case you would respond YES .



prompt: **Error/Turn**

Input the number of **seconds** of error per turn (as a whole number), and then stroke CONT

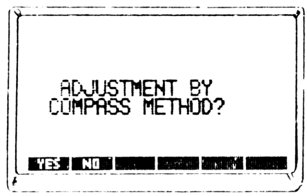
prompt: **Beginning Point #?**

Input the point number used for the first traverse point, and stroke CONT

prompt: **End Point #?**

Input the highest number (not including sideshots) used in the traverse, stroke CONT

If you answer "NO" to the angle adjust prompt, or after the angle adjustment has been completed, the screen to the right will prompt you for the adjustment type.



A "YES" answer will use Compass method, and a "NO" answer will default to Transit method adjustment. The prompts are the same for both methods. If you have just completed the angle adjusment, these prompts will NOT appear, and the adjustment will be made automatically after selection of the type.

If you elected to not adjust the angles (or if you had pre-adjusted them before running the traverse), the prompts are as follows:

prompt: **Beginning Point #?**

Input the point number of the beginning point and stroke CONT

prompt: **End Point Number?**

Input the highest point number used (not including sideshots) for the traverse. Stroke CONT

The adjusted information will be output, one course at a time.

To continue to the next course, stroke CONT

adjusting the sideshots

After completion of the adjustment routine, the sideshots may be reset to the adjusted traverse by inverse traversing from point to point in the "**TRAV**" program.

Inverse (by point number) from the backsight point to the instrument point and reset the sideshots by angle and distance.

The solutions to intersection problems are needed all the time in surveying. We use an intersection formula to find out where two lines cross, then make that point the new PI or the new lot corner. Or, we need to know how far a point is offset from a given line.

Next to just plain traversing, this is the most used type of calculation in surveying. We've tried to make it easy, with all of the options displayed in the menu at one time.

Any distance input is done with the distance key, and any direction can be input as bearing, azimuth or code.

The 'CODE' key may be used to recall a stored bearing between two points in storage. The point numbers are input in the form **AAA** SPC **bbb**, where **AAA** is the first point number, and **bbb** is the second. See page 3-**44** for information on the shifted functions of 'CODE'.

to use the program

Begin by stroking **XEQ**, and then the key corresponding to **INT-X**

Beginning Point?

1. Input the beginning point number, stroke CONT

OUTPUT will be the point number and coordinates of the beginning point.

End Point?

2. Input the point number of the ending point, then stroke CONT

Save as NUMBER?

3. Input the point number you wish to assign to the intersection point, then stroke CONT

NOTE: You can also use **unstored** coordinates. Input the North Coordinate and stroke SPC , Input the East Coordinate and stroke +/- before CONT. The program will prompt for a point number to assign.

The prompts and responses on the previous page are the same for all of the intersection routines. Select the type of intersection you need, and follow the keystroke instructions below.

bearing - bearing

1a. Input the first bearing and stroke **SPC**

Input the quadrant code, stroke

or

1b. Input the point code for the bearing you want to extract as the first bearing, and then stroke

or

1c. Input the azimuth of the first course and stroke

2a. Input the bearing of the second line, then stroke **SPC**

Input the quadrant code and stroke

or

2b. Input the point code for the bearing you want to extract as the second bearing, and stroke

or

2c. Input the azimuth of the second course and stroke

OUTPUT will be the bearing and distance from the beginning point to the intersection, the point number and coordinates of the intersection point, the bearing and distance from the intersection point to the end point, then the point number and coordinates of the last point.

bearing - distance

1a. Input the bearing of the first course, stroking **SPC**

Input the quadrant code, then stroke [illegible]

or

1b. Input the point code for the bearing you want to extract, and stroke [illegible]

or

1c. Input the azimuth of the first course, and then stroke [illegible]

2. Input the distance for the second line, and stroke [illegible]

OUTPUT will be the bearing and distance from the beginning point to the intersection, the point number and coordinates of the intersection point, the bearing and distance from the intersection point to the end point, then the point number and coordinates of the last point.

Because there are two possible answers with this solution type, a reminder prompt appears

2nd Solution

YES NO

3. Examine the answers and decide if they are the correct solution. If they are not, go on to the second solution by stroking YES

Output will be the same as for the first solution with the exception that the first point is not printed out again. The intersection point coordinates in storage will be replaced by the new ones.

distance - distance

1. Input the first distance, stroke DIST

2. Input the second distance, and then stroke DIST

OUTPUT will be the bearing and distance from the beginning point to the intersection, the point number and coordinates of the intersection point, the bearing and distance from the intersection point to the end point, then the point number and coordinates of the last point.

2nd Solution

YES NO

3. If you want the second solution, stroke YES

   If you do not want the second solution, stroke NO

Output of the second solution is similar to that of the bearing - distance solution, replacing the original coordinates at the intersection point with the new ones.

offset to a line

1a. Input the bearing of the known line (from which the end point is offset) and stroke SPC

   Input the quadrant code then stroke BRNG

or

1b. Input the code for the bearing you want to extract and stroke CODE

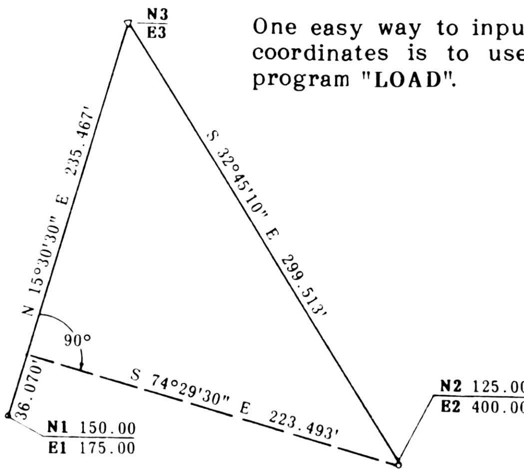
or

1c. Input the azimuth of the line and stroke AZIM

2. Stroke the key which corresponds to OFS

Before beginning with the keystroke examples, we need to store points **1** and **2** in the illustration below.

One easy way to input the coordinates is to use the program "**LOAD**".



We'll start with an example using the **bearing-bearing** routine.

prompt: **Beginning Point?**

keystrokes:

1 CONT

output: #1 N= 150.0000
E= 175.0000

prompt: **End Point?**

keystrokes:

2 CONT

prompt: **Save as NUMBER?**

keystrokes:

3 CONT

prompt bar:

BRNG CODE [illegible] DIST [illegible] NEXT

keystrokes:

1 **5** **.** **3** **0** **3** **SPC**

1 BRNG

prompt bar:

BRNG CODE [illegible] DIST [illegible] NEXT

keystrokes:

3 **2** **.** **4** **5** **1** **SPC**

2 BRNG

output:

N 15°30'29.3" E
Dist = 235.467
#3 N= 376.8941
E= 237.9586

CONT

S 32°45'10.2" E
Dist = 299.513
#2 N= 125.0000
E= 400.0000

Note that these examples assume that we've already responded to the 'current file' and 'select output' screens.

This is an **offset** example, using the same illustration.

Beginning Point?

keystrokes:

[1] [CONT]

output:

```
#1   N= 150.0000
     E= 175.0000
```

End Point?

keystrokes:

[2] [CONT]

Save as NUMBER?

keystrokes:

[3] [CONT]

prompt bar:

Since the bearing that we want (from 1 to 3) is still stored from the last example, let's just call it up from storage:

keystrokes:

[1] [.] [0] [0] [3] [COO]

When the display clears and the **INPUT** prompt appears, indicate that this is an OFFSET:

keystroke:

[OFS]

output:

```
N 15°30'30.0" E
Dist = 36.070

#3   N= 184.7571
     E= 184.6444
```

[CONT]

```
S 74°29'30.0" E
Dist = 223.493

#2   N= 125.0000
     E= 400.0000
```

For a last example, which also uses the 'second solution', do the **distance-distance** intersection, using the same coordinates as in the illustration, and the distances which were output in the first example.

Answer the beginning prompts with the same point numbers as in the other two examples, and then input the first distance

keystrokes:

[2] [3] [5] [.] [4] [6] [8]

[DST]

When the display clears and the **INPUT** prompt appears, input the second distance.

keystrokes:

[2] [9] [9] [.] [5] [1] [3]

[DST]

The output which follows is not the correct answer for the directions we are going:

output: S 2°49'40.4" E

Dist = 235.467

#3 N= -85.1803
E= 186.6170

CONT

N 45°25'59.6" E

Dist = 299.513

#2 N= 125.0000
E= 400.0000

prompt: **2nd Solution**

YES NO

keystroke:

YES

output: N 15°30'29.8" E

Dist = 235.467

#3 N= 376.8941
E= 237.9586

CONT

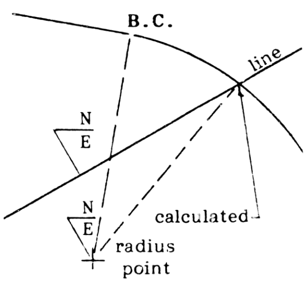
S 32°45'10.2" E

Dist = 299.513

#2 N= 125.0000
E= 400.0000

line to curve intersection

We often need to calculate the intersection point of a line with a curve. This can be done using the **bearing-distance** routine.



What you need to know (and usually do) is the radius of the curve, the coordinates of its radius point, a known coordinate anywhere on the line and the direction of the line.

The line's direction can be either azimuth or bearing.

In the illustration (above) the known point along the line would be used as the beginning point, the radius point of the curve as the end point.

The known bearing is that of the line, and it is input in the direction **toward** the curve.

The second part uses the radius of the curve as the known distance.

stationing

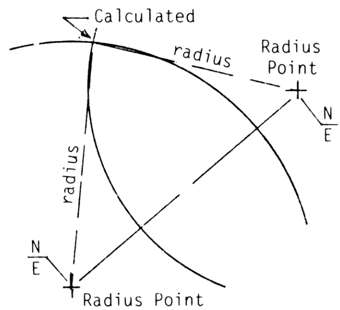
If the station along the curve is needed, it can be easily calculated by using the difference between the output radial bearing and the radial bearing at the B.C. of the curve (see "**TRIG**", 4-**22**). Once the length is known, it is added to the station at the B.C. to obtain the station at the intersection point.

curve to curve intersection

Use the **distance-distance** routine to solve this one. Be cautious when your angle of intersection approaches 90°, as solutions may 'bracket' the true solution when too close to a 90° angle.

The two things that need to be known in this case are the coordinates of the radius points and the radii.

One radius point is used as the beginning point and the other as the end point. Use the radius of the first as distance one.



Stationing along either of the curves is found in the same manner as above.

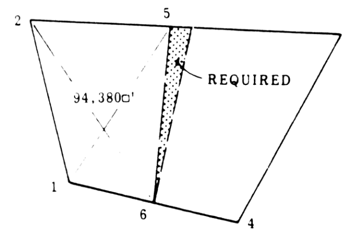
Note that if you inverse between the two radius points, this may also be solved as a triangle with three sides known ("**TRI**", 4-**3**).

Once the angles at the radius points are known the radials to the intersection point may be calculated by adding or subtracting from the bearing along the inversed side.

There are two types of solution routines for solving for a predetermined amount of area. These are used to "part the land", or cut off a specific quantity of property from a larger parcel.

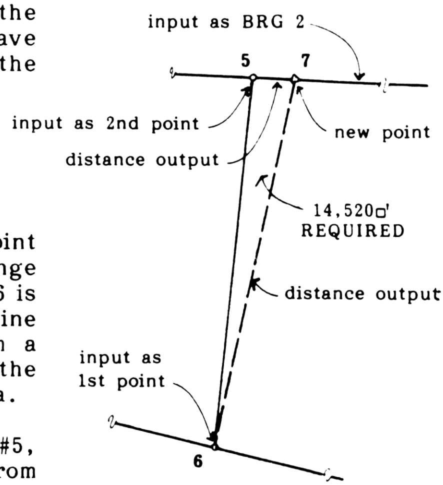
The illustration (below) shows a typical use of the type of solution called **Line Through A Point**. A parcel boundary has been run (points 1 through 4) and the area calculated.

To divide it into two equal parcels, first set two arbitrary points, 5 and 6 (6 is the half-way point along the base 1-4) and calculate the area of one parcel.



TOTAL AREA
= 217,800□'

In this case, the area of the parcel 1, 2, 5, 6, 1 was run, and the shaded portion represents the remainder needed to have each parcel contain ½ of the original area.

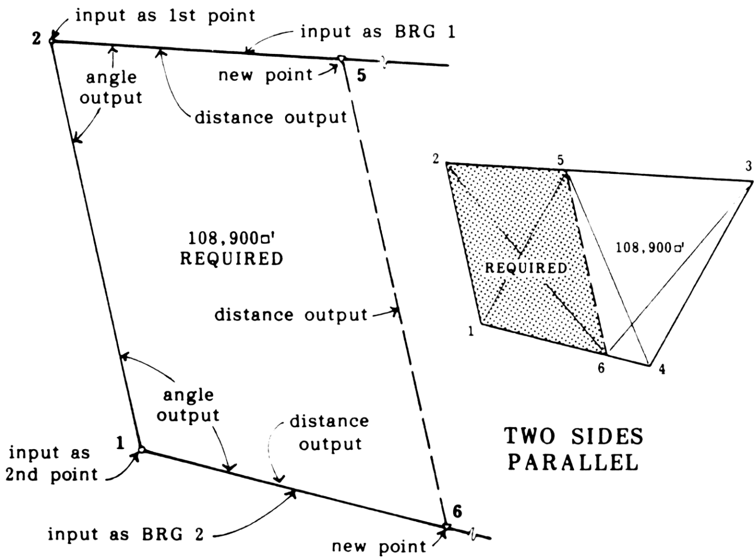


In this example, point number 6 is used as hinge point, and a line from 6 is intersected with the line from #5 to #3 to form a triangle that contains the required amount of area.

Input would be point #5, the bearing of the line from 5 to 3, point 6 and the area needed.

Two Sides Parallel is the second of the routines, and defines the boundary of a trapezoidal parcel whose area is predetermined.

If the requirement is that the area be one-half of the original parcel, but with the dividing line parallel to one of the lines of the original parcel, we would use this type of solution.



TWO SIDES PARALLEL

Input required for this method would be point #2, the bearing of line 2-3, point #1, the bearing of line 1-4, and the required area. The bearings are input in the direction **away** from the known points.

user instructions

The next two pages contain the instructions for using the program to solve for pre-determined areas.

These are followed by keystroke examples (the examples assume that you have already responded to the 'current file' and 'select output' screens.

line through a point

1st Point Number?

Input the point number for the first point (the "hinge" point) and stroke

NEXT or EXIT

2nd Point Number?

Input the point number which represents the fixed point and stroke

NEXT or EXIT

Bearing 2?

Input the known bearing of the fixed line from the second point, and stroke

SPC

Input the quadrant code in the direction away from the known point, then stroke

CONT

Next Point Number?

Input the UNUSED point number you want to assign to the intersection point, stroke

CONT

Required Area?

Input the required area in square feet (or square meters, etc.) and stroke

CONT

OUTPUT will be the distance and bearing from the hinge point to the new calculated point, then the distance and bearing along the known course from the new point to the second point, followed by the coordinates of the new (intersection) point.

EXIT

two sides parallel

1st Point Number?

Input the point number of the first point

NEW or EXIST

Bearing 1?

Input the bearing from the first point

SPC

Input the quadrant code, in the direction away from the point, and stroke

CONT

2nd Point Number?

Input the point number of the second fixed point and stroke

NEW or EXIST

Bearing 2?

Input the bearing of the line radiating from the second fixed point and stroke

SPC

Input the quadrant code, in the direction away from the point, and stroke

CONT

Next Point Number?

Input the next UNUSED point number. This point number will be assigned to the point along the line defined by bearing 1. The next highest point number will be assigned to the point along bearing 2. Stroke

CONT

Required Area?

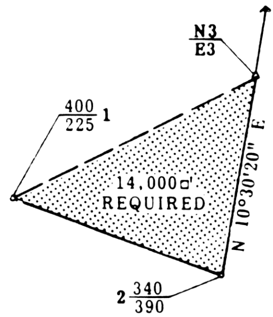
Input the required area and stroke

CONT

OUTPUT will be the distances and bearings of the three lines, followed by the coordinates of the two new points.

We will begin the keystroke examples with the **triangular**, or **line through a point** routine. The illustration to the right will be used for the example.

Input the coordinates for points #1 and #2 (from the example) with "**LOAD**".



prompt: **Triangle?**

YES NO

keystrokes:

YES

prompt: 1st **Point Number?**

keystrokes:

1 ENTER

prompt: 2nd **Point Number?**

keystrokes:

2 ENTER

prompt: **Bearing 2?**

keystrokes:

1 0 . 3 0 2 SPC

1 CONT

prompt: **Next Point Number?**

keystrokes:

3 CONT

prompt: **Required Area?**

keystrokes:

1 4 0 0 0 CONT

output:

```
1-3 Dist = 218.218
    N 63°01'37.7" E
3-2 Dist = 161.687
    S 10°30'20.0" W
                # 3
       N = 498.9768
       E = 419.4805
```

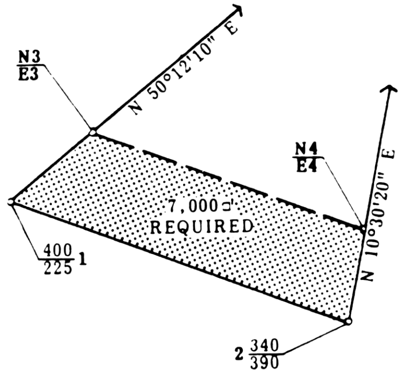
keystroke:

EXIT

It is always a good idea to run an inverse traverse to check the area.

The next example uses the same coordinates for points #1 and #2, so you won't have to input new coordinates.

For the example of the **two sides parallel** routine, we will use the illustration shown to the right.



prompt: **Triangle?**

keystrokes:

[NO]

prompt: **1st Point Number?**

keystrokes:

[1] [EXIST]

prompt: **Bearing 1?**

keystrokes:

[5] [0] [.] [1] [2] [1] [SPC]

[1] [CONT]

prompt: **2nd Point Number?**

keystrokes:

[2] [EXIST]

prompt: **Bearing 2?**

keystrokes:

[1] [0] [.] [3] [0] [2] [SPC]

[1] [CONT]

prompt: **Next Point Number?**

keystrokes:

[3] [CONT]

prompt: **Required Area?**

keystrokes:

[7] [0] [0] [0] [CONT]

output:

```
1-3 Dist = 50.923
    N 50°12'10.0" E
3-4 Dist = 142.595
    S 70°01'00.8" E
4-2 Dist = 44.611
    S 10°30'20.0" W
```

[CONT]

```
      # 3
N = 432.5941
E = 264.1246
      # 4
N = 383.8634
E = 398.1340
```

[EXIT]

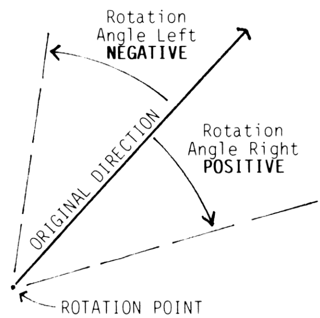
Coordinate transformation, or bearing rotation is used to change a traverse from one "grid system" to another. A field traverse may be run without knowing the real basis of bearings, by beginning with an assumed bearing.

When a basis of bearings becomes known (we finally got the description from the client), the bearings of the traverse may be rotated to match the "deed" bearings.

angle convention

The difference between the assumed (old) bearing along a known line and the deed (new) bearing of the same line is called the ROTATION ANGLE.

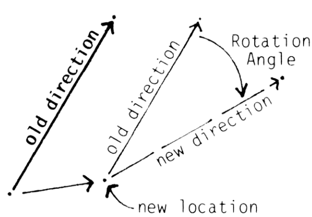
This program accepts the rotation angle as POSITIVE for clockwise and NEGATIVE for counterclockwise.



input options

There are two types of setup input possible with this program. The first is used when the rotation angle is known.

With this system, the whole figure is rotated a known amount of angle, with one of the existing points held as the rotation, or pivot, point (see above).

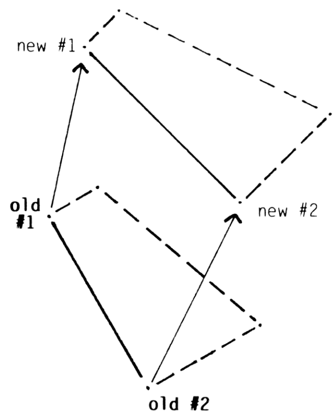


Using this input option, you may also change the location of the rotation point. This is like saying that you are going to pick the whole figure up, move it to the new location and turn it, all at the same time.

You may also change the scale during the rotation process, by introducing a scale factor.

The second of the input options is used when two points in the 'old' system also have known coordinates in the 'new' system.

Because you are giving the program the two sets of coordinates, the program calculates both the rotation angle and scale factor, based upon the difference in azimuth and distance between the two 'old' coordinates and the two 'new' coordinates.



This system is sometimes used to relate a ground traverse to grid system coordinates.

The instructions for input of the required information for each of the systems is shown on the following pages.

Once the setup information has been input for either system, the solution steps for transformation of the points are the same for both systems.

additional options

There are two options available within the program for control of the output and processing. You can select **inverse**, and the output will include the direction and distance of the courses between the rotated points, and you can select **block** to do the output automatically. When the given block of point numbers has been processed, you can also rotate any additional points.

renumber option

A unique feature within this program allows you to transform points and renumber them at the same time, leaving the original coordinates as they were, in the old system. This can be handy for calculating the location of similar buildings on different lots (example, page 3-**42**).

rotation angle known

Call the program up from the base menu by stroking

CROT

prompt: **Rotation Angle?**

CONT

Input the rotation angle, in degrees, minutes and seconds. If the angle is counterclockwise, change sign with the key, then stroke

CONT

prompt: **Scale Factor?**

CONT

If the scale factor is 1:1, it is not necessary to input anything. If it is <u>not</u> 1, input the new factor before stroking

CONT

prompt: **ROTATION POINT NUMBER?**

EXIST NEW RE #

Input the point number of the rotation point, stroke

EXIST or NEW

OUTPUT will be a display "Rotation @", followed by the point number and coordinates of the rotation point.

prompt: **New Coordinates?**

YES NO

If the coordinates of the rotation point are the same in the new system as in the old, stroke

NO

If there are different coordinates for this point in the new system, stroke

YES

If the answer to the last prompt was "yes", you will also receive the following prompt:

prompt: **INPUT COORDINATES**

CONT

Input the new north-coordinate, stroke SPC

Input the new east-coordinate, stroke CONT

OUTPUT will show "which becomes" followed by the new coordinates. NOTE: This point has NOT been transformed yet. You will still need to input this point number for translation later.

prompt: **OPTIONS**

INVS BLK CONT

go to "OPTIONS"

two points in each system known

When the prompt **Rotation Angle?** is displayed, no input is necessary. Just stroke CONT

prompt: **ROTATION POINT NUMBER?**

EXIST NEW RE #

Input the point number of the pivot point, stroke EXIST or NEW

prompt: **New Coordinates?**

YES NO

If the rotation point will have new coordinates in the new system stroke YES. If the coordinates are the same in both systems, stroke NO.

If the answer was "yes", you will receive the following prompt:

prompt: **INPUT COORDINATES**
CONT

Input the new north-coordinate, stroke

SPC

Input the new east-coordinate, stroke

CONT

prompt: **SECOND POINT NUMBER?**
EXIST NEW RE-#

Input the point number of the second known point and stroke

EXIST or NEW

prompt: **INPUT COORDINATES**
CONT

Input the new north-coordinate of the second point and stroke

SPC

Input the new east-coordinate of the second point and stroke

CONT

The direction and distance to this point from the pivot point will be output, followed by the adjusted coordinate. This point HAS been transformed.

prompt: **OPTIONS**
INVS BLK CONT

OPTIONS

At this point in the program you decide on the output you want. This can be just the new coordinates for the transformed points, or you can also calculate the directions and distances between the new coordinates, by stroking

INVS

If you plan to use both options, stroke the INVS key first, then proceed with the 'block'. With or without the inversing, you can automatically rotate a block of coordinates as follows:

Input the beginning (lowest) point number in the block and stroke

SPC

input the highest number in the block and stroke

CONT

With the 'block' option, the output will begin automatically, transforming all of the points within the block before stopping.

After selecting the options you want, stroke CONT

renumbering

You can change the point numbers as the points are rotated if you stroke RE # before stroking →NEW. This option will NOT work if you are rotating a block of coordinates, and requires the input of each point.

prompt: **New Point Number?**

CONT

Input the number you want the point to have in the NEW system and stroke

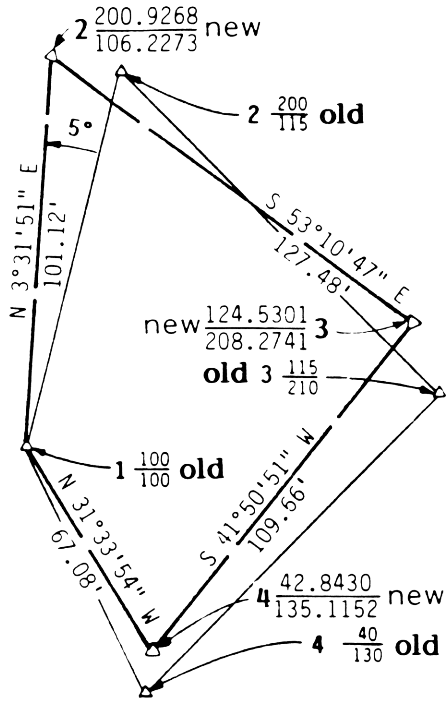
CONT

OUTPUT will be the transformed coordinates under the new point number. The original point number still retains the original coordinates.

keystroke examples

The next few pages contain keystroke examples, which you can use to familiarize yourself with the way the program works.

The inverse output in the examples is determined by pre-selection through PARA.



The small traverse shown above will be used for the keystroke examples. Before beginning, we can get some practice with "**LOAD**", using it to input the "old" system coordinates. Use "**PARA**" to set azimuth output.

For the first example we'll rotate the bearings 5° to the left. We can also use the **auto-inverse** and **block** options.

stroke

CROT

prompt:
Rotation Angle?

keystrokes:

5 +/- CONT

prompt:
Scale Factor?

keystrokes:

CONT

prompt:
ROTATION
POINT NUMBER?

EXIST NEW RE #

keystrokes:

1 EXIST

OUTPUT:

printer:

```
Rotation @
                # 1
   N = 100.0000
   E = 100.0000
```

prompt:

New Coordinates?

YES | NO

keystroke:

NO

prompt:

OPTIONS:

INVS | BLK | CONT

keystrokes:

INVS

2 SPC 4 BLK CONT

OUTPUT:

printer:

```
Az = 3°31'50.8"
HD = 101.119
  New
                # 2
   N = 200.9268
   E = 106.2273

Az = 126°49'12.6"
HD = 127.475
  New
                # 3
   N = 124.5301
   E = 208.2741

Az = 221°50'51.4"
HD = 109.659
  New
                # 4
   N = 42.8430
   E = 135.1152
```

prompt:

NEXT POINT?

keystrokes:

1 NEW

OUTPUT:

printer:

```
Az = 328°26'05.8"
HD = 67.082
  New
                # 1
   N = 100.0000
   E = 100.0000
```

prompt:

NEXT POINT?

keystroke:

EXIT

Reload (or re-rotate) points 2, 3 & 4 for the second example. This will be an example of the type where the coordinates of two points are known in each system, and uses "new" coordinates of 200/200 for #1 and 300/215 for #2.

Switch to bearing output ("**PARA**") before starting.

keystroke:

CROT

prompt:

Rotation Angle?

keystrokes:

CONT

prompt:

ROTATION

POINT NUMBER?

EXIST | NEW | RE #

keystrokes:

1 EXIST

OUTPUT:

printer:

```
    Rotation @
                 # 1
        N = 100.0000
        E = 100.0000
```

prompt:

New Coordinates?

YES NO

keystroke:

YES

prompt:

INPUT COORDINATES

keystrokes:

2 0 0 SPC

2 0 0 CONT

OUTPUT:

printer:

```
 which becomes
                 # 1
        N = 200.0000
        E = 200.0000
```

prompt:

SECOND

POINT NUMBER?

EXIST NEW RE #

keystrokes:

2 EXIST

prompt:

INPUT COORDINATES

keystrokes:

3 0 0 SPC

2 1 5 CONT

OUTPUT

printer:

```
 N 8°31'50.8" E
 HD = 101.119
   New
                 # 2
        N = 300.0000
        E = 215.0000
```

prompt:

OPTIONS:

INVS BLK CONT

keystrokes:

INVS

3 SPC 4 BLK CONT

OUTPUT:

printer:

```
 S 48°10'47.4" E
 HD = 127.475
   New
                 # 3
        N = 215.0000
        E = 310.0000
 S 46°50'51.4" W
 HD = 109.659
   New
                 # 4
        N = 140.0000
        E = 230.0000
```

screen:

```
  NEXT POINT?
 S 46°50'51.4" W
 HD = 109.659
   New
                # 4
       N = 140.0000
       E = 230.0000
 RE #  →NEW  EXIT
```

keystrokes:

1 →NEW

OUTPUT:

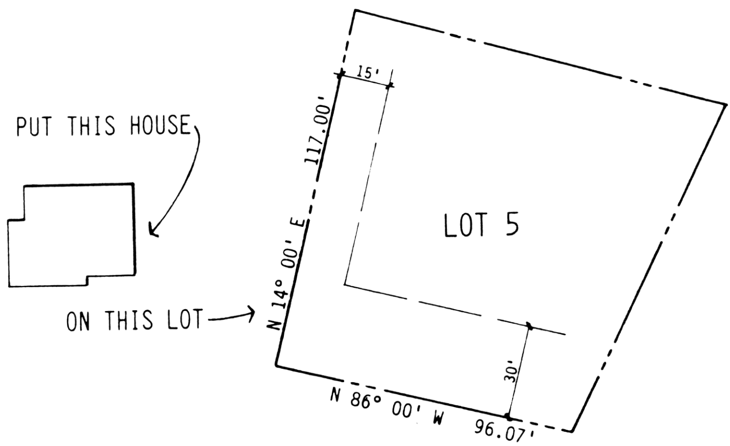
printer:

```
 N 26°33'54.2" W
 HD = 67.082
   New
                 # 1
        N = 200.0000
        E = 200.0000
```

keystroke:

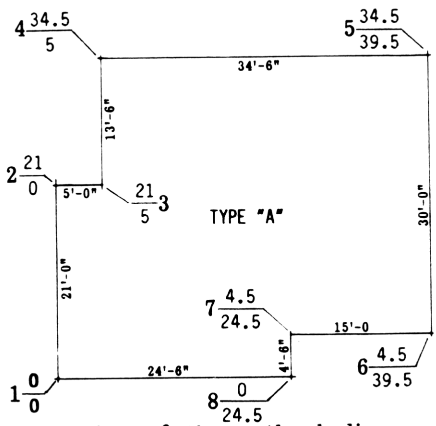
EXIT

If this problem looks familiar, you'll be happy to know that this program has a quick solution to it.



Quite often the client wants to put type "A" houses on lots 3, 5, 8 and 11 . . . type "B" on lots ....

Use the building's dimensions to set "dummy" coordinates on each of the corners, as shown to the right, and then rotate them into position.



Pre-calculate the intersection of the setback lines, you already know the bearing of the lot line. With this program you can store dummy points for the different types of buildings being used, and when you rotate them, use the re-number feature to give each lot an individual set of coordinates for layout. And, you will still have the original dummy points to use for the next lot.

Input the dummy points (from page 3-**42**) and using the "calculated" coordinates for the setback corner, shown to the right, we'll try it out.

Rotation Angle?

keystrokes:

[1] [4] [CONT]

Scale Factor?

keystroke:

[CONT]

ROTATION POINT NUMBER?

keystrokes:

[1] [EXIST]

output:

```
Rotating @ #1
#1   N= 0.0000
     E= 0.0000
```

New Coordinates?

[YES] [NO]

keystroke:

[YES]

INPUT COORDINATES

keystrokes:

[5] [3] [3] [5] [.] [4] [1]

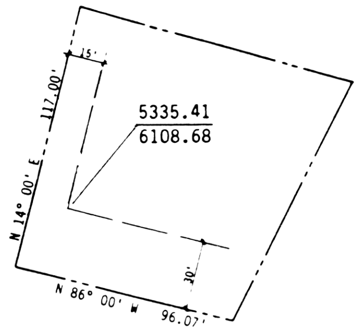
[SPC]

keystrokes:

[6] [1] [0] [8] [.] [6] [8]

[CONT]

Next Point?



keystrokes:

[REC] [1] [ENEW]

New Point Number?

keystrokes:

[2] [1] [CONT]

output:

```
New
#21   N= 5335.4100
      E= 6108.6800
```

Next Point?

keystrokes:

[REC] [2] [ENEW]

New Point Number?

keystrokes:

[2] [2] [CONT]

output:

```
New
#22   N= 5355.7862
      E= 6113.7604
```

Continue through point number 8=28. You can use "**DUMP**" to check that the original points are still intact.

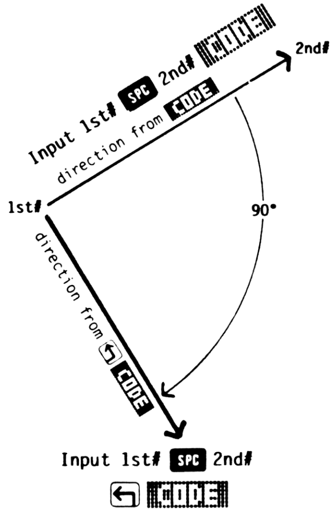
additional 'CODE' functions

In Version 3.0 we have added some shifted functions to the 'CODE' function.

The "**left-shifted**" function will always bring up the bearing (or the azimuth) at right angles to the bearing between the two furnished point numbers.

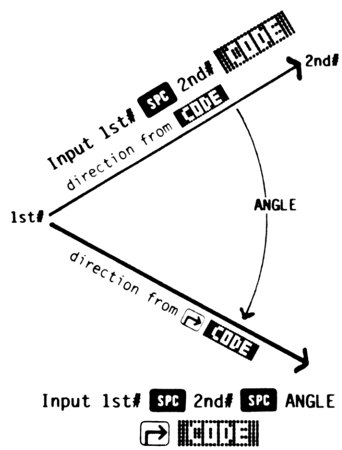
The direction of the line is the same as that which would result from a 90° deflection angle, if turned from the second point, backsighting the first point, or a 90° angle right from the first point while sighting the second.

To reverse the direction of the line, reverse the order of input of the two point numbers.



The "**right-shifted**" function requires the input of the two point numbers **and** an angle, separated by spaces.

The angle may be any angle to the right or (change the sign) to the left. Use this function as though you were at the first point sighting the second.



As an additional piece of information, you can use the function, '**COD**', to bring up a quick inverse, either from the keyboard or in one of your own programs.

Input 2 point numbers and evaluate '**COD**'; the distance will be in Level 2, the azimuth in Level 1. The azimuth is in degrees, minutes and seconds.

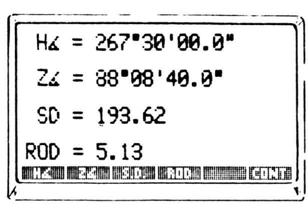
traversing in 3-dimensions

The 'dimension' setting (in PARA) determines how the traverse program works. The 3-DIM programming is meant specifically for field traversing. If you work with your coordinates in 2-DIM, elevations (and/or descriptors) may always be added with edit.

When your parameters are set to 3-DIM, the traverse program begins with prompts for the instrument location, the H.I. and the basis of bearings. After all of that is out of the way, the prompt, "**INPUT SHOT**" appears with the menu for input: H∡ Z∡ S.D. ROD CONT

The program creates a parallel file for the raw (input) data, with a maximum number of 100 points. Again, this program is meant for use while **doing** a traverse, and the main file, if sized at 100, should be large enough to do most work.

When the traverse is completed, the files will be resized to use the minimum amount of space, and the 'raw data' file is identified by a suffix of "R" in the filename. This file may be downloaded to your computer for adjustment with a program such as STAR*NET (Starplus Software Inc., Oakland, CA).



As the shot is input, the input values will appear on the screen as a check. If the value shown is incorrect, you can re-enter it.

After input of the slope distance the current rod height will be displayed. If you want to change it, input the new rod height and stroke ROD before stroking CONT. If the rodperson says that the rod height is still the same, just stroke CONT

Once the shot has been input, it can then be given a label, by using this menu: PT#: NAT DESC SIDE CURV CLOSE

For the next consecutive point number, just stroke PT#, or input a point number **before** stroking

PT#

Use either the DITTO (same as last description) or the DESC key to describe the point. To type in a descriptor, stroke

↱ − α **descriptor** ENTER DESC

Now, label the point as either a sideshot or a traverse point (move-up) by stroking

SIDE or MVUP

Sideshots are stored and the program is recycled for the next shot. Move-up sets the parameters for the instrument next being set at the **current shot**, backsighting the **current point**.

Move to the next point, backsight this one, and continue traversing. When the shot has been entered for the closing point, stroke

CLOSE

The sum of the horizontal distances will be output, displayed with a prompt for the traverse type. These types are discussed on pages 3-**3** and 3-**4**, and there is no actual closure for an OPEN traverse.

For a closed traverse, you will be prompted for the closing point number. In a TYPE A traverse, this will be the original point of beginning. In a TYPE B traverse, the coordinates may be input with NEW. The output which follows gives the precision, closing direction-distance, and the elevation difference.

special functions

Because this program is meant to do the traverse the way we actually do it in the field, functions have been included to facilitate repetition and meaning of the angles and distances.

horizontal angle

The programming provides for both REPEATING and DIRECTIONAL repetition measurement of the angles.

These functions are accessed through the shifted functions of the [HZ], [ZA] and [SD] keys. Input is the rep number, [SPC], measurement.

Let's look first at the method of input using a **repeating** theodolite, which uses the keystrokes ↰ [HZ]

> Input 1, stroke [SPC], input the angle and stroke ↰ [HZ]
>
> Continue with input of the zenith angle and slope distance. After you have doubled the angle, input 2 [SPC], input the accumulated double angle, and stroke ↰ [HZ]

The first angle is displayed as input. As additional angle measurements are input, the MEAN angle will be shown just below the original value. The current number of reps will also be indicated.

```
10°15'20.0"
4→  10°15'17.5"
```

directional theodolite

The input is similar with a directional instrument, with the exception that the input is

shot #, [SPC] backsight reading, [SPC] , foresight reading,

and the shot is entered by stroking ↱ [HZ]

Note! If you use either of the shifted functions for input of the horizontal angle, you MUST use the shifted functions for the zenith angle and slope distance, even if you don't measure the values more than once.

zenith angle

Input is much the same as for the horizontal angle. Input the shot number, stroke [SPC], input the angle and stroke ↰ [ZA]

It doesn't matter whether the scope is erect or inverted during the shot. The angles are meaned as they are input and displayed in the 90±° form.

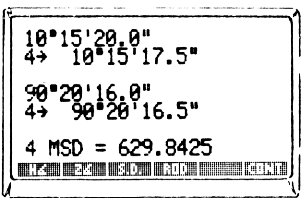
After input of the last reading for the zenith angle, input the slope distance.

slope distance

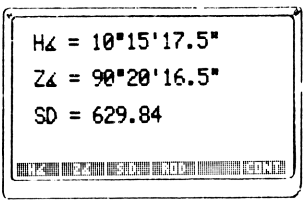
Multiple input is the same as for the angles, input the shot number, SPC, slope distance measurement and stroke

← S.D.

The display to the right shows a completed set (four reps) of measurements. When all of the measurements are completed, stroke CONT



The MEANED angles and slope distance will be displayed, as shown on the second screen, and the program will continue with the rod prompt.



If the rod is the same stroke CONT, or (if it has changed), input the correct (new) value and stroke ROD before you stroke CONT.

The data will be processed and the 'labeling' menu will be shown. PT# RAPT DESC SIDE MUP CLOSE

completing the shot

The shot is completed by the addition of the point number, description and indicator (sideshot or move-up). The adjustment programs (angle, compass or transit) may be used to complete the traverse. The **RAW DATA FILE** should NOT be edited, with the exception of the rod reading. This file is not adjusted by any of the adjustment routines, but may be down-loaded to your PC for use with adjustment routines.

If this is to be done, it is better to **not** adjust the file with the HP48 programs, and just re-load the adjusted PC file later.

GEOM TRI

Surveyors are constantly needing the solution to a triangle to solve field problems. With this in mind, we have included a triangle solution program that is not only easy to use, but also offers more ways to solve a triangle than any other triangle program available.

using the program

It is **always** a good idea to make a sketch of the triangle before beginning. Access the program by stroking GEOM TRI, and the screen and menu shown to the right will be displayed.

```
TRIANGLES
Input 3 known parts,
using the indicated
key each time

Solution is automatic
SIDE ANGLE AREA      MORE EXIT
```

There is very little else to know about this program, just stroke the ▭ key below the menu listing that corresponds to the part you are entering, and enter three parts in order.

area as a part

If one of the known parts is the area, it should be input first, followed by any two other adjacent parts.

two solutions

There are two solution types that will always have a possible second answer, SIDE-SIDE-ANGLE and AREA-SIDE-SIDE. When there is a second solution, the menu prompt bar will change to show

2ND   MORE EXIT

If the first solution was the correct one for your triangle, either stroke MORE to begin a new triangle or stroke EXIT to leave the program.

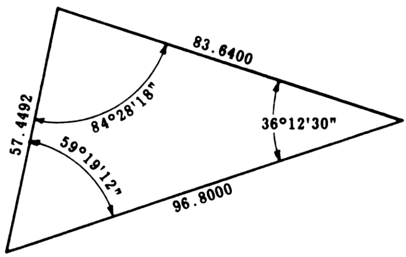
If you want the second solution to be output, stroke 2ND. The printed output is shown to the left.

```
Side 1 = 57.4492
Angle 1 = 84°28'17.9"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.1"

AREA = 2391.3505
```

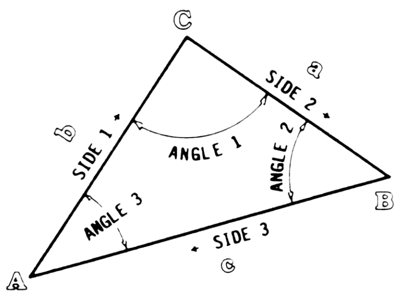
```
2nd Solution:
Side 1 = 83.6400
Angle 1 = 36°12'30.0"

Side 2 = 38.1740
Angle 2 = 120°40'47.9"

Side 3 = 57.4492
Angle 3 = 23°06'42.1"

AREA = 943.0510
```

NOTE: There is no solution for a triangle where the three angles are the only known parts, since this condition can produce an infinite number of similar triangles.

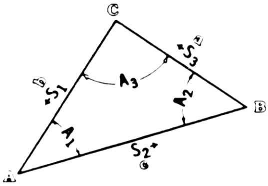
The triangle shown to the right will be used for the examples. It should be noted that the output will vary slightly, depending on the number of places input, particularly in the input of the angles.



The notations for angles and sides is familiar to HP users, but is not the standard, or 'textbook', notation which you have learned in trigonometry (side **a** opposite angle **A**, side **b** opposite angle **B** and side **c** opposite angle **C**). The sides and angles are numbered, in order, going around the triangle.



The example triangle (above) shows this style of labeling, compared to the standard notation for sides and angles. **Side 1** may be assigned to any side that is convenient to use, depending upon the available information about the triangle. It should be located at a side where the known information then falls into position for solution by one of the routines.



In the example, the assigned designations go clockwise. If it will better fit the information available, it may go counter-clockwise instead, as shown to the left.

Side 1, Side 2, Side 3

THREE SIDES KNOWN is one of the most used solutions for triangles, particularly in recent years in surveying.

The lower cost and higher accuracy of electronic distance measurement equipment has resulted in more trilateration being used, instead of time-consuming repetitions of the angles.

keystroke example:* [TRI]

[8] [3] [.] [6] [4]

[SIDE]

[9] [6] [.] [8]

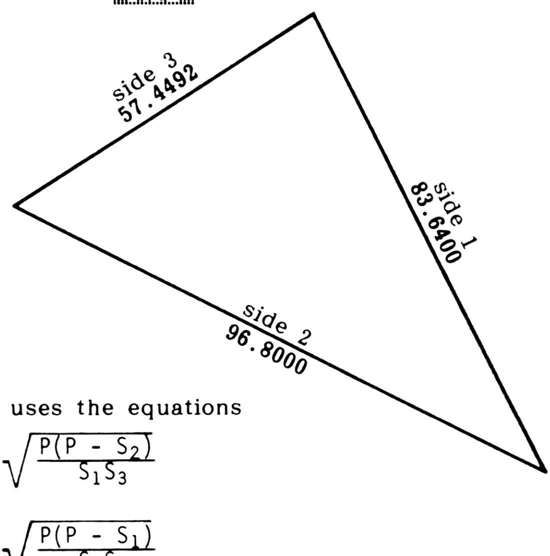
[SIDE]

[5] [7] [.] [4] [4] [9] [2]

[SIDE]

output:

```
Side 1 = 83.6400
Angle 1 = 36°12'30.0"

Side 2 = 96.8000
Angle 2 = 59°19'12.1"

Side 3 = 57.4492
Angle 3 = 84°28'18.0"

AREA = 2,391.3506
```



This routine uses the equations

$$A_3 = 2\cos^{-1}\sqrt{\frac{P(P - S_2)}{S_1 S_3}}$$

$$A_2 = 2\cos^{-1}\sqrt{\frac{P(P - S_1)}{S_2 S_3}}$$

where $P = \frac{1}{2}(S_1 + S_2 + S_3)$

and

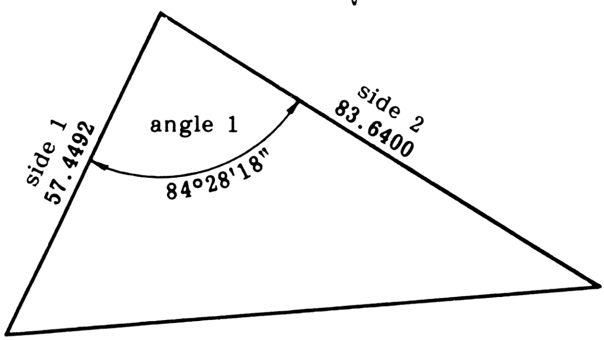
$$A_1 = \cos^{-1}(-\cos(A_3 + A_2))$$

*Note: the display prompt is always the same (page 4-1) and is not shown in the keystroke examples.

Side 1, Angle 1, Side 2

TWO SIDES AND THE INCLUDED ANGLE KNOWN is resolved by finding the third side, and then solving the triangle as shown on the previous page. The third side is found through the use of the equation

$$S_3 = \sqrt{S_1^2 + S_2^2 - 2 S_1 S_2 \cos A_1}$$



5 7 . 4 4 9 2 SIDE

8 4 . 2 8 1 8 ANGL

8 3 . 6 4 SIDE

output:

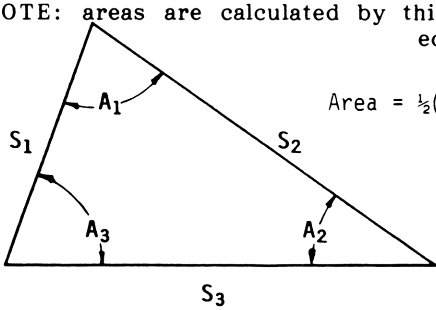
Side 1 = 57.4492
Angle 1 = 84°28'18.0"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.0"

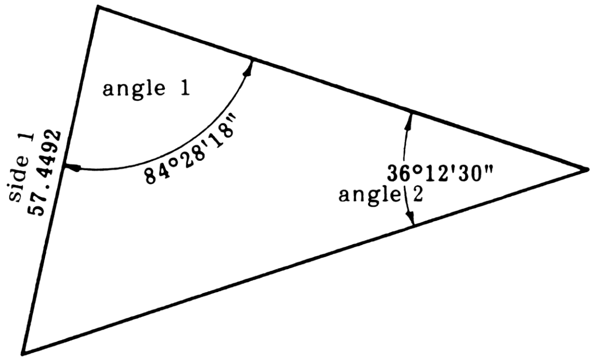
AREA = 2,391.3506

NOTE: areas are calculated by this program using the equation shown below.



$$\text{Area} = \tfrac{1}{2}(S_1 S_3 \sin A_3)$$

Side 1, Angle 1, Angle 2



ONE SIDE AND THE TWO FOLLOWING ANGLES KNOWN. This solution first solves for the third angle with the equation

$$A_3 = \cos^{-1}\left(-\cos\left(A_1 + A_2\right)\right)$$

Once angle 3 has been found, the remainder of the triangle is solved as Angle, Side, Angle (see page 8 for equations) to determine the other missing sides.

5 7 . 4 4 9 2
SIDE
8 4 . 2 8 1 8
ANGLE
3 6 . 1 2 3
ANGLE

output:

Side 1 = 57.4492
Angle 1 = 84°28'18.0"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.0"

AREA = 2,391.3500

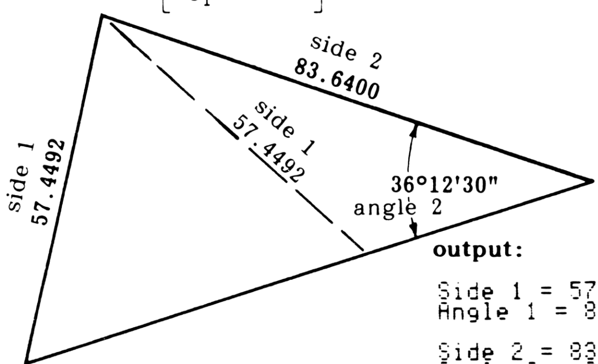
To solve another triangle, stroke [illegible] after the output. To return to the main menu, stroke [illegible]. (typical after all solution output)

Side 1, Side 2, Angle 2

TWO SIDES AND THE FOLLOWING ANGLE KNOWN has two possible solutions. When this configuration is used, both solutions are output. The second solution will not necessarily show the parts in the same order as the input.

The other two angles are calculated with the equations below, and the remaining side is calculated as an Angle, Side, Angle configuration.

$$A_3 = \sin^{-1}\left[\frac{S_2}{S_1}\sin A_2\right] \qquad A_1 = \cos^{-1}\left[-\cos\left(A_2 + A_3\right)\right]$$



5 7 . 4 4 9 2
SIDE

8 3 . 6 4
SIDE

3 6 . 1 2 3
ANGL

END MORE EXIT

END

output:

```
Side 1 = 57.4492
Angle 1 = 84°28'17.9"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.1"

AREA = 2,391.3505

 2nd Solution
Side 1 = 83.6400
Angle 1 = 36°12'30.0"

Side 2 = 38.1740
Angle 2 = 120°40'47.9"

Side 3 = 57.4492
Angle 3 = 23°06'42.1"

AREA = 943.0510
```

Angle 3, Side 1, Angle 1

TWO ANGLES AND THE INCLUDED SIDE ARE KNOWN.

$$S_2 = S_1 \frac{\sin A_3}{\sin A_2}$$

$$S_3 = S_1 \cos A_3 + S_2 \cos A_2$$

$$A_2 = \cos^{-1}\left(-\cos(A_3 + A_1)\right)$$

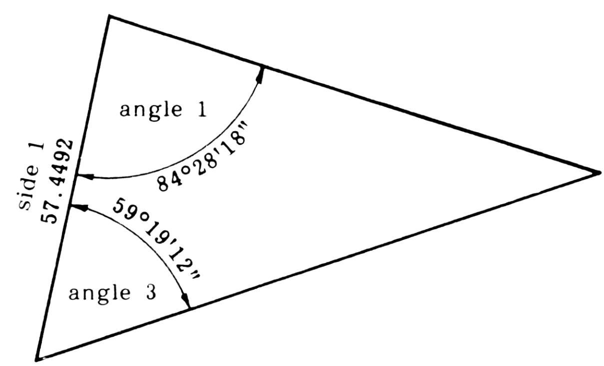
This configuration is solved by using the equations shown to the left.

The Angle, Side, Angle routine has also been used as a secondary solution to some of the other routines, after the problem has first been reduced to these three known parts.

[5] [9] [.] [1] [9] [1] [2]
[ANGL]
[5] [7] [.] [4] [4] [9] [2]
[SIDE]
[8] [4] [.] [2] [8] [1] [8]
[ANGL]

output:

```
Side 1 = 57.4492
Angle 1 = 84°28'18.0"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.0"

AREA = 2,391.3500
```



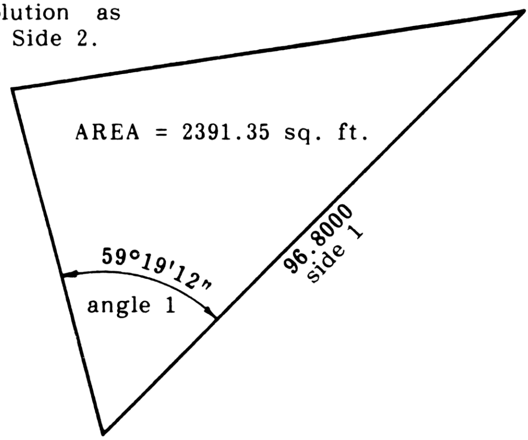
Area, Side 1, Angle 1

THE AREA, ONE SIDE AND ONE ANGLE KNOWN is the first of the three routines in this program which allow the area to be used as one of the known parts. Whenever the area <u>is</u> one of the parts, it is input first.

The equation

$$S_2 = \frac{2\ \mathrm{AREA}}{S_1 \sin A_1}$$

is used first to reduce the problem for solution as Side 1, Angle 1, Side 2.



2 3 9 1 . 3 5
AREA
9 6 . 8
SIDE
5 9 . 1 9 1 2
ANGL

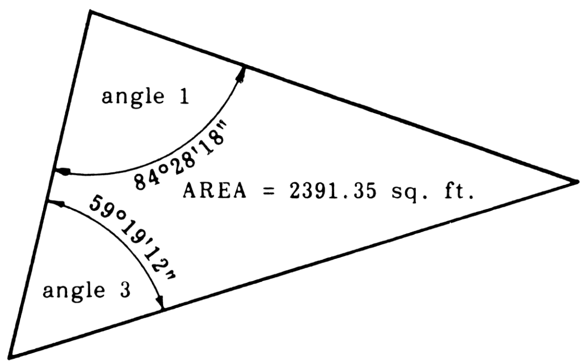
output:

```
Side 1 = 96.8000
Angle 1 = 59°19'12.0"

Side 2 = 57.4492
Angle 2 = 84°28'18.0"

Side 3 = 83.6400
Angle 3 = 36°12'30.0"

AREA = 2,391.3500
```

Area, Angle 3, Angle 1

AREA AND TWO ANGLES KNOWN is first solved for the included side, and then solved as Angle, Side, Angle. The first angle input is treated as Angle 3, the second as Angle 1. The equation used for finding side 1 is

$$S_1 = \sqrt{\frac{2\sin A_2(\text{AREA})}{\sin A_1 \sin A_3}} \quad \text{where } A_2 = \cos^{-1}(-\cos(A_1 + A_3))$$



output:

```
Side 1 = 57.4492
Angle 1 = 84°28'18.0"

Side 2 = 83.6400
Angle 2 = 36°12'30.0"

Side 3 = 96.8000
Angle 3 = 59°19'12.0"

AREA = 2,391.3500
```

2 3 9 1 . 3 5 AREA

5 9 . 1 9 1 2 ANGL

8 4 . 2 8 1 8 ANGL

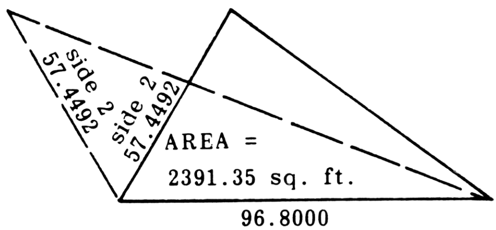
Area, Side 1, Side 2

AREA AND TWO SIDES KNOWN is another problem which has two possible solutions.

We first find Angle 1 with the equation

$$A_1 = \sin^{-1}\left[\frac{2AREA}{S_1\ S_2}\right]$$

and then solve as Side, Angle, Side. The second solution is possible where angle 1 may also be equal to 180° - angle 1. This value is substituted, and the second solution is output.



2 3 9 1 . 3 5 AREA

9 6 . 8 SIDE

5 7 . 4 4 9 2 SIDE

END MORE EXIT

END

output:

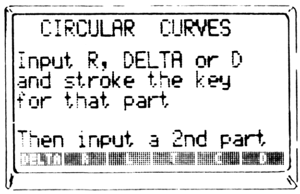
```
Side 1 = 96.8000
Angle 1 = 59°19'12.0"

Side 2 = 57.4492
Angle 2 = 84°28'18.0"

Side 3 = 83.6400
Angle 3 = 36°12'30.0"

AREA = 2,391.3500

 2nd Solution
Side 1 = 96.8000
Angle 1 = 120°40'48.0"

Side 2 = 57.4492
Angle 2 = 37°55'33.4"

Side 3 = 135.4461
Angle 3 = 21°23'38.6"

AREA = 2,391.3500
```

GEOM CIRC

This program is intended to solve circular curves for the unknown parts of the curve. Programs for staking curves are included in the **LAYout** Directory, and give several options for doing the layout.

The input for this program requires that you know the **central angle**, the **radius*** or the **degree of curve*** before beginning. Any one of these parts is input first, and then any other known part.

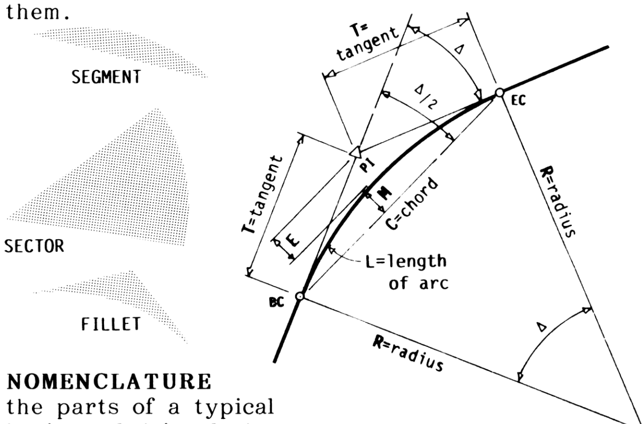
When you access the program, by stroking CIRC, the screen shown to the right is displayed as a reminder. Once the curve data is output the menu prompt bar offers further options.



MORE | | AREAS | | | NEXT

Stroking the MORE key will return you to the top of the "**CIRC**" program, ready to calculate another curve.

Or, stroking the AREAS key will output the areas of the Segment, Sector and Fillet for the curve just calculated. It is not necessary to output the areas unless you need them.



NOMENCLATURE
the parts of a typical
horizontal (circular) curve

*Use of RADIUS or DEGREE OF CURVE varies in different locations. Since they are functions of each other, they can **not** both be used as the two required parts (for input) in this program.

keystroke example

To give you a chance to try out the circular curve program, we have included the example shown below. Keeping it simple, we've used a curve with a central angle of 25°00'00" and a radius of 500'.

To begin, stroke

[GEOM] [CIRC]

screen:

```
  CIRCULAR  CURVES
Input R, DELTA or D
and stroke the key
for that part

Then input a 2nd part
DELTA  R        T  C  D
```

keystrokes:

[2] [5] [DELTA]

[5] [0] [0] [R]

screen:

```
DELTA and  RADIUS
HAVE BEEN INPUT
```

then

```
. . WORKING . . . .
DELTA  R        T  C  D
```

OUTPUT:

```
R = 500.0000
Δ = 25°00'00.0"
L = 218.166
T = 110.847
C = 216.440
M = 11.852
E = 12.140
```

screen:

```
R = 500.0000
Δ = 25°00'00.0"
L = 218.166
T = 110.847
C = 216.440
M = 11.852
E = 12.140
MORE  AREAS  EXIT
```

keystroke:

[AREAS]

OUTPUT:

printer:

```
AREAS:
Segment = 1714.2564
Sector = 54541.5391
Fillet = 882.1265
```

screen:

```
AREAS:
Segment = 1714.2564

Sector = 54541.5391

Fillet = 882.1265
MORE  AREAS  EXIT
```

keystroke:

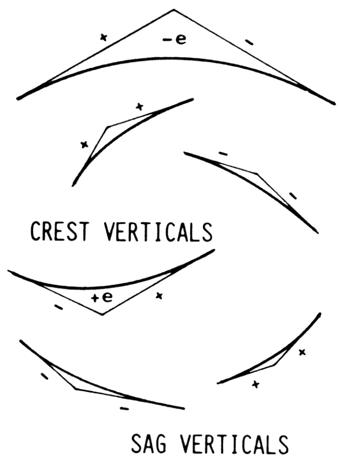
[EXIT]

menu prompt bar:

The main program for vertical alignment solves grade or vertical curve problems, calculating the elevations at any station along the vertical alignment. You can also solve for the station when the elevation is known, on the tangents and in **symmetrical** curves.

types of verticals

Vertical curves are usually described as 'crest' or 'sag' verticals, as shown to the right.



The program will also calculate the turning point (high point on crest, low point on sags) station and elevation. When both tangent grades have the same sign, there is no actual turning point, and the program will tell you that there is no high or low point in the curve.

The symbol, **e**, is commonly used to denote the distance from the tangent intersection to the curve. In this program, the value is computed, and shown as the second (POVC) elevation in the curve data.

A symmetricl curve is one in which the lengths of the two tangents are equal. The program allows selection by input of the length, **L**, if it is a symmetrical curve, or lengths **L1** and **L2** if it is not.

intervals*

Automatic calculation of the elevations at a given interval of stationing is possible. The interval instruction is cancelled at the end of each tangent or curve, and must be reset for the next segment. This allows the use of a different interval along the tangent than in the curve.

*To allow you to take advantage of the interval option, but still be able to calculate specific stations within the portion you are working on, you are prompted for additional station input.

user's instructions

When you stroke [GEOM] [VERT] the prompt screen shown to the right will appear. If you are using a printer, the information will be automatically printed out as it is entered. Then,

```
VERTICAL CURVE
Input, in order, the
Beginning Station,
Beginning Elevation,
and 1st % of grade
```

Input any station which has a known elevation and stroke

[STA]

Input the elevation of the station, and then stroke

[ELEV]

Input the grade (percent). If the grade is negative, stroke [+/-]. Stroke

[GRAD]

The information will also be displayed, giving you a chance to check your input, as shown to the right.

The prompt, **NEXT CURVE?** and the next prompt bar guide you to the next step.

```
BEGINNING @ STATION
10+50.000
Elev. = 106.000
  grade 1 = -1.500%
NEXT CURVE?
```

At this point you may tell the calculator to stop at a given station (when you are using the automatic interval, it will stop calculation at that station), or you can give it one of the indicated stations along the vertical curve.

To stop the calculation at a given station, input the station, then stroke

[STOP]

or

Input the station at the desired part of the curve, then stroke the appropriate key,

[BVC] [PVI] or [EVC]

If a curve was indicated, additional input of the curve's length is needed next, and a new prompt bar appears:

L | | L1 | L2 | |

If the curve is symmetrical, input the total length of the curve and stroke **L**

or

If the curve is asymmetrical, input the length of the first tangent, stroke **L1**

Input the length of the second tangent and stroke **L2**

The required information about your vertical alignment has now been input and the new prompt bar offers the options for types of solution:

STA | | ELEV | | INTVL |

If you want to calculate the station at which a given elevation occurs, input the elevation and stroke **ELEV**

The station at which the (input) elevation occurs will be output.

or

Input the first station for which you want to calculate an elevation. Stroke **STA**

The output, in this case, will be the station (input) and the elevation at that station.

the interval option

After output of the initial station, you may set the interval if you like.

Input the interval you would like to use and stroke

INTVL

This will bring up the prompt for any additional stations to be input. If the printer is ON, output is automatic. If OFF, continue stroking STA each time for the next output, without additional input.

If you calculated an elevation (above) you will want to output a starting station **before** input of the interval.

The program will continue to calculate the elevations until either the next BVC station or the 'stop' station is reached.

When a BVC station is reached, the program will automatically output the first portion of the vertical curve data, and prompt for the outgoing percent of grade.

Input the grade (percent). If it is negative, stroke +/- before

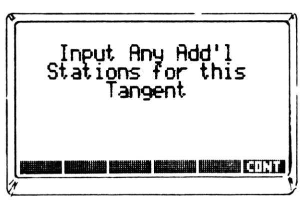
CONT

The output will be the remainder of the curve data, including the turning point and the elevation on the curve at the PVI station (POVC).

At this point you can calculate (symmetrical curve) the station at which a given elevation occurs, if needed, reset the interval, or calculate any station's elevation within the curve.

additional stations

If you have used the interval option you may still calculate more elevations, at any non-interval stations. The calculator will prompt for additional (oddball) stations with the screen shown to the right.



The extra stations needed are input on the command line, separated by a space.

Input the additional stations

station 1 SPC, station 2 SPC, etc.

When all of the 'extra' stations have been input stroke CONT

The stations will be output in order, and the extra stations marked with a *. At the end of each tangent or curve the interval selection is automatically canceled.

keystroke examples

Just because holding two sets of station, elevation and grade produces an oddball station for a PVI (see also **4-23**) doesn't mean that the BVC and EVC have to have oddball stations too.

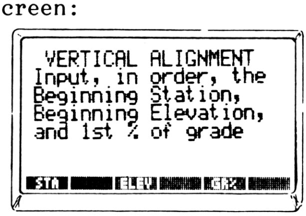
For a first example, assume that station 10+00, elevation 120.00 and station 16+00, elevation 125.00 were held, with grades of +2% and -1.6%. The PVI would fall at station 14+05.556.

We can still hold even stations to begin and end the vertical curve. Let's use 13+00 BVC and 15+00 EVC and calculate it as an asymmetrical curve, with L1 = 105.556' and L2 = 94.444'.

Starting the example at 10+00 gives us a typical vertical tangent situation, followed by our curve. Note that the alignment may be calculated continuously with this program.

Stroke VERT to begin

screen:



keystrokes:

1 0 0 0 STA

1 2 0 ELEV

2 GR%

screen:

```
BEGINNING @ STATION
10+00.000
Elev. = 120.000
  grade 1 = 2.000%

NEXT CURVE?
B.V.C. P.V.I. E.V.C.          STOP
```

keystrokes:

1 3 0 0 B.V.C.

prompt bar:

L | L1 L2

keystrokes:

1 0 5 . 5 5 6

L1

9 4 . 4 4 4

L2

prompt bar:

STA | ELEV | INTVL

keystrokes:

5 0 INTVL

screen:

```
Input Any Add'l
Stations for this
Tangent

                              CONT
```

keystroke:

CONT

output:

```
10+50.00   121.00
11+00.00   122.00
11+50.00   123.00
12+00.00   124.00
12+50.00   125.00

   200.00' V.C.

B.V.C. 13+00.00
   Elev 126.00

  grade in = 2.00%
```

prompt:

GRADE OUT?

keystrokes:

1 . 6 +/- CONT

screen:

```
 P.V.I. 14+05.556
El's: 128.111/127.214

 0% grade @
14+20.469  127.236

  grade out = -1.600%
STA  ELEV  INTVL
```

keystrokes:

5 0 INTVL

prompt:

```
Input Any Add'l
Stations for this
Curve
```

keystrokes:

1 3 6 4 . 9 7

SPC

1 4 2 2 . 1 6

13+25.000

13+50.00 126.80

*13+64.97 126.96

14+00.00 127.19

*14+22.16 127.24

14+50.00 127.15

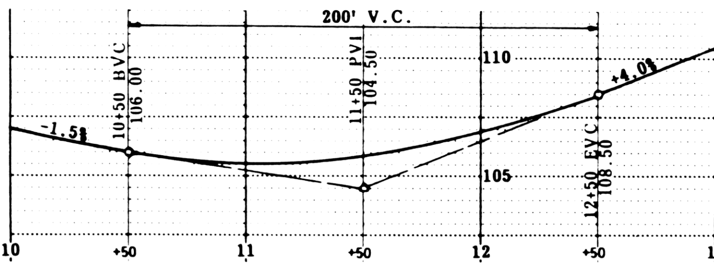
screen:

EVC 15+00.00
Elev 126.60

NEXT CURVE?

etc . . .

The curve below is symmetrical, and we'll use it as a second example, at the same time trying out the ELEV option. Assume that we want to know where elevation 108.00 will be in the curve, and where elevation 110.00 will be in the tangent following the curve.



screen:

VERTICAL ALIGNMENT
Input, in order, the
Beginning Station,
Beginning Elevation,
and 1st % of grade

STA ELEV GRD%

keystrokes:

1 0 5 0 STA

1 0 6 ELEV

1 . 5 +/- GRD%

screen:

BEGINNING @ STATION
10+50.00
Elev. = 106.00
grade 1 = -1.50%

NEXT CURVE?

B.V.C. P.V.I. E.V.C. STOP

keystrokes:

1 0 5 0 B.V.C.

prompt bar:

keystrokes:

2 0 0 L

prompt bar:

STA | | ELEV | | INTVL |

keystrokes:

1 0 5 0 STA

output:

200.00 ' V.C.

B.V.C. 10+50.00
Elev 106.00

grade in = -1.50 %

prompt:

GRADE OUT?

keystrokes:

4 CONT

output:

P.V.I. 11+50.000
El's 104.50/105.88

0% grade @
11+04.545 105.59

grade out = 4.000%

prompt bar:

STA | | ELEV | | INTVL |

keystrokes:

1 0 8 ELEV

output:

ELEV 108.000 at
Sta 9+72.18
and 12+36.91

prompt bar:

STA | | ELEV | | INTVL |

keystrokes:

1 1 0 0 STA

output:

11+00.00 105.59

prompt bar:

STA | | ELEV | | INTVL |

keystrokes:

5 0 INTVL

screen:

Input Any Add'l
Stations for this
Curve

| | | | | CONT

keystroke:

CONT

output:

11+50.00 105.88

12+00.00 106.84

EVC 12+50.00
Elev 108.50

prompt:

NEXT CURVE?

B.V.C. | P.V.I. | E.V.C. | | | STOP

keystrokes:

1 6 0 0 STOP

prompt bar:

STA | | ELEV | | INTVL |

keystrokes:

1 1 0 ELEV

output:

ELEV 110.00 at
Sta 12+87.50

prompt bar:

STA | | ELEV | | INTVL |

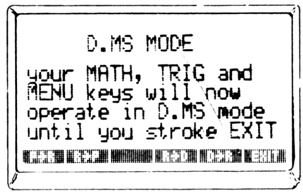
. . . etc.

GEOM TRIG

This is a type of "utility" program, giving you access to the trignometric functions without having to go through 4 keystrokes to get to the conversion keys. We've also added some additional functions which you may find useful.

new functions

When you access this program, by stroking GEOM TRIG, there is a short wait while the calculator assigns the new functions to their keys, and then the screen (right) is displayed.



There are four other new functions which are accessed with menu keys and four (two old and two new) functions assigned to the **shifted** math keys. These all work in Degree, minute and second format, without conversion of either the input or output.

the menu keys

First, to avoid confusion should you want to use these functions in your own programs, the menu keys hold "labels" rather than programs. There are functions, called "R→P" and "P→R" included elsewhere in the programming. They **DO NOT** work in D.MS format, they require the angle to be in decimal degrees. The functions you are using within this mode actually use the others as subroutines.

examples

P→R One way that this function is used is to find the Departure and Latitude of a course with a known azimuth and length. We'll use an azimuth of 25°25'25" and a length of 125.00' for the example.

2 5 . 2 5 2 5 ENTER

1 2 5 P→R

The result (right) shows the departure in Level 2 and the latitude in Level 1.

Reversing the process, stroking [illegible] will return the azimuth (D.ms) to Level 2 and the length to Level 1.

To try out the other two functions, **D→R** and **R→D**, try the following:

Find the length of a curve which has a radius of 500.00' and a central angle of 12°22' (length = the radius times the central angle (D.ms), in radians).

1 2 . 2 2 [illegible]

1: 0.2158

5 0 0 ×

1: 107.9195

Find the central angle of a curve with an arc length of 135.20' and a radius of 450.00'.

1 3 5 . 2 ENTER

4 5 0 ÷ [illegible]

1: 17.1251

the math keys

The shifted functions of the ÷ − × and + keys allow addition, subtraction, multipliction and division of angles directly. For the latter two, the angle is entered first.

Try out these functions with this one: Add 25°15'30" to 13°40'20", subtract 6°15'14", divide by 3 and then multiply by two. The keystrokes are shown below.

2 5 . 1 5 3 ENTER

1 3 . 4 0 2 ↱ +

6 . 1 5 1 4 ↱ −

3 ↱ ÷

2 ↱ ×

(answer 21°47'04")

When using these functions you can shift with either the ↱ or ↰ key.

the trig functions

Use these keys as you normally would, with the exception that all of the input and output is in degrees, minutes and seconds.

GEOM GRAV

This program has been designed to calculate the intersection station and elevation for two grades which pass through known points.

The known information is typed in, using the ☐ key under the menu. The results are displayed, but not printed out. If you want a copy of the printout, the final (answer) screen may be printed out by pressing the ON and MTH keys simultaneousy.

user instructions

Access the program by stroking GEOM GRAV

menu bar: STA1 ELV1 STA2 ELV2 G1 G2

Input the first (down-station) station, and stroke the key under STA1

Input the elevation for that station and stroke ELV1

Input the second (up-station) station , stroke STA2

Input the elevation for that station, and then stroke ELV2

Input the percent of grade for the line through the first station. If the grade is negative, stroke +/- before G1

Input the outgoing grade (in percent). If negative, +/- G2

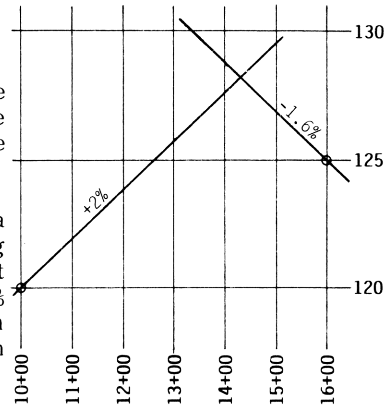
error message

If you give the calculator an impossible set of instructions, "WON'T WORK!" will appear in the display instead of the solution.

keystroke example

For the example of the keystrokes we'll use the information shown to the right.

In the example we have a +2% grade in, passing through station 10+00 at elevation 120.0, and a -1.6% grade out, passing through station 16+00 at elevation 125.0.



keystrokes:

[illegible]

screen:

```
VERTICAL INTERSECTION

  Input, in order,
1st Station, Elevation
2nd Station, Elevation
  Grade 1, Grade 2

STA1 ELV1 STA2 ELV2 G1 G2
```

keystrokes:

1 0 0 0 STA1

1 2 0 ELV1

1 6 0 0 STA2

1 2 5 ELV2

2 G1

1 . 6 +/- G2

OUTPUT:

screen:*

```
10+00.000  120.000
   grade 1 = 2.000%

14+05.556  128.111 PVI

   grade 2 = -1.600%
16+00.000  125.000
MORE                EXIT
```

If you are using a printer, and want a copy for your records, stroke ON and MTH at the same time to print out the screen.

If you want to solve another intersection, stroke MORE, or stroke EXIT to leave the program.

*With stationing above 99+99, or elevations above 999.99, the "PVI" portion of the answer will not be completely displayed.

NXT EVOL

volumes by average end area

This program calculates the end-area of a section and the volume of the average between two sections a known distance apart. It uses a baseline and stationing as the basis.

Access the program through [illegible], stroking NXT EVOL. Everything works from the softkeys, the intervals are calculated for you, and the program saves keystrokes by having separate keys for LEFT, RIGHT and CENTERLINE input.

Input the station and stroke STA

The first and last stations are usually 'zero end-area' so no further input is necessary for them, stroke CONT

You may work clockwise or counterclockwise, starting anywhere on the section, with the input of the elevation and offset points. It is not necessary to re-input the first point again as the last point, just work agound the figure and when finished stroke CONT

The break points are input as

elevation SPC offset

Then stroke either [illegible] or [illegible]

If the shot is a centerline shot, just input the elevation and stroke [illegible]

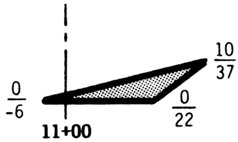
In the example, Station 10+52.5 is a zero end-area station, with sections taken at 11+00, 12+00, 13+00 and the ending zero end-area station at 13+27.

10+52.5

keystrokes:

1052.5 STA CONT

```
Beginning at
     Sta 10+52.50
X-Section Area =
              0 ft²
```

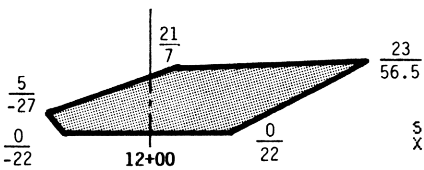



```
Sta 11+00.00
X-Section Area =
         140.0000 ft²
```

keystrokes:

1100 [STA] 0 6 [L] 10 37 [R] 0 22 [R]
[CONT]

```
Sta 10+52.50
to Sta 11+00.00
VOL = 3325.00 ft³
    = 123.15 yds³

Σ = 123.15 yds³
```

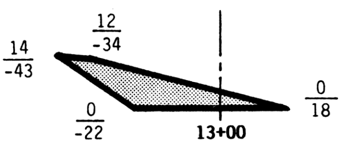



```
Sta 12+00.00
X-Section Area =
        1121.7500 ft²
```

keystrokes:

1200 [STA] 0 22 [L] 5 27 [L] 21 7 [R]
23 56.5 [R] 0 22 [R] [CONT]

```
Sta 11+00.00
to Sta 12+00.00
VOL = 63087.50 ft³
    = 2336.57 yds³

Σ = 2459.72 yds³
```

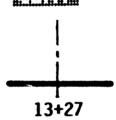



```
Sta 13+00.00
X-Section Area =
         282.0000 ft²
```

keystrokes:

1300 [STA] 0 22 [L] 14 43 [L] 12 34 [L]
0 18 [R] [CONT]

```
Sta 12+00.00
to Sta 13+00.00
VOL = 70187.50 ft³
    = 2599.54 yds³

Σ = 5059.26 yds³
```




```
Sta 13+27.00
X-Section Area =
             0.00 ft²
```

keystrokes:

1327 [STA] [CONT]

```
Sta 13+00.00
to Sta 13+27.00
VOL = 3807.00 ft³
    = 141.00 yds³

Σ = 5200.26 yds³
```

LAYT CURVE

To use the curve layout program stroke LAYT then the key which corresponds to "CURVE". This brings up the first prompt bar DELTA R L T C D

1. Input the two known parts of the curve, the same as when you use the curve calculation program, **"CIRC"**.

Output will be the curve data for the curve, through the output of the chord.

Known Station?

B.C. P.I. E.C.

2. Input the station at the B.C., P.I. or E.C. and stroke the key which corresponds to the station input.

Output will be the stations of the B.C., P.I. and E.C., followed by the prompt bar for selection of the type of layout you want, D+LC D+SC T OFS

3a. For deflection and LONG chord, D+LC

3b. For deflection and SHORT chord, D+SC

3c. For tangent-offset solution, stroke T OFS

STA OFS INTVL

4. If the curve is to be staked along an offset to centerline, input the offset distance (stroke the +/- key if the offset is to the left of centerline) and stroke OFS

5. If you want to calculate stations at constant intervals, input the interval distance and then stroke INTVL

Input the first station for which you want a solution, stroke

[STA]

Output will be the station and either the deflection/chord or tangent/offset solution to the station. If you have selected an interval, the program will continue through the curve without further input, outputting the next station each time you stroke [STA].

When the interval stations have been completed the program will again display the prompt bar, allowing input of additional stations which were not at even interval.

keystroke example

As an example, we'll solve for deflection/long chord layout of a curve with a radius of 500' and a central angle of 30°, along a 10' offset to the left.

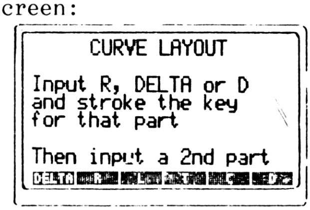
The example uses the interval function to solve at 50' intervals, and the curve is assumed to have a B.C. station of 10+00. After completion of the interval calculations, an additional station will be calculated at station 11+61.27.

The curve layout program is in the "**LAY**" Directory

Stroke

[LAY] [CURVE]

screen:

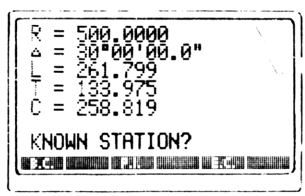


keystrokes:

3 0 [DELTA]

5 0 0 [R]

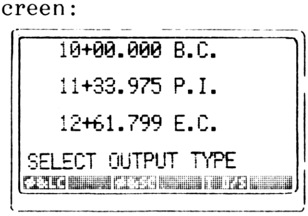
screen:



keystrokes:

1 0 0 0 [BC]

screen:



keystrokes:

[CALC] [1] [0] [+/-] [O/S]

screen:

```
Offset = 10.00' Left
 STA        O/S              INTVL
```

keystrokes:

[5] [0] [INTVL]

[1] [0] [5] [0] [STA]

screen:

```
Sta 10+50.000
Chord = 50.979
ø = 2°51'53.2"
 STA        O/S              INTVL
```

keystroke:

[STA]

screen:

```
Sta 11+00.000
Chord = 101.830
ø = 5°43'46.5"
 STA        O/S              INTVL
```

keystroke:

[STA]

screen:

```
Sta 11+50.000
Chord = 152.427
ø = 8°35'39.7"
 STA        O/S              INTVL
```

keystroke:

[STA]

screen:

```
Sta 12+00.000
Chord = 202.643
ø = 11°27'33.0"
 STA        O/S              INTVL
```

keystroke:

[STA]

screen:

```
Sta 12+50.000
Chord = 252.352
ø = 14°19'26.2"
 STA        O/S              INTVL
```

keystroke:

[STA]

screen:

```
STATION?
 STA                         EXIT
```

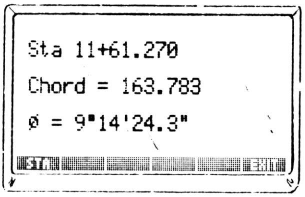
Since all of the 50' intervals are completed, the non-interval station can now be calculated, as follows:

keystrokes:

1 1 6 1 .

2 7 STA

screen:



keystrokes:

EXIT

In the example, the instrument would occupy a point opposite the Beginning of Curve, at ten feet to the left, and either backsight a 10' left offset downstation (and flop the scope) or sight with the full deflection of 15° toward a 10' left offset at the E.C.

NOTE that the chords or offsets output by this program are corrected distances if the curve was calculated along an offset line.

The program has been designed to allow for office printout of the data, to send with a field crew, or for the calculations to be done as needed in the field.

If you are using the program in the field, with an EDM, you can also take advantage of the "**MATH**" routine, which does the 'come in/go out' math for you (see "**SPRAY**", page 5-**15**).

This routine allows you to input the trial shot distance, stroke MTH and get an output of how much further or nearer the rodperson has to be to set the point.

important note

If you begin this program and enter [illegible] but do not complete the curve, EXIT will not be in the menu. To leave the program stroke VAR and then EXIT.

user's instructions

Initialize the program and bring the first of the prompts to the screen by stroking ALIGN

prompt: **SELECT OPTIONS**
COORD SPRAY GRID CONT

If you want to input a grid-to-ground factor, stroke GRID

If you want coordinates calculated and output stroke COORD

If you are calculating angle-distance ties for layout, stroke SPRAY

If you are planning to use the grid factor, it should be the first option selected. After you have selected the option(s) you want, stroke CONT

Different prompts will appear, depending on the options you have selected. Prompts when GRID has been selected are marked (**G**), prompts which appear when COORD has been selected are marked (**C**) and prompts which appear when SPRAY is selected are marked (**L**).

(G) prompt: **Grid to Ground Factor?**
COORD SPRAY GRID CONT

Input the factor (if no correction is wanted, input 1) and stroke CONT

(C) prompt: **First Point number for Coordinate Storage?**

Input the point number you want to start with for the calculated coordinates. Stroke CONT prompt: **INSTRUMENT POINT NUMBER?**

EXIST NEW

Input the point number of the instrument station and stroke NEW or EXIST

If NEW was selected, the following prompt will also appear:

prompt: **INPUT COORDINATES**

CONT

Input the north coordinate, stroke SPC

input the east coordinate for the point and stroke CONT

prompt: **BACKSIGHT POINT NUMBER?**

EXIST NEW

This prompt only appears if angle right output was chosen. Input the point number for the backsight point, stroking either NEW or EXIST. If "NEW" was chosen, input the coordinates of the point in the same manner as described above.

prompt: **KNOWN STATION POINT NUMBER?**

EXIST NEW

Input the point number which will correspond to the beginning station you are going to use. If it is a new pair of coordinates, again follow the input instructions above.

prompt: **Station?**

CONT

Input the beginning station and stroke CONT

prompt: **Tangent Direction?**
BRNG AZIM CODE

Input the bearing of the tangent and stroke SPC

Input the quadrant code, stroke BRNG

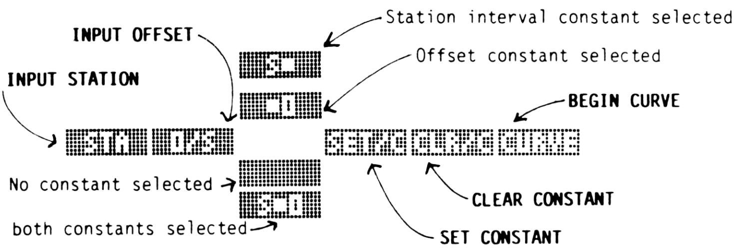
or

Input the azimuth and stroke AZIM

or

Input the code that describes the direction, stroke CODE

At this point you may begin your calculations, using the menu for the input.



using constants

When you input a constant and stroke SET K/C a prompt, "**Station Interval?**" appears. If it is a station interval, stroke YES, and the indicator block will show S. Answer the prompt NO if the constant is an offset, and the indicator block will show O.

If both a station interval **and** an offset constant are stored, the indicator block will show S O.

Constants are cleared in the same way by stroking the CLR K/C key.

Note: When a station interval constant is requested, an additional prompt, **Next Curve?**, appears. This is to prevent over-run of the curve's beginning station.

prompt: **Next Curve?**

If there is a curve in the alignment ahead, input the beginning station and stroke **CONT**

or

If there is no curve in the alignment ahead, input a station that is past the end of the work area, stroke **CONT**

circular curves

Curved portions of the alignment can be included in the calculations, with or without offsets. If offsets are calculated, they are calculated as **radial** offsets to the current station.

Input the B.C. station and stroke **STA**

When the B.C. station is displayed, calculate any needed offsets, and then stroke **CURVE**

prompt: **Central Angle?**

Input the value (°.'") of the central angle. If the curve is to the left, stroke the **+/-** key. Stroke **CONT**

prompt: **Radius?**

Input the radius of the curve, stroke **CONT**

The output will be the central angle, the radius and the length of the curve.

If a station which is upstation of the end of the curve is input, the station **at** the E.C. will be displayed instead.

Intentionally inputting a station which is too large is the correct procedure! It's faster than input of the longer station, with decimal feet included, and it also uses the (more accurate) calculated length, rather than the rounded off value.

station and offset

The prompt bar will be displayed with the begining station above it.

If you want to calculate offsets at this station, input the offset, stroke

OFS

If you do not want offsets at this station, input the station you want and then stroke

STA

then

If you want the default offset, stroke

OFS

If you want a different offset, input the offset and then stroke

OFS

The output will depend on which options you have selected.

If you want an additional offset at this station, input the offset and then stroke

or

If a station interval constant is set stroke STA to bring up the next station.

OFS

or

Input the next station and stroke

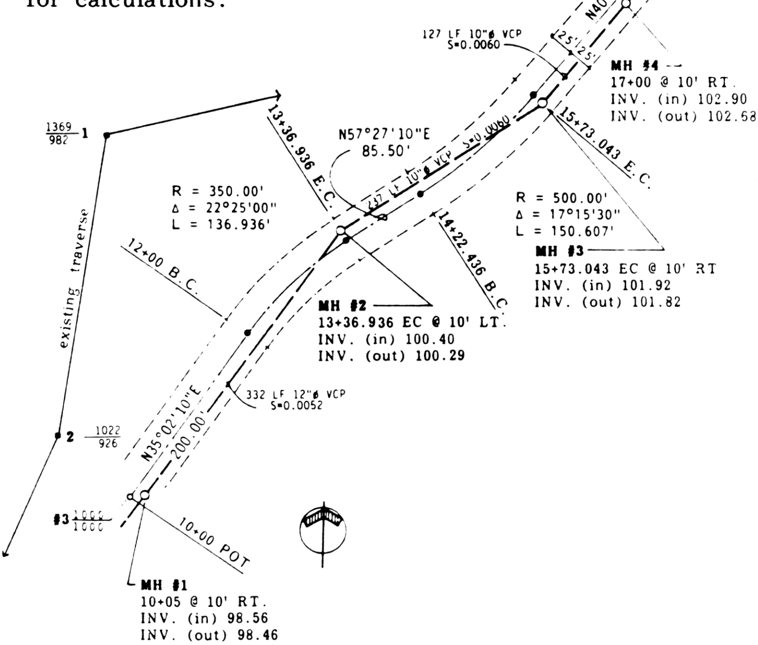
STA

The station is only output with the first offset. The other calculations at the same station will be labeled "@ XX' Rt" or "@ XX' Lt".

The keystroke example will use the alignment illustrated below. The location of the manholes to be installed in the street are offsets from points on the centerline of the street.

Working from the existing traverse, with the instrument at point #1, the example uses point #2 as a backsight.

If we assume that none of the coordinates are in storage, we can use #3 for the number at 10+00, and #4 as our first number for calculations.



Begin by bringing up the program by stroking ALIGN.

SELECT OPTIONS

COORD | | SPRAY | | GRID | CONT

keystrokes:

COORD SPRAY CONT

First Point Number for Coordinate Storage?

| | | | | | CONT

keystrokes:

4 CONT

SELECT OUTPUT

STA | | AZIM | | | |

keystroke:

STA

INSTRUMENT POINT NUMBER?

EXIST | | NEW | | |

keystrokes:

1 NEW

INPUT COORDINATES

| | | | | | CONT

keystrokes:

1 3 6 9 SPC

9 8 2 CONT

BACKSIGHT POINT NUMBER?

EXIST | | NEW | | |

keystrokes:

2 NEW

INPUT COORDINATES

| | | | | | CONT

keystrokes:

1 0 2 2 SPC

9 2 6 CONT

KNOWN STATION POINT NUMBER?

EXIST | | NEW | | |

keystrokes:

3 NEW

INPUT COORDINATES

| | | | | | CONT

keystrokes:

1 0 0 0 SPC

1 0 0 0 CONT

Station?

| | | | | | CONT

keystrokes:

1 0 0 0 CONT

screen:

```
Sta 10+00.000



Tangent Direction?
BRNG AZIM CODE
```

Tangent Direction?

BRNG | AZIM | CODE | | |

keystrokes:

3 5 . 0 2 1

SPC 1 BRNG

screen:

```
Sta 10+00.000



                # 3
       N = 1000.0000
       E = 1000.0000
 STA  O/S      SET/C CLR/C CURVE
```

keystrokes:

1 0 0 5 STA

display:

```
Sta 10+05.000
```

keystrokes:

1 0 O/S

screen:

```
Sta 10+05.000
     @ 10.00' right
∡ Rt = 346°20'58.9"
H.Dist = 371.78
              # 4
       N = 998.3530
     E = 1011.0584
 STA  O/S      SET/C CLR/C CURVE
```

keystrokes:

1 2 0 0 STA

CURVE

Central Angle?

keystrokes:

2 2 . 2 5 CONT

Radius?

keystrokes:

3 5 0 CONT

screen:

```
12+00.000 B.C.

 Δ = 22°25'00.0"
 Radius = 350.000'
 Length = 136.936'

 STA  O/S      SET/C CLR/C CURVE
```

keystrokes:

1 4 0 0 STA

screen:

```
Sta 13+36.936 E.C.




 STA  O/S      SET/C CLR/C CURVE
```

keystrokes:

1 0 +/- O/S

screen:

```
Sta 13+36.936 E.C.
     @ 10.00' left
∡ Rt = 285°18'01.6"
H.Dist = 247.99
              # 5
     N = 1266.2871
     E = 1207.7171
 STA  O/S      SET/C CLR/C CURVE
```

keystrokes:

1 4 2 2 .

4 3 6 STA CURVE

Central Angle?

keystrokes:

1 7 . 1 5 3 +/- CONT

Radius?

keystrokes:

5 0 0 CONT

screen:

```
Sta 14+22.436

Δ = -17°15'30.0"
Radius = 500.000'
Length = 150.607'
```

keystrokes:

1 6 0 0 STA

screen:

```
Sta 15+73.043
```

keystrokes:

1 0 OFFS

screen:

```
Sta 15+73.043
      @ 10.00' right
∡ Rt = 257°09'42.2"
H.Dist = 424.610
                # 6
      N = 1396.1848
      E = 1405.7401
```

keystrokes:

1 7 0 0 STA

screen:

```
Sta 17+00.000
```

keystrokes:

1 0 OFFS

screen:

```
Sta 17+00.000
      @ 10.00' right
∡ Rt = 247°02'13.9"
H.Dist = 520.700
                # 7
      N = 1493.1617
      E = 1487.6758
```

keystroke:

EXIT

The example used both coordinate output and layout (SPRAY) options, but could have used either one by itself. Or GRID. Using two options gives you an idea of what the output should look like for each.

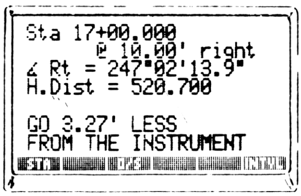
When you use the program with the coordinate option, the points are stored. This means that the actual lengths between the manholes can be found by inversing. With this in mind, the program is also handy for design work on storm and sanitary sewers.

math option

After output of the angle (or azimuth) and distance to a point you want to set, you can use this option to help set the point.

When you take the trial shots, put the rodperson on line and check the distance to the rod.

Input the measured distance and stroke MTH to get a readout similar to the one shown to the right.



There is also a provision for using this option with the slope distance.

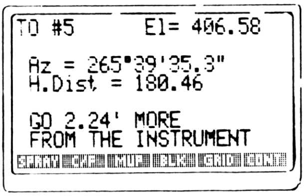
If your EDM is a type that doesn't reduce the slope distance to horizontal internally, you may still use the math option as follows:

Input the slope distance, stroke ENTER

Input the zenith angle and stroke ← MTH

The slope distance will be reduced to horizontal and then used to give you the difference in the distance to the trial shot and the required distance.

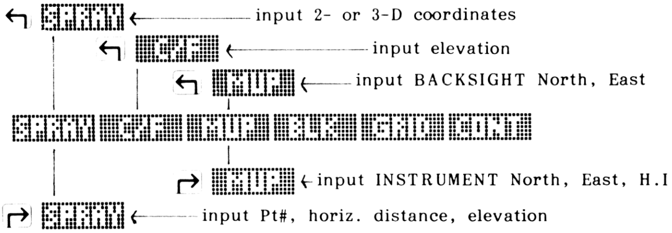
This program is full of options, and is probably the most comprehensive layout program ever written. In addition to the **GRID** and **MATH** options, you may work with stored or unstored coordinates, or input from an office-furnished layout sheet. It will also calculate the cuts or fills using elevations either in storage or input in the field.



If the coordinates include the elevation, it will be shown in the upper right-hand of the screen, and that allows you to use the option as well as the layout portion of the program.

The zenith angle and the slope distance are input, using the **shifted "MATH** option" with the shot on the just set hub. Stroking will bring up both the hub elevation and the cut or fill.

If the coordinates were input as 2-dimensional originally, the elevation may be added to the information by using the **shifted** function of the key, prior to taking the final shot on the hub. The output will be the same as shown.



The **left-shifted** key allows input of the coordinates when they are not already stored. The may be either 2- or 3-dimensional, and are input on the command line separated by spaces, followed by ↰ .

The **right-shifted** key is similar, but takes the input of the point number and pre-calculated distance. To use the cut/fill option, the elevation may also be added at this point, or input with ↰ prior to the actual shot on the hub, after it is set.

The various setup options are explained below.

the block option

You can lay out a consecutive 'block' of points by using this option.

Input the point number of the first point to be set and stroke SPC

Input the last point number of the block, stroke BLK

After each point is set, calculate the next one by stroking CONT

grid correction

If you are working at an elevation where grid correction is a factor to consider, you can enter a grid-to-ground factor prior to input of the first point number for solution.

Input the grid factor and stroke GRID

the move-up option

This one allows you to move to a different setup point without having to do all of the input from scratch.

Input the point number of the new setup point and stroke SPC

If you have selected angle right output, input the point number of the backsight point

or

If you have selected azimuth output, input 0 (zero)

Stroke MUP

There are also shifted functions on the "**MOVE-UP**" key, to be used when you are working from an office-furnished dump sheet and do not have the coordinates stored.

The northing and easting of the new backsight point, separated by a space, can be entered with the **left-shifted** key, and the northing, easting and H.I. of the new instrument location (again, separated by spaces) is entered with the **right-shifted** key.

Input of the H.I. is only needed if you plan to use the cut/fill option, and is the elevation of the scope, not the distance above the point occupied by the instrument.

cut or fill

To use the cut/fill option with stored coordinates that include the finish grade elevation, make sure that your PARAMETERS are set for 3-D before beginning. The elevation will be displayed as part of the layout information.

After setting the hub, take a shot on it, using the **shifted** MTH function (see page 5-**14**). The program will display the cut or fill.

If the elevation is NOT in storage, it may be entered with the **left-shifted** key.

parameters

Again, this is important. You won't get 3-dimensional output if you are set to 2-D, and the status of the printer is also important.

When the printer is ON and you use , the whole block will be printed out. For use in the field, set the printer to OFF.

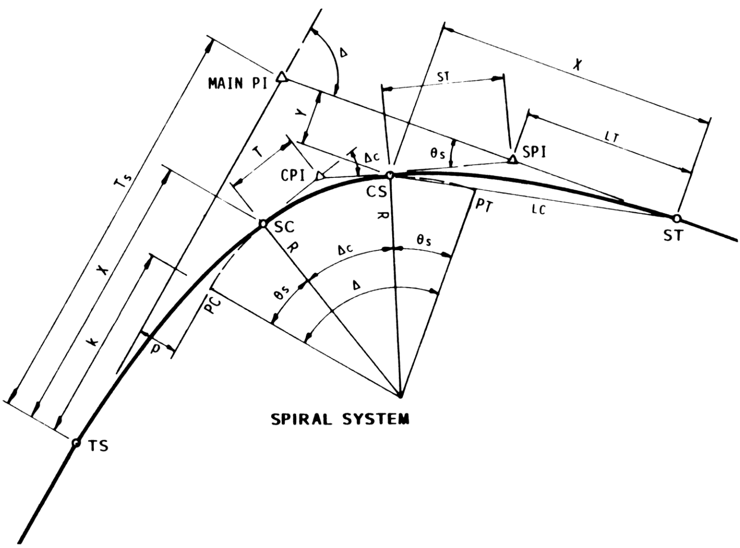
generating the output

When you use the shifted functions of the key, the output is automatic, because you have supplied the data. To use the program with **stored** points, just input the point number and stroke

spiral curves

Spiral curves are used as transition curves, between the tangent and the circular curve, on railroads, highways and rapid transit systems. The latter is probably the most important at the present time, because of the number of rapid transit projects in the last few years.

New funding for interstate highway projects has also renewed the need for the surveyor to be able to work with spiral curves in the field and office.



SPIRAL SYSTEM

The 'parts' of a typical spiral system are shown above. In a spiral system, the circular portion which would normally be at right angles to the two tangents is modified by the amount of arc length subtended by an angle equal to θ_s. This original point of tangency would occur at a distance of **k** from the T.S. and the S.T., at an offset distance of **p**.

The length of the spiral is designated as L_s. Note that the distance, **X**, is the tangent distance to the S.C. (the tangent distance to any point on the spiral is usually shown as **x**). The offset at the S.C. is **Y**, and to any point, **y**. Values to any point on the spriral can be calculated using the same formulas as for the total spiral, since θ_s is the variable.

The formulas used in this program are shown below.

T_S = Tangent length from T.S. to main P.I.

$= (R-P)\mathrm{Tan}\tfrac{1}{2}\Delta + K$

$P = Y - R(1-\mathrm{Cos}\theta_S)$

$K = X - R(\mathrm{Sin}\theta_S)$

$$X = L_S\left(1 - \frac{\delta_S{}^2}{10} + \frac{\delta_S{}^4}{216} - \frac{\delta_S{}^6}{9360} + \frac{\delta_S{}^8}{685440} - \frac{\delta_S{}^{10}}{76204800} \cdots\right)$$

$$Y = L_S\left(\frac{\delta_S}{3} - \frac{\delta_S{}^3}{42} + \frac{\delta_S{}^5}{1320} - \frac{\delta_S{}^7}{75600} + \frac{\delta_S{}^9}{6894720} - \frac{\delta_S{}^{11}}{91808640} \cdots\right)$$

$ST = Y\ \mathrm{Sin}\theta_S$

$LT = X - Y\ (\mathrm{Tan}\theta_S)$

L_S = Length of Spiral

$\theta_S = \frac{L_S\ D_C}{200}$

$LC = \sqrt{X^2 + Y^2}$

D_C = Degree of Curve (arc definition)

The D_C, θ_S and Δ are in degrees.

$\delta_S = \theta_S$, expressed in radians.

All of the other dimensions are in feet.

If the tangent distance and offset to any point are known, the deflection and long chord are easily calculated, so the spiral may be laid out using either method. This program solves for the coordinates of the points, and may be used without having to set up on the spiral itself.

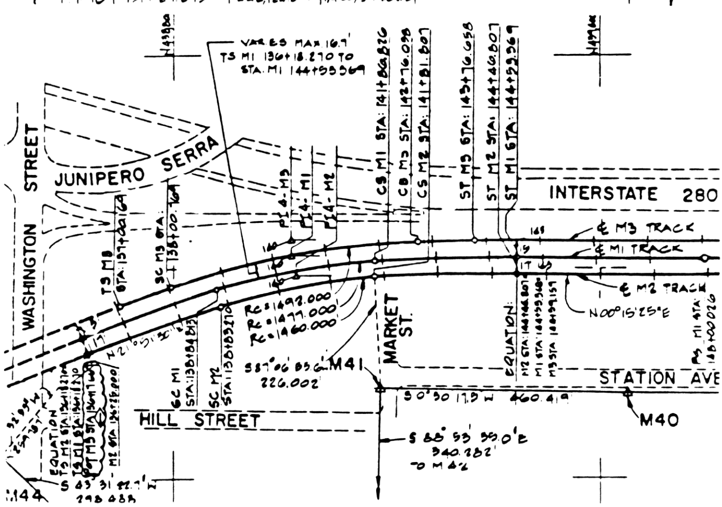
If the instrument is set up at the T.S., sighting toward the P.I., the output is the same as deflection and long chord output, but with the advantage that offsets may also be set.

The illustration below shows portions of a set of plans for BART in the San Francisco Bay Area. For an example of this program's use with a spiral system we will use the **M2** alignment data.

The coordinates are on California State Grid system, and to avoid typing in the parts of the coordinates that aren't within the scope of the job; we'll drop the 4 in the northing and 1,4 in the easting, to make the coordinate values a little more managable.

ALIGNMENT DATA- ℄ M2 TRACK

	POINT	STATION	COORDINATES N	COORDINATES E	CURVE DATA	DESIGN SPEED
M2-LINE	POT	130+99.549	438,156.773	1,430,742.993	PI STA 140+36.445	
	PI-4		439,026.499	1,430,394.100	Δ = 22°06'55" T_{S1} = 418.175' T_{S2} = 418.175'	
	TS	136+18.270	438,638.388	1,430,949.792	L_{S1} = 265.000' X_{S1} = 264.782'	V = 50 MPH
	SC	138+83.270	438,887.117	1,430,456.646	θ_{S1} = 9°11'59.2' Y_{S1} = 8.012'	E = 5.250"
		CC.CURVE 4	439,309.651	1,431,857.310	Δc = 11°42'56.5' R_c = 1460.000' L_c = 298.937'	U = 2.710"
	CS	141+81.807	439,179.855	1,430,402.800	L_{S2} = 265.000' X_{S2} = 264.782'	
	ST	144+46.807	439,444.670	1,430,395.975	θ_{S2} = 9°11'59.2' Y_{S2} = 8.012'	
	PI-10	151+27.015	440,124.871	1,430,399.025		



The plans give the values to the nearest 0.001' and tenth of a second, so you will have some rounding off ($\Delta_C + 2[\theta_S]$ is one tenth second short of adding up to Δ). Calculating the values, holding Δ, R_C and L_S, you will get θ_S=5°11'59.2376" and Δ_C=11°42'56.5249". If these values are input you will get an answer without any roundoff.

using the program

All of the input is exactly the same with the exception that you identify the curve as a **spiral** curve by stroking ↰ before you stroke CURVE. This brings up a menu prompt bar for input with the softkeys:

LS RC DC θS ENTR EXIT

Input the length of the spiral (if this is a curve to the left, stroke +/-) and stroke LS

A second part must be input next. For easiest input the radius or degree of curve for the circular portion is best. It will also give more consistant answers when you are calculating a complete system.

Input the radius of the circular portion and stroke RC

or

Input the degree of curve for the circular portion and stroke DC

or

Input the spiral angle and stroke θS

Next, indicate whether it is an ENTRANCE spiral or an EXIT spiral.

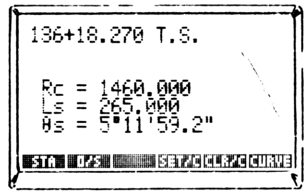
If it is an entrance spiral, stroke ENTR

or

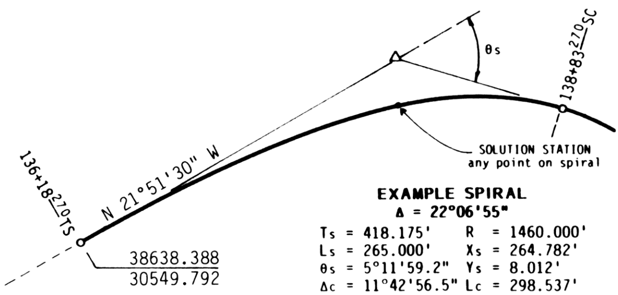
If it is an exit spiral, stroke EXIT

As shown to the right, the spiral data will be displayed (and, if print mode is on, printed out).

The prompt bar for input will again be displayed, and input of the station and offset are the same as in the previous example.



For a keystroke example, we can use the data below, and calculate the even stations. We'll also calculate an offset to these at 20' right and left.



Start by stroking ALIGN

SELECT OPTIONS

COORD | SPRAY | GRID | CONT

keystrokes:

COORD CONT

First Point Number for Coordinate Storage?

CONT

keystrokes:

1 4 CONT

KNOWN STATION POINT NUMBER?

EXIST | NEW

keystrokes:

1 NEW

INPUT COORDINATES

CONT

keystrokes:

3 8 6 3 8 .
3 8 8 SPC
3 0 5 4 9 .
7 9 2 CONT

Station?

CONT

keystrokes:

1 3 6 1 8 .
2 7 CONT

Tangent Direction?

BRNG | AZIM | CODE

keystrokes:

2 1 . 5 1 3 SPC 4 [BRNG]

output:

```
                # 1
N = 38638.3860
E = 30549.7920
```

keystrokes:

↰ [CURVE]

[LS] [RC] [DC] [XS] [ENTRS] [EXITS]

keystrokes:

2 6 5 [LS]

1 4 6 0 [RC]

output:

```
Rc = 1460.000
Ls = 265.000
θs = 5°11'59.2"
```

keystroke:

[ENTRS]

screen:

```
136+18.270 T.S.

 Rc = 1460.000
 Ls = 265.000
 θs = 5°11'59.2"
STA OFS  SETPC CLRPC CURVE
```

keystrokes:

1 3 7 0 0 [STA] [OFS]

screen:

```
Sta 137+00.000
        @ centerline

                 # 14
        N = 38714.3292
        E = 30519.5814
STA OFS  SETPC CLRPC CURVE
```

keystrokes:

2 0 +/- [OFS]

screen:

```
Sta 137+00.000
      @ 20.00 left

                 # 15
      N = 38707.0435
      E = 30500.9556
STA OFS  SETPC CLRPC CURVE
```

keystrokes:

2 0 [OFS]

screen:

```
Sta 137+00.000
      @ 20.00 right

                 # 16
      N = 38721.6149
      E = 30538.2071
STA OFS  SETPC CLRPC CURVE
```

keystrokes:

1 3 8 0 0 [STA] [OFS]

screen:

```
Sta 138+00.000
        @ centerline

                 # 17
        N = 38807.9847
        E = 30484.5431
STA OFS  SETPC CLRPC CURVE
```

keystrokes:

2 0 +/- [OFS]

screen:

```
Sta 138+00.000
      @ 20.00 left


              # 18
    N = 38801.3372
    E = 30465.6802
STA  OFS    SETPC CLRPC CURVE
```

keystrokes:

2 0 OFS

screen:

```
Sta 138+00.000
      @ 20.00 right


              # 19
    N = 38814.6322
    E = 30503.4061
STA  OFS    SETPC CLRPC CURVE
```

keystrokes:

1 3 9 0 0 STA

display:

Sta 138+83.270 S.C.

With the entrance spiral completed, you may go on to calculate the stations you need in the circular portion.

The program has already taken you to the SC, which is the beginning of the circular portion, so continue by stroking

CURVE

Answer the prompt for the central angle with input of the value for Δc. Remember, if this were a curve to the left, the central angle would be entered as a negative number by stroking +/- before CONT .

unequal spirals

The calculations for the exit spiral are exactly the same as for the entrance spiral. Because of the way the programming has been assembled, any spiral may be calculated as an individual curve or as part of a continuing alignment.

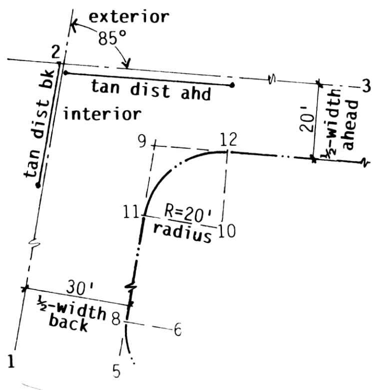
This also allows the spirals to have different lengths within the same system, be compounded, or even be reverse compounded, and still be calculated using this program. Each curve, regardless of whether it is circular or spiral can be calculated as it is reached on the alignment.

PRORA

This program is designed to solve an often encountered problem in urban surveys. A.L.T.A. morgtage surveys require the measurement of property lines for a block and the proration of the lot dimensions shown on the original tract map to match existing measured distances.

Since the work is usually done by measuring the centerline distances in the streets around the block, getting out of the street as soon as possible is important. This program lets you set the required property line intersection points ("prods"), or offsets to them while you are still set up at the street intersection.

Even better, it stores the coordinates of the points so that inversing from one to the other gives you the "measured" distance along the property line. The program also calculates the distances down the centerlines to be opposite the prods, if you want to set those points.



using the program

The program is used in a **clockwise** direction, and a point-code is input to establish the backsight bearing. The code is in the form **AAA** SPC **bbb**, where **AAA** is the point number at the instrument and **bbb** is the point number at the backsight.

Enter the program by stroking RUN PRORA

The program is self-prompting from there on.

Input the point number at the instrument, stroke SPC

Input the point number at the backsight and stroke CONT

INPUT DEFLECTION ∡

Input the exterior deflection angle and stroke CONT

or

If you would rather work with the interior angle, bring up the interior angle prompt. Without input, stroke CONT

RADIUS?

Input the radius (if there is no radius, input 0) and stroke CONT

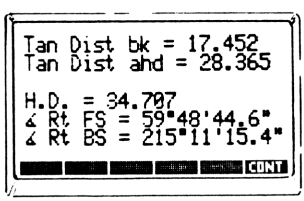
½-WIDTH BACK?

Input the distance from centerline to property line for the 'back' street, stroke CONT

½-WIDTH AHEAD?

Input the centerline to property line distance for the street 'ahead' and stroke CONT

The output will be similar to that shown to the right.

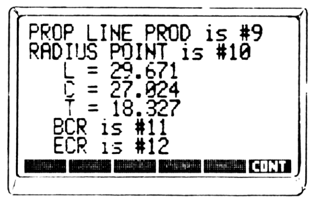


The 'H.D.' is the distance to the prod, and the angles are given from both sights, so you may use them from the direction you happen to be sighting, without having to take another shot.

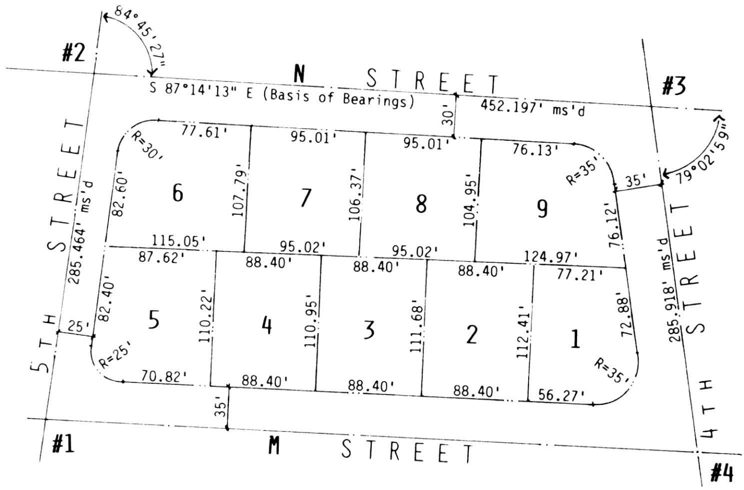
POINT NUMBER?

Input the point number you want to use for the property line intersection point. If there is a return curve, the program will assign the next **three consecutive** point numbers to the curve points. Stroke CONT

The remainder of the output tells you the point numbers assigned to the points and gives the curve data for the curve.



Offsets to these points could have been set, instead of the actual points by input of the half-widths **minus** the offset distance, and the radius **plus** the offset distance. A procedure is suggested below:



If lot #6 in the block shown above is the lot being measured, a procedure for the survey (with EDM) might be as follows:

1. With targets set at centerline points #1 and #3, occupy #2.
2. Measure the angle 1-2-3 and, with an assumed coordinate value for #3, traverse from #3 to #2 and to #1. This stores the coordinates for these points.
3. While you are still set up at #2 you can use the program **ALTA** to set the property line points at the northwest corner of lot 6.
4. Reset, with targets at #2 and #4, the instrument at #3. Measure the angle and distance to #4 and set a coordinate for #4 with the traverse program. Use the program to calculate the northeast corner points of lot 9, and set whatever prods you will need.

[illegible] CHDS

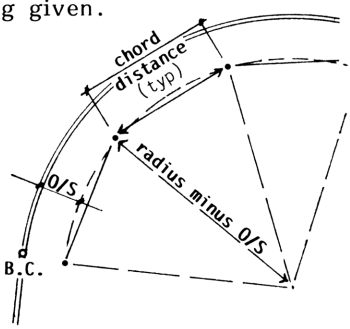
Primarily intended for layout of curb returns, this program prompts for input of the **radius**, **central angle** and **offset** being used. The radius that is input is the actual radius. The offset is assumed to be inside the curve (if it isn't, stroke +/- to change the offset sign).

options

There are two options offered by the program. The first is "**MAX**", where you input a maximum length between curb points, the other is "**EQ**", which divides the curve into **N** equal spaces. "**MAX**" also divides the curve equally, but not exceeding the spacing given.

output

The output is the chord distance to pull from point to point to double-tape the offset points in. The second distance is the radius minus the offset, pulled from the radius point. Using this method eliminates the need to set up an instrument at the radius point.



When you've finished, stroke [illegible]

[illegible]

This program uses the same prompts as "**CHDS**", but is used to set the offset hubs radially, from the radius point.

With the instrument at the radius point, sighting the B.C. of the return, the radial angle to the first offset point is calculated by the program. The distance is a constant, equal to the radius minus the offset.

As each offset point is set, stroking [illegible] brings up the angle to the next offset point.

Continue around the return until the last angle equals the central angle, and check into the E.C. offset hub.

When finished, stroke [illegible]

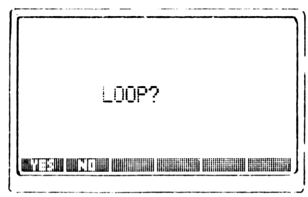
There are any number of reasons to run levels. This program tries to take all of them into account, with one exception; this is not designed as a cut-or-fill program.

It can be used to determine the cut or fill at any shot, but the subtraction is done manually. More on that later.

control

Control runs, or level loops designed to establish new benchmarks are one option. These can be run and adjusted, then output to a printer. A provision for reading 3-wire shots is also included.

The screen shown to the right will be displayed when you first enter the program.



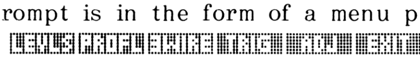
A "yes" or "no" answer is required, and sets up the parameters for the program.

If you plan to adjust the levels at the end of the level run, stroke YES

or

If you do not plan to adjust the loop at the end of the level run, stroke NO

The next prompt is in the form of a menu prompt bar:



If you are running "trig" levels, or want to use the 3-wire option, indicate it at this point, with just one keystroke.

If you are running TRIG levels, stroke TRIG

or

If you are going to use the 3-wire option, stroke 3WIRE

or

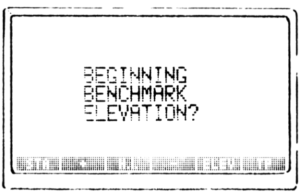
If you do not plan to use either of these options, go on to the next step:

Select the TYPE of level run you want.

If you want to run a loop with just turning points and benchmarks, stroke

or

If you are going to run levels that include sideshots to various points along the route, stroke



The screen prompt for the beginning elevation is next.

Input the elevation of the bench, and stroke
*

From this point on, use the 48 as an 'electronic field book', and stroke the appropriate keys as you go.

stationing

If you are running a loop for profiles, taking grid shots, as-building, etc., you need to identify the shot. In the case of a profile, this is easy, just input the station and stroke , before entering the rod reading.

On a grid, you have to be a little more inventive. The program expects a **number** at this point, and will output a **station**. To avoid confusion between a grid shot and a station shot, we have added a separate function.

If you label your grid 1 through n in one direction, and

*The **closing** benchmark is entered as a **shifted** turning point, by stroking ↰ .

This program defines "H.I." as an elevation, not as the distance (defined as a "plus rod") above the setup point.

1 through m in the other, a shot can be input as n.m, and will be output as nxm to indicate the grid point. The difference is that you have to stroke ↰ or ↱ before you stroke .

cuts and fills

When the shot's elevation is output (in profile mode) you can input the finish grade elevation and stroke the − key.

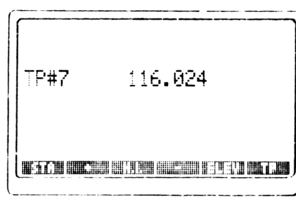
A positive answer indicates a CUT, and a negative answer indicates a FILL.

The cut or fill is not recorded by the program, but may be written onto the stake while the rodperson is still there.

turning points

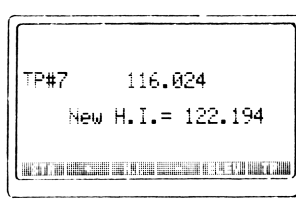
As with the stations, the turning point is identified **before** the rod reading. Stroke , then input the reading before stroking . The program will number the turning point and display it in place of a station, along with the elevation.

The elevation of the turning point is displayed.



After the level is moved to the next setup, input the rod reading at the turn and stroke .

The screen will change to show the new height of instrument elevation.



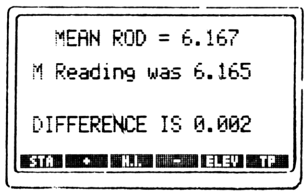
The current H.I. will then be shown at the top of the display for any shots taken prior to the next turn.

3-wire

The program works the same way for the three-wire levels, with the exception that the rod readings are input using this prompt bar:

The display will prompt for input of the rod readings, and they may be entered in any order (you can also change your mind about one of the readings, and re-input it). When the readings have been entered, stroke

The display will show the difference between the mean rod and the rod reading, and the 'field book' prompt bar.



If the shot is acceptable, input it by stroking or .

or

If the shot was not acceptable, return to the rod reading menu by stroking NXT

The 3-wire prompt bar will be redisplayed, and the shot may be taken over.

trig leveling

When an EDM is used for taking the shots, they need to be reduced by the program. The first prompt when you call for is for input of a distance factor, the difference between the elevation of the (topmounted) EDM and the theodolite's H.I. elevation.

This is used to correct for a difference in the rod reading **when the height of the instrument and EDM are not at the same elevation.**

If your EDM sends it's beam through the same lens as the line of sight, or uses a target on the rangepole to compensate, enter 0, and then stroke

or

If there is a difference between the H.I. of the EDM and that of the theodolite, input the difference (decimal feet) and stroke.

A second prompt, **NUMBER OF SIGHTINGS?**, will be displayed.

Enter either a 1, if you are planning to only take one shot to each target, and stroke CONT

or

Enter a 2 if you plan to take both a direct and reverse reading to each target. Stroke CONT

The 'Beginning Benchmark Elevation' is the elevation of the point you are starting at, and the distance to the scope (plus rod) above the point is input with [illegible]. The H.I. will be displayed at the top of the screen.

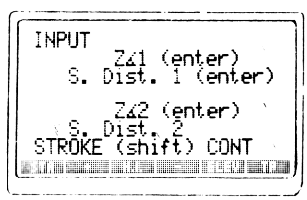
Indicate the next shot by input of a station and stroke [illegible]

or

Identify the shot as a turning point by stroking [illegible]

Instead of the prompt for NEXT SHOT, you get this screen (this one is for both direct and reverse shots).

It doesn't matter which shot is entered first, the direct or the reverse.



After input of the shot readings, and stroking CONT, the mean difference in elevation will be shown.

For trig levels, the rod reading is the distance from the pole's prism (or target if they are not the same). Input the rod reading and stroke either [illegible] or [illegible] to complete the shot.

cut and fill

If you selected [illegible], a cut or fill can be calculated when the elevation of the shot is output, as with other profile runs (see page 6-**3**). The screen prompt will have disappeared during the calculation, but input the next set of angles and distances and stroke CONT to continue.

closing

When you arrive at the end of the run, and take the final shot to the closing benchmark,

> Indicate that this is the closing benchmark by stroking ↰ [illegible], input the rod reading and stroke [illegible]
>
> Input the elevation of the bench (even if it is the same as the beginning bench) and stroke ELEV

The output will vary, depending on what kind of levels you've been running. The closing error will be shown, and if you were running 3-wire, the total distance traveled will be shown (doesn't include sideshots). A prompt, **OUTPUT?**, will also be in the display.

> If you want to download the level run at this point, stroke YES

or

> If you do not want to output the notes, stroke NO

adjustment

The run can only be adjusted if the response to the original prompt, **LOOP?** was yes.

> If you want to adjust the level run, stroke ADJ

The program will go through the loop and adjust all of the elevations. When it has completed the adjustment it will again prompt to allow output of the notes.

the file system

This group of programs is designed to handle **necessaries**. Like loading a file of coordinates prior to going to the field, or dumping the results to a printer when you get back. Stroke UTIL and then FILES , to bring up the 'files' menu

NEW EXIST RSIZE LIST COPY EXIT NXT PURGE

creating a new file

To create a NEW file, the name of the file must first be entered into level 1, prior to stroking NEW.

Stroke α and type in the filename. Stroke ENTER NEW

prompt: **FILE SIZE?**

Input the size you want for the file and stroke CONT

There is a pause while the file is created. The new file also becomes the **current** file.

existing file

Use the EXIST key to make an existing file the **current** file, at any time. The name of the file must be in level 1 prior to stroking this key.

Stroke α and type in the filename. Stroke ENTER EXIST

or

Use LIST to bring the name of the file you want to use to level 1 and stroke EXIST

The **current** file has been changed and the program will return to the main menu prompt bar.

resizing a file

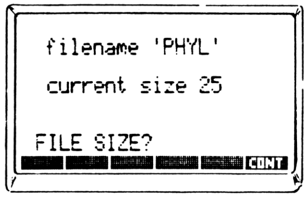
This program allows you to resize the **current** file. This may be done to save memory while a file is on 'standby', or to enlarge the file to allow more coordinate storage.

Stroke α and type in the filename. Stroke **ENTER** **RSIZE**

or

Use **LIST** to bring the name of the file you want to level 1, and then stroke **RSIZE**

Stroking the **RSIZE** key will bring up a prompt screen.



The current file size will be shown on the screen, as shown to the right, as a reminder and includes the prompt

FILE SIZE?

Input the NEW size for the file and stroke **CONT**

The file will be resized to the new size and becomes the **current** file.

listing the files

A function is provided to bring up a list of the existing files. You can scroll up or down through the list, using ▲ or ▼ to bring a particular filename to level 1 for use with the other options.

copying a file

There are several reasons why you might want to copy a file (make a duplicate file), and this program may be used to do so.

As an example, if you have a completed traverse and want to store a copy of the unadjusted file.

Put the filename into level 1, stroke **COPY**

prompt: **COPY AS (filename)?**

Put a new filename into level 1 and stroke **CONT**

A copy of the file is now stored under the new filename and the NEW file is the current file.

purging a file

The purging routine is in the second page of the FILES menu, stroke NXT to get to it.

This command totally deletes the referenced file when you use it. As a 'safety' feature, it will not work unless you first input either a filename or a block code.

Stroke α , type in the filename, stroke ENTER to put in the name of the **existing** file that you wish to delete, and then stroke PURGE

or

Use the LIST function to bring up the list of files, then use the ▼ or ▲ keys to bring the filename you want to Level 1. Stroke PURGE

With the file now deleted, select the file you will want to use next and make it the **current** file by stroking EXEC

purging a block of coordinates

To delete a group of coordinates from the **current** file, input the 'code' for the block of coordinates to be deleted and stroke PURGE

The code is in the form **AAA** SPC **bbb**, where **AAA** is the first point number and **bbb** is the last.

load

Stroking the LOAD key brings up a menu bar for input of coordinates to the **current** file: PT# N E ELEV CONT

Coordinates are stored under the point number indicated, and may contain 2- or 3-dimensions, with or without a descriptor.

Input the point number first, stroke PT#

Input the North-coordinate, stroke [N]

Input the East-coordinate, stroke [E]

optional:

Input the elevation, stroke [ELEV]

A descriptor may be entered at this point, as follows:

Stroke [↱] [-] [α] and type in the descriptor you want to use. Stroke [ENTER] and then [CONT]

dump

This program outputs your coordinates to a printer. The type of printer being used and the on/off status is set in [PARAM]. If the printer is OFF the program halts after output of each point.

Like '**PURGE**', the complete file will be output if you enter the filename, or just a given block of coordinates may be output by use of the 'block code', **AAA** [SPC] **bbb**.

edit

The [EDIT] function is designed to edit any **given point** in the current file. Input a point number and stroke [EDIT]

This function may also be used to duplicate points under a new point number. If a new point number is given, the original point is still stored, and a copy of the data is stored under the new number.

The prompt bar, [PT#] [N] [E] [ELEV] [DESC] [CONT], is used, input the new data for any portion of the coordinates and stroke the appropriate key. When finished with a particular point, stroke [CONT].

To edit the total file, stroke ['] and the [ ] key under the filename (in the secondary pages of the main menu) and then stroke [↱] [+/-].

The actual conversion of numbers in different systems is done using the softkey menu shown below.

[FT] [+] [-] [→CM] [ ] [ ]

When you first enter the program you are prompted for the output type with a menu bar: [FI16] [FI32] [DECF] [MET] [ ] [ ]

The response to this prompt can be any of

foot, inch & fraction
decimal feet
metric (centimeters)

The foot/inch/fraction selection has 2 settings, and the key to stroke depends on how much accuracy you want. The choices are nearest sixteenth or nearest thirtysecond.

from foot/inch/fraction

When you are converting foot/inch/fraction numbers to decimal feet, the [FT] key is used to input the **feet** portion. When stroked it brings up the secondary menu for input of the inches and fraction.

The remainder of the input is in the form **whole inches** [INCH] **or whole inches** [SPC] **fraction numerator** [ ], where [ ] denotes the denominator softkey.

For example, 6'-4" is input as

[6] [FT] [4] [INCH]

and 6'-4¼" would be input as

[6] [FT] [4] [SPC] [1] [4]

In either case, the output will be dependent on the parameters selected. Using 6'-4" as an example, if output is wanted in decimal feet, the display will show **6.333'**, if metric was selected, the display will show **193.0 cm**, and if foot/inch/fraction was selected, the calculator will display **6'-4"**.

Let's assume that the latter were the case. You may still change the number to centimeters (as a check, for instance) by stroking [→CM], and back again by stroking [ ]

It should also be noted that if foot/inch/fraction has been selected, the decimal foot equivalent is still in Level 1 on the stack. Stroke **ON** to clear the display.

To input a foot/inch/fraction number that is less than one foot, input 0 **FT** to bring up the fractional menu, and then continue.

If the number is less than an inch, input 0 **FT**, 0 **SPC**, the number, and stroke the denominator softkey.

metric input

Some architects, when dimensioning in the metric system, use millimeters, others centimeters. On the site plan the dimensions may even be in meters. The program uses **centimeters**, and displays to the nearest 0.1 centimeter.

As with money, the metric system can be manipulated to change the value. If the dimensions are given in millimeters, just move the decimal point 1 digit to the left. Don't do this with your checkbook.

Input the number and stroke **CM**. The display will show whatever the parameter selections decide.

decimal feet input

Use the **FT** softkey. Again, the display will depend on the parameters you have set. The decimal equivelent will remain in Level 1.

adding and subtracting

Using the **+** and **-** keys, you can add or subtract the currently displayed value to the previously displayed value, regardless of type of input. If your output parameter calls for the output to be in foot/inch/fraction, and you input 6.333 feet, with **FT**, the display will show **6-4"**.

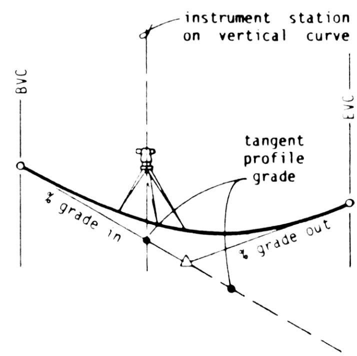
Now, input 20.3 centimeters with **CM**, and the display is **0'-8"**. Stroke **-** to subtract, and the display shows the answer, **5'-8"**.

Before starting with the user's instructions and the keystroke examples, this is a good place to look at how the program is set up.

There are several different ways to slope stake, and we've tried to make this program as convenient as possible for everyone.

The main program allows setting of a catch point at any convenient location, and the station at which the stakes are set is output. There is also a subroutine which lets you precalculate the angle and distance to a specific station, at any offset, to begin.

vertical grade



When the station that is occupied by the instrument is within a vertical curve, the elevation that is input as profile grade is the elevation of the **vertical tangent** at the instrument station.

It's a simple matter to calculate this elevation since it is just the extention of the back vertical tangent (even when your instrument is upstation of the PVI).

slope ratio

The slope ratio is carried as a constant, and is displayed each shot. If you want to change it, input the new number before stroking CONT. If it is the ratio you plan to use just stroke CONT.

half-width

Like the slope ratio, the half-width (distance from the centerline to the hinge point) is displayed each shot. If it is the width you want to use, stroke CONT If not, input the new number and then stroke CONT.

Both of the variables above may be changed at any time, during the shot input, allowing for widening the roadbed or flattening the slope as the cut or fill approaches a "daylight" area.

referencing

A routine has also been included which calculates the angle to turn and the distance to measure to set the reference stake. The amount of offset is input by the user and the angle and distance are calculated. A shot taken on the reference hub after it is set will output the elevation of the RP and the plus or minus to the slope stake.

setting specific stations

This program allows the flexibility of setting stakes wherever the terrain forms a high or low point, to best define the top or toe of slope. Since the program calculates the finished grade at any station you shoot, it isn't necessary to only set those points with elevations shown on the profile.

To stake at specific stations, at 25' or 50' intervals, you can set the station at an offset (determined as approximate catch point based on the drawings) for the first trial station, using STA and OFS.

After input of the station and offset the program will output the angle and distance to that point. Set a temporary point there and take a shot on it. The output will tell you which way, and how far, to the catch point.

Since the ground changes, it should be understood that this distance is where the catch point would be if the ground were flat.

the cut and fill factors

The actual point where the slope will end is seldom at the same elevation as profile grade. Use of these factors causes the adjustments to profile grade to be made automatically. These factors are discussed in more detail on page 8-6.

user's instructions

The program uses the menu for a prompt bar and the softkeys for input of the required data. Begin by stroking SLOPE , and the first of the prompt bars appears for information input relative to the instrument setup:

INSTRUMENT DATA

STA	OFS	HI	CURVE	TARG	CONT

Input the station which (or opposite which) is occupied by the instrument. Stroke

STA

If the instrument is on centerline, input 0. If the instrument is at an offset distance from centerline, input the offset distance. If the offset is left of centerline, stroke +/- before stroking

OFS

Input the **elevation** of the height of instrument. Stroke

HI

If the instrument station is within a curve, stroke CURVE to bring up the prompt bar for curve data input.

If there is a height difference (i.e.: top-mounted EDM) between the scope line of sight and the EDM line of sight that is NOT corrected by use of an offset target at the rod, input the distance (difference) and stroke TARG.

When all of the data pertaining to the instrument has been input, stroke

CONT

BACKSIGHT DATA

STA	OFS		CURVE		CONT

Input the backsight station, stroke

STA

Input the offset distance (0, if the backsight is on centerline). If the offset is to the left, stroke [+/-] before stroking

[O/S]

If the backsight is on a curve (but the instrument was not) stroke [CURVE] to bring up the curve data prompt bar.

CURVE DATA

[B.C.] [Δ] [R] [ ] [ENDC] [CONT]

This bar will appear, if neither the instrument nor backsight were on a curve, to give you a chance to input curve data for a curve in the work area.

Input the beginning station of the curve and stroke

[B.C.]

Input the central angle (if the curve is to the left, stroke [+/-]) and stroke

[Δ]

Input the radius of the curve and stroke

or [R]

Input an ending station and stroke

[ENDC]

After all of the needed data has been input, stroke

[CONT]

GRADE DATA

[P. GR] [% IN] [C.FAC] [F.FAC] [V.C.] [CONT]

Input the **finished grade elevation** at the instrument station. If the instrument is at a station which is located within a vertical curve, input the elevation of the **back tangent profile grade.** Stroke

[P. GR]

Input the percent of grade for the vertical alignment, If negative, **+/-**. Stroke **%IN**

Input the "cut factor" (see page 8-**6**). This is the difference between profile grade and the hinge-point of the cut section. If negative, **+/-**, then stroke **CFAC**

Input the "fill factor" (page 8-**6**). This is the difference in elevation between the finished (profile) grade and the hinge-point in a fill section. If negative, **+/-** and stroke **FFAC**

If there is a **vertical curve** in the work area, stroke **VC** to bring up the vertical prompt bar. When all data has been input, stroke **CONT**

vertical curve input

When there is a vertical curve in the work area, the information is input as shown below:

VERTICAL CURVE DATA

BVC **L** **%OUT** **CONT**

Input the station at the beginning of the vertical curve. In the case of a grade break, instead of a curve, input the station at the grade break. Stroke **BVC**

Input the length of the vertical curve. If this is a grade break instead of a curve, input 0, then stroke **L**

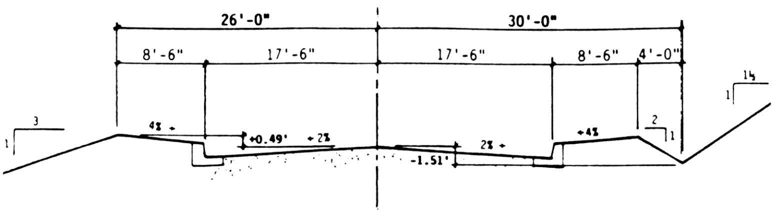
Input the percent of grade leaving the vertical curve. If negative, **+/-** before stroking **%OUT**

stroke **CONT** to continue.

cut and fill factors

These factors are simply the difference between the finish profile grade and the hinge point you are staking to. If, for instance, you were staking to the back of a bench in a cut section, this could be a difference of 20 or more feet.

If you look at the typical sections shown below, the hinge point of the cut would be at the flowline of the ditch, while the fill section hinge point is located at the edge of the shoulder.



TYPICAL FILL SECTION ℄ TYPICAL CUT SECTION

The **factor** will be the sum of the differences. We have 17.5 feet at -2%, +0.5' for the curb, and 8.5' at +4% on both sides; (17.5)(-.02) + 0.5 + (8.5)(.04) = +0.49. This is the **fill factor**, and is the difference between the profile grade and the edge of the shoulder on both sides.

In the cut section, we have the added ditch, 4 feet at a 2:1 slope, which is -2'. This gives a **cut factor** of -1.51.

This typical section will be used for the keystroke example, so the factors in the example will be as shown.

staking menus

There are two menus used in the slope-staking process. The first is for input of the measurements. It also has keys that let you tell the program whether you are staking a cut or a fill: HZ ZA SD C F ADD

If you have been staking in a cut section, but want to switch to a fill section, prior to data input, stroke F

If you have been staking a fill, and want to switch to a cut section, prior to data input, stroke C

The second menu, which is displayed with the output information (or may be reached by stroking NXT), allows entry of new factors if the typical section changes from the original design:

STA O/S USE NEWC NEWF REF

To change the cut factor, input the new factor and stroke NEWC

to change the fill factor, input the new factor and stroke NEWF

The program will calculate the catch points based on the new input, until you change it again. The new factor also becomes the **current** factor. If you have just input a new fill factor, but are currently staking a cut, stroke C before beginning input of the shot.

setting a specific station/offset

The second menu also has keys for setting a specific station and offset as a starting point.

Input the station you want to set, stroke STA

Input the offset, stroke O/S

The angle and distance to the selected station/offset will be displayed, and setting this point gives you a starting point for that station's slope stake.

shot data input

INPUT SHOT

HZ ZA SD C F ROD

The first time this prompt appears, or at any time when you want to change the hinge point (from cut to fill or from fill to cut), select the appropriate factor

If staking a cut, stroke [C]

If staking a fill, stroke [F]

INPUT SHOT

[H∠] [Z∠] [S.D.] [C] [F] [ROD]

Input the horizontal angle [H∠]

Input the zenith angle [Z∠]

Input the measured slope distance [S.D.]

Because, in slope staking, the rod is always a minus, the program will indicate that; you need not change the sign. Once the rod has been input you do not have to input it again, unless it changes. Stroking [ROD] will cause the program to repeat the last rod reading for this shot.

Input the rod reading, stroke [ROD]

The next two prompts are for the "half-width" and the slope ratio. Since they may both vary, the value about to be used for the calculation is displayed, and may be changed at this time by input of a new value, prior to stroking [CONT].

W/2 = 0.00

[ ] [ ] [ ] [ ] [ ] [CONT]

Input the correct half-width value, if it is different than the value which is displayed [CONT]

Slope Ratio = 0:1

[ ] [ ] [ ] [ ] [ ] [CONT]

Input the correct slope ratio value if different from the value displayed CONT

At this point the display will show the distance to the actual catch point. If the distance is considered to be within tolerance, stroke USE. If it is not, have the rodman move to a new position (based on the displayed information), return to the other menu by stroking NXT, and begin with input of the data from the new shot.

to set a reference stake

Once you have reached the catch point, stroking USE will display the information for making out the slope stake.

Input the distance you want to use, from the slope stake to the reference stake. If it is left of centerline, +/-, then stroke REF

The distance and angle to set the reference stake will be displayed. Set the point and take a shot on it. The data is input in the same manner as any other shot, but the reference stake information will be output.

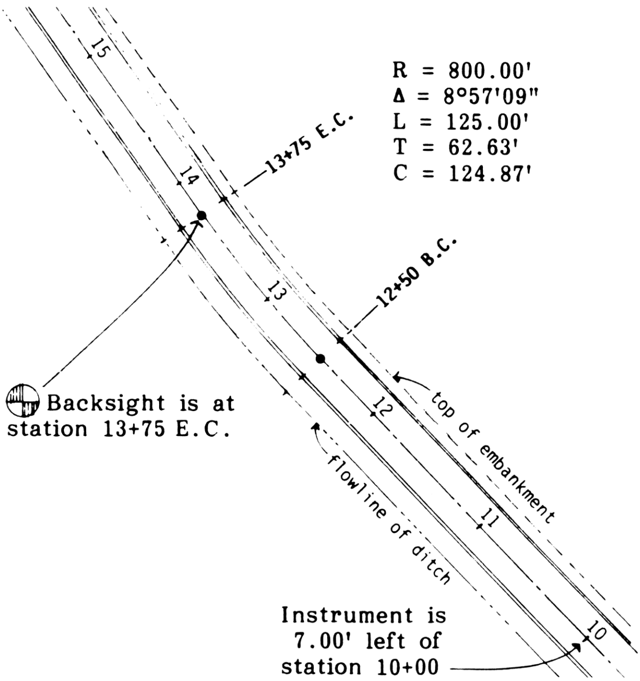
program limits

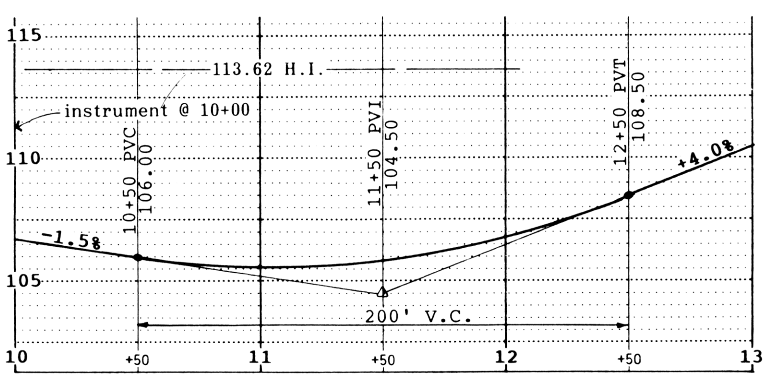
There are no limits to the widths of a cut or fill staked with this program. In bench cuts, the cut factor can move the hinge point elevation to any needed height, and the half-width can account for the total offset to the hinge point.

The program **is** limited to working only in one horizontal and one vertical curve at a time. Much field time may be saved by pre-planning the setup points for maximum coverage of each area to be staked.

keystroke example

In addition to the typical section on page 8-**6**, the example uses the horizontal and vertical alignments shown on the next page. After the keystroke example we have included some more 'practice shots' that you can try.


R = 800.00'
Δ = 8°57'09"
L = 125.00'
T = 62.63'
C = 124.87'
13+75 E.C.
12+50 B.C.
15
14
13
12
11
10
top of embankment
flowline of ditch
Backsight is at station 13+75 E.C.
Instrument is 7.00' left of station 10+00


115
110
105
113.62 H.I.
instrument @ 10+00
10+50 PVC
106.00
11+50 PVI
104.50
12+50 PVT
108.50
-1.5%
+4.0%
200' V.C.
10
+50
11
+50
12
+50
13

Access the program and bring up the first of the data input prompts by stroking

[SLOPE]

INSTRUMENT DATA

[STA] [O/S] [H.I.] [CURVE] [TARG] [CONT]

keystrokes:

[1] [0] [0] [0] [STA]

[7] [+/-] [O/S]

[1] [1] [3] [·] [6] [2]

[H.I.] [CONT]

BACKSIGHT DATA

[STA] [O/S] [ ] [CURVE] [ ] [CONT]

keystrokes:

[1] [3] [7] [5] [STA]

[0] [O/S]

[CURVE]

CURVE DATA

[B.C.] [Δ] [R] [ ] [ENDS] [CONT]

keystrokes:

[1] [2] [5] [0] [B.C.]

[8] [·] [5] [7] [0] [9]

[Δ]

[8] [0] [0] [R]

[CONT]

BACKSIGHT DATA

[STA] [O/S] [ ] [CURVE] [ ] [CONT]

keystroke:

[CONT]

PROFILE DATA

[R.GR] [%IN] [C.FAC] [F.FAC] [V.C.] [CONT]

keystrokes:

[1] [0] [6] [·] [7] [5]

[R.GR]

[1] [·] [5] [+/-] [%IN]

[1] [·] [5] [1] [+/-]

[C.FAC]

[·] [4] [9] [F.FAC]

[V.C.]

VERTICAL CURVE

[BVC] [L] [%OUT] [ ] [ ] [CONT]

keystrokes:

[1] [0] [5] [0] [BVC]

[2] [0] [0] [L]

[4] [%OUT]

[CONT]

This completes all of the required data input.

screen:

```
INPUT SHOT
 H∠  Z∠  S.D.  C  F  ROD
```

Prior to the first shot input, we need to say whether it's a cut or fill section. We'll assume that the first shot will be fill, and stroke

[F]

Let's try a shot using the following data:

horiz. angle = 13°10'20"
zenith angle = 92°05'15"
slope distance = 187.40'
the rod reading is 5.00'

keystrokes:

[1] [3] [.] [1] [0] [2]

[H∠]

[9] [2] [.] [0] [5] [1] [5]

[Z∠]

[1] [8] [7] [.] [4]

[SD]

[5] [ROD]

W/2 = 0.00

[ ] [ ] [ ] [ ] [ ] [CONT]

keystrokes:

[2] [6] [CONT]

Slope Ratio = 0:1

[ ] [ ] [ ] [ ] [ ] [CONT]

keystrokes:

[3] [CONT]

display: TRY 2.55' RIGHT

You need to have the rodman go about 2½ feet to the right. We take a shot there, with the following data:

horiz. angle = 13°57'00"
zenith angle = 92°42'00"
slope distance = 187.90'
the rod reading is 5.00'

Input this new data. Note that the ½ width and slope ratio now show the correct information, as last input.

keystrokes:

[1] [3] [.] [5] [7]

[H∠]

[9] [2] [.] [4] [2]

[Z∠]

[1] [8] [7] [.] [9]

[SD]

[ROD]

W/2 = 26.00

[ ] [ ] [ ] [ ] [ ] [CONT]

keystroke:

[CONT]

Slope Ratio = 3:1

[ ] [ ] [ ] [ ] [ ] [CONT]

keystroke:

[CONT]

display: TRY 0.94' LEFT

The ground slope has made a difference, and you need to go left about a foot, at the same elevation, for a catch point. On this shot we get:

horiz. angle = 13°30'25"
zenith angle = 92°27'00"
slope distance = 187.70'
the rod reading is 5.00'

keystrokes:

1 3 . 3 0 2 5 [H∠]

9 2 . 2 7 [Z∠]

1 8 7 . 7 [S.D.]

[ROD]

W/2 = 26.00

[CONT]

keystroke: [CONT]

Slope Ratio = 3:1

[CONT]

keystroke: [CONT]

display: `TRY 0.09' RIGHT`

This is close enough to use for the catch point. Stroke NXT [USER]

screen:

```
Sta 11+80.200
@ 44.9' RIGHT
CP El = 100.6

  FILL 6.3 @ 18.8

H.P. El = 106.87
STA  D/S  USER  NEWC  NEWF  REF
```

Let's set the reference point at 15'

keystrokes:

1 5 [REF]

screen:

```
to set RP

DIST = 192.22
∠ = 17°48'26.7"

INPUT SHOT
H∠  Z∠  S.D.  C  F  ROD
```

INPUT SHOT

H∠ Z∠ S.D. C F ROD

Turn the angle shown and set the reference point at the distance given. After the point is set, take a shot on it to get the elevation. We'll use the following data:

horiz. angle = 17°48'30"
zenith angle = 93°06'15"
slope distance = 192.20'
the rod reading is 5.00'

keystrokes:

1 7 . 4 8 3 [H∠]

9 3 . 0 6 1 5 [Z∠]

1 9 2 . 2 [S.D.]

[ROD]

screen:

```
RP Elev = 98.2

+2.4 @ 15.0 to S.S.

INPUT SHOT
H∠  Z∠  S.D.  C  F  ROD
```

Let's try one on the cut side. First, change to the cut factor by stroking

[C]

We'll use the following data:

horiz. angle = 348°23'00"
zenith angle = 89°19'50"
slope distance = 201.90'
the rod reading is 5.00'

keystrokes:

[3] [4] [8] [·] [2] [3]

[H∢]

[8] [9] [·] [1] [9] [5]

[Z∢]

[2] [0] [1] [·] [9]

[SD]

[ROD]

W/2 = 26.00

[ ] [ ] [ ] [ ] [ ] [CONT]

keystrokes:

[3] [0] [CONT]

Slope Ratio = 3.00

[ ] [ ] [ ] [ ] [ ] [CONT]

keystrokes:

[1] [·] [5] [CONT]

display: TRY 0.29' RIGHT

Let's go a little to the right and try another shot. We get:

horiz. angle = 348°25'40"
zenith angle = 89°17'20"
slope distance = 202.30'
the rod reading is 5.00'

keystrokes:

[3] [4] [8] [·] [2] [5] [4]

[H∢]

[8] [9] [·] [1] [7] [2]

[Z∢]

[2] [0] [2] [·] [3]

[SD]

[ROD]

W/2 = 30.00

[ ] [ ] [ ] [ ] [ ] [CONT]

keystroke:

[CONT]

Slope ratio = 1.50:1

[ ] [ ] [ ] [ ] [ ] [CONT]

keystroke:

[CONT]

display: TRY 0.02' LEFT

keystrokes:

[NXT] [USE]

screen:

```
Sta 11+99.787
@ 38.7' LEFT
CP El = 111.1

  CUT 5.8 @ 8.7

H.P. El = 105.33
STA  O/S  USE  NEWC NEWF REF
```

keystrokes:

[1] [0] [+/-] [REF]

(we're going left)

screen:

```
to set RP

DIST = 204.09
∡ = 345°39'14.3

INPUT SHOT
H∡ Z∡ S.D. C F ROD
```

keystrokes:

3 4 5 . 3 9 H∡

8 7 . 3 2 Z∡

2 0 4 . 1 S.D.

ROD

After setting the RP, we get a shot of:

horiz. angle = 345°39'00"
zenith angle = 87°32'00"
slope distance = 204.10'
the rod reading is 5.00'

screen:

```
RP Elev = 117.4

-6.3 @ 10.0 to S.S.

H∡ Z∡ S.D. C F ROD
```

extra practice

The shots listed below will give you some additional practice with the keystrokes. The shots are to the catch points, still using the same alignment. Remember to change the cut or fill factor (if necessary) before input of the shot. Forgetting to do so will result in an error at the catch point.

1. **Cut section**, set the RP @ 15'. Use a zenith angle of 88°12'20" for the RP shot.

horiz. angle = 354°40'50"
zenith angle = 88°29'25"
slope distance = 400.70'
the rod reading is 5.00'

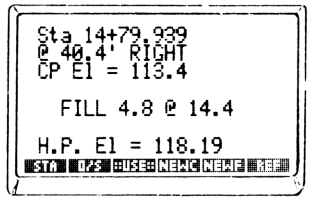
```
to set RP

DIST = 403.87
∡ = 352°35'47.1
```

```
RP Elev = 121.3

-2.1 @ 15.0 to S.S.
```

```
Sta 13+94.686
@ 39.6' LEFT
CP El = 119.2

  CUT 6.4 @ 9.6

H.P. El = 112.78
H∡ Z∡ S.D. C F ROD
```

2. **Fill section,** set the RP @ 15'. Use a zenith angle of 89°40'30" for the RP shot.

horiz. angle = 6°13'55"
zenith angle = 89°25'40"
slope distance = 477.50'
the rod reading is 5.00'



to set RP

DIST = 477.67
∡ = 8°01'53.3

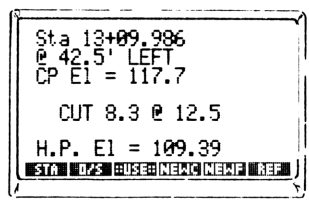
RP Elev = 111.3

+2.1 @ 15.0 to S.S.

---

3. **Cut section,** set the RP @ 15'. Use a zenith angle of 87°52'10" for the RP shot.

horiz. angle = 351°23'30"
zenith angle = 88°21'00"
slope distance = 315.00'
the rod reading is 5.00'



to set RP

DIST = 317.39
∡ = 348°42'48.1

RP Elev = 120.4

-2.7 @ 15.0 to S.S.

The two most likely sources of error when using this program are forgetting to change the factor when going from cut to fill (or fill to cut) and not checking that the half-width and slope ratio are correct for the shot being taken.

This program stores the raw data, slope stake information and reference point information from each of the catch points actually set. If you plan to keep a record of the information, before beginning the next group of stakes the information should be output to a printer.

To print out the run, first make sure that the calculator is set to 'PRINT ON'. Stroke ↑ and then SLOPE.

NXT DATA

This program is designed for **topo** pickup in the field. The Interface Pak, when published will work with this (and other) programs in Version 3.0 to allow hookup directly to most total stations.

The file that contains the control for the survey should be the **current file** before you begin. This allows the program to use existing points for the setup before switching over to the data collection files. If you are not using existing points, input them as NEW, but remember that these points will reside in the **current** file.

Access the program by stroking DATA

FILENAME? Select size

Input of the filename and selection of the file size are done at the same time.

Stroke α, type in the filename for the topo file. Stroke ENTER and then the softkey which corresponds to the size of the file you want to create.

The next prompt allows the user to select the point # to be used for the first record at this setup. The file will contain blank registers up to this point (default is 1). The program will set up two parallel files, one for the finished data and one for the raw data that is input.

INSTRUMENT @
POINT NUMBER?
EXIST | | NEW | | |

Input the point number of the instrument point and stroke EXIST

or

Input the point number, stroke NEW

INPUT COORDINATES
PT# | N | E | ELEV | | CONT

Stroke PT#

Input the north coordinate, stroke N

Input the east coordinate, stroke

[E] [CONT]

Input of the elevation and descriptor are optional.

H.I.?

[ ] [ ] [ ] [ ] [ ] [CONT]

This prompt requires the elevation of the scope, input the H.I. and stroke [CONT]

BACKSIGHT
POINT NUMBER?

[EXST] [ ] [NEW] [ ] [ ] [ ]

Input the point number of the backsight point and stroke [EXST]

or

Stroke [NEW] and input the new point (see NEW, above).

The parallel files are set up by the program, and it switches over to use them. The locations of the instrument and backsight are inserted as record numbers 1 and 2, in both files.

In record number 1, the height of instrument occupies the field normally held by the elevation. A descriptor is added to each of these records which gives the filename and point number which were used for the points.

The prompts for shot input will begin with shot number **3**, since 1 and 2 are the setup information.

INPUT SHOT #3*

[H<] [Z<] [S.D.] [ROD] [ ] [CONT]

Input the horizontal angle (D.ms) and stroke [H<]

Input the zenith angle (D.ms) and stroke [Z<]

* This initial shot number will vary, depending on the first record number set. It will always be the **third** record at each setup. This feature is handy for use with multiple setups for the same job, allowing the merging of the files without changing the record numbers of the shots taken in the field.

Input the slope distance, stroke

[S.D.]

ROD = 0.00

[H∠] [Z∠] [S.D.] [ROD] [ ] [CONT]

The current rod reading is always shown as a reminder (the first time, 0.00). Check with the rodperson to see if it is still current and save some editing time later.

To change the rod reading, input the correct value and stroke

[ROD] [CONT]

or

If it is the correct rod, stroke

[CONT]

LABEL SHOT #3

[RPT] [DESC] [ ] [ ] [ ] [ ]

There are two options for descriptor input:

To input the descriptor, stroke

[↱] [−] [α]

Type in the description, up to 28 characters, and stroke

[ENTER] [DESC]

or

If the description of the current shot is the **same** as the last shot, take advantage of the 'repeat' function and stroke

[RPT]

INPUT SHOT #4

[H∠] [Z∠] [S.D.] [ROD] [ ] [CONT] . . .

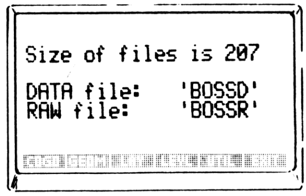
Repeat the steps for input of the shots and descriptions until you have completed the work from this topo point.

When finished, to exit the program, stroke

[NXT] [EXIT]

The parallel data files are resized to conserve storage registers.

The display will show the final file size, and the names of the **two** files that have been created. The suffix, D, indicates the Data file, R the raw file.



The original file (control file) has been made the **current** file.

The new files will be inserted into the file list automatically and may be accessed with NXT, from the main menu, and may be edited with VISIT (stroke ↱ +/-), or as individual points with EDIT, after making the file to be edited current.

The most common cause of required editing is through lack of communication regarding the rod reading. If this is the case, BOTH files must be edited.

NOTE: With this system, each setup requires a new file, even though the setups are all part of the same job. In the example, 'BOSS' is the filename. The filename for the second setup could be 'BOSS2'

The files may later be combined in your PC, using whatever file management system you have. See pages 2-**4** and 2-**5** for information on transfer and conversion of the files.

Caution: The symbol, ", may not be used as part of a descriptor. If you want to indicate inches, us "in" instead of the symbol, or let it be implied, as in 10 OAK TREE.

The raw data is stored as

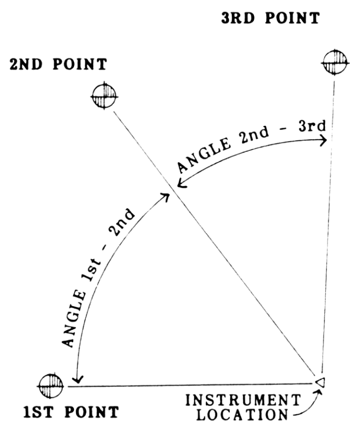
Horizontal angle Slope distance Zenith angle Rod

with the angles in Degree, minute and seconds. The stored finish data is in the form

North-coord East-coord Elevation Description

Resection is used to determine the position of the instrument (relative to known points). Because some of the programs use station/offset instead of coordinates for location, the resection program will work either way.

By backsighting any **three** known points, angles can be turned from the first point to the second, and from the second point to the third.



The backsighting should be done, and the angles should be input to the program, in a clockwise direction, so begin with the left-most point.

The angles are entered first. The points may be input as coordinates, point numbers, or stations and offsets.

Access the program by stroking

Input the two angles, using the angle keys, then

for coordinate input

Input each pair of coordinates as

N-Coord **SPC** **E-Coord**

The program continues automatically after the input of the third point.

point number input

Just input the three point numbers as

point number

POINT NUMBER?

Input the point number you want to use for storing the instrument point, stroke **CONT**

The rest of the solution is automatic, and the stored point can be used with '**SPRAY**' to set any additional layout points.

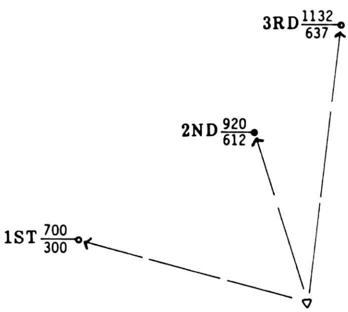
input with station/offset

This is sometimes the best way to go if you need the setup point for working with the EDM Slope Staking program.

Input each of the station/offsets as

station **STA** **offset** **OFS**

If you are sighting centerline stations, input 0 for the offset. The output will be the station and offset of the instrument.



We will use the points shown in the illustration to the left for a keystroke example. The three points may be assumed to be known, but not in storage.

If you want to try the input by point number, you can pre-store the stations with '**LOAD**' prior to beginning.

In this example the angle from the 1st point to the 2nd point was measured as 37°57'07", and the angle from the 2nd point to the 3rd as 17°00'47".

Access the program and input the angles first. Next, either by pre-storage or by coordinate input, put in the three points.

```
Instrument @

     N = 626.0002
     E = 907.0010
```

Try one with the station/offset option using angles of 58°09'57" and 24°58'16", sighting centerline stations 10+00, 12+50 and 13+50. You should get an answer of Station 12+17.32 at 192.17' right.

The following are functions or systems contained in the software from Surveying Packet #1 that may be used in your own programs just as if they were HP48 commands.

Angular input **D.ms** indicates that the angle is in Degrees, minutes and seconds, while **D.dd** indicates that the number is in Decimal degrees.

ANGLE FUNCTIONS

AAθ

ACTION: Labels the azimuth as a text string ready for storage in the display or for printout.

REQUIREMENTS: Azimuth (D.ms) in Level 1

RESULT: Returns labeled text string to Level 1, retains original value in Level 2

AA1

ACTION: Changes bearing to azimuth and returns labeled text string ready for storage in the display or for printout

REQUIREMENTS: Bearing (D.ms) in level 2, Quad code in Level 1

RESULT: Returns labeled text string to Level 1, azimuth (D,ms) to Level 2

BB1

ACTION: Changes azimuth to bearing as text string ready for storage in the display or for printout.

REQUIREMENTS: Azimuth (D.ms) in Level 1

RESULT: Returns bearing text string (D.ms) to Level 1, retains original azimuth value in Level 2

DCK

ACTION: (D.ms) Checks for negative angles and changes negative angles to positive.

REQUIREMENTS: Angle (D.ms) in Level 1

RESULT: If angle<0, returns angle value between 0°-360° to Level 1

DCK1

ACTION: (D.dd) Checks for negative angles and changes negative angles to positive.

REQUIREMENTS: Angle (D.dd) in Level 1

RESULT: If angle<0, returns angle value between 0°-360° to Level 1

DM

ACTION: Changes angle (D.ms) to labeled text string.

REQUIREMENTS: Angle (D.ms) in Level 1

RESULT: Text string replaces angle in Level 1

COORDINATE FUNCTIONS

General information: The coordinate storage system stores 4-dimensional coordinate pairs into its storage registers at the POINT NUMBER indicated by a number stored in the variable, **PT**. The functions always work on the **current** file.

CUP

ACTION: Actually a coordinate display system supplement. Displays the point number, northing and easting.

REQUIREMENTS: North coordinate in level 2, East coordinate in Level 1 and the point number in the variable PT

RESULT: The point number and coordinates are stored into the display system, which is then activated. The north coordinate remains in Level 2, and the east coordinate remains in Level 1

DUMP

ACTION: Outputs current file or specified block of coordinates.

REQUIREMENTS: If Flag 55 is set, output is directed to printer, If Flag 55 is clear, output is to display (single-step)

For FILE output, filename should be in Level 1, for block output, first point# in Level 2, last point# in Level 1

RESULT: Coordinate listing from current file is output as point number, northing and easting.

IN

ACTION: Stores a coordinate pair into the register specified by the point number in variable PT.

REQUIREMENTS: North coordinate in Level 2, East coordinate in Level 1

RESULT: Coordinates are stored into the specified register as a 2-dimensional vector and removed from the stack

---

LOAD

ACTION: Used to manually input coordinates.

REQUIREMENTS: none

RESULT: Calls up inter-active menu for coordinate input

OUT

ACTION: Recalls a coordinate pair as specified by the variable PT

REQUIREMENTS: The point number stored into the variable PT

RESULT: Returns the coordinate pair; north coordinate is in Level 2, east coordinate in Level 1

PINN

ACTION: Stores coordinate pairs as specified by point number in Level 1.

REQUIREMENTS: North coordinate in Level 3, east coordinate in Level 2 and point number in Level 1

RESULT: Stores coordinates into current file and clears display

---

POUT

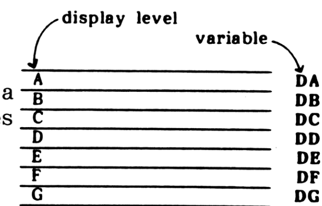
ACTION: Recalls coordinate pair as specified by point number.

REQUIREMENTS: Point number in Level 1

RESULT: Recalls coordinate pair to stack, with north coordinate in Level 2 and east coordinate in Level 1. Point number remains in variable PT

DISPLAY SYSTEM

Text strings are stored into a variable 'Dx', where x denotes the display line to be used.



C1

ACTION: Clears line A of the display.

REQUIREMENTS: none

RESULT: Blank line at A in the display

C7

ACTION: Clears display

REQUIREMENTS: none

RESULT: Totally clears display system

COO

ACTION: Right-justifies text string for display output.

REQUIREMENTS: Text string in Level 1

RESULT: Adds leading blank spaces to text string so the it may be displayed right-justified

---

GENERAL USE FUNCTIONS

D7

ACTION: Activates display.

REQUIREMENTS: Depending on usage, may require HALT to prevent continuation into following program steps

RESULT: Display shows information stored in variables '**DA**' through '**DG**' in display lines A through G

PTC

ACTION: Displays prompt in line G and menu with continue stroke.

REQUIREMENTS: Text string for the prompt in Level 1

RESULT: Displays prompt string in line G and halts for input. After CONT stroke, resets the previous menu and continues with any following program steps in program calling it

---

PR

ACTION: Printer control. Use in programs instead of HP48 function 'PR1'.

REQUIREMENTS: If Flag 55 is set, output is directed to the printer. If Flag 55 is clear, calculator works at zero delay.

RESULT: Programs run faster by only setting a delay when necessary for printout

---

P→R

ACTION: Polar to rectangular conversion.

REQUIREMENTS: Azimuth (D.dd) in Level 2, distance in Level 1

RESULT: Departure in Level 2, Latitude in Level 1

R→P

ACTION: Rectangular to polar conversion.

REQUIREMENTS: Departure in Level 2, Latitude in Level 1

RESULT: Returns azimuth (D.dd) to Level 2, distance to Level 1

TM

ACTION: Displays temporary "continue" menu when input is needed.

REQUIREMENTS: none

RESULT: Provides CONT stroke for use in programming

STORAGE SYSTEM

General information: There are 50 storage registers (0 through 49) which may be used for scratch or storage of data during a program. The following functions work with this set of registers.

CLRG

ACTION: Clears all of the storage registers.

REQUIREMENTS: none

RESULT: All registers will contain zeros

RX

ACTION: Recalls object from storage register.

REQUIREMENTS: Register number in Level 1

RESULT: Object in specified register is returned to Level 1

SX

ACTION: Stores object into specified register.

REQUIREMENTS: Object to be stored in Level 2, storage register number in Level 1

RESULT: Returns object from specified storage register to Level 1 after storing it

---

MANUAL ONLY
List $40.00 U.S.

ISBN 0-944889-18-2