COGO-10/20
User's Manual

Order No. AA-5510A-TK

July 1978

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard. massachusetts

First Printing, July 1978

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10

CONTENTS

FIGURES

PREFACE

COGO-10/20 is a FORTRAN language program designed to solve plane coordinate geometry problems on DECsystem-10 and DECSYSTEM-20 computers. COGO-10/20 runs under the TOPS-10 and TOPS-20 Operating Systems.

This manual explains the COGO-10/20 commands and tells the user how to create and enter COGO input. The COGO input language was specifically designed for the civil engineer and surveyor. The user is expected to have a basic knowledge of geometry. No previous experience with computers is required.

The usual units of measurement in COGO are feet for length and degrees, minutes, and seconds for angles. If meters or some other unit of length is used throughout the problem, the results are still accurate, but output labeled "feet" is actually meters. Area results given in acres are only accurate if lengths have been entered in feet.

The system commands necessary to install COGO-10/20 are given in the COGO-10/20 Installation Guide, DEC order no. AA-5511A-TK.

Chapter 1 describes COGO in general terms; Chapter 2 lists conventions and definitions of terms needed to use the COGO command language properly; Chapter 3 contains the explanation of each COGO command; and Chapter 4 presents sample problems that illustrate how a related series of COGO commands can be used to solve practical problems.

Appendix A gives a summary of COGO-10/20 commands; Appendix B lists the COGO error messages; and Appendix C provides information on how to generate files. A glossary is included following the last appendix.

Throughout this manual, the following documentation conventions are used:

UPPERCASE	In all representations of user input, uppercase letters denote information that must be entered exactly as shown.
lowercase	In all representations of user input, lowercase letters denote variable information determined by the user.
[ ]	Square brackets enclose optional information that can be omitted at the user's discretion. If any optional data is specified, all preceding optional data must be specified also. The brackets are not part of the actual input.

red print User input is printed in red to differentiate from computer output.

RET All command and input strings are terminated by pressing the RETURN key, unless otherwise specified.

Users who wish to become more familiar with the TOPS-10 and the TOPS-20 Operating Systems can find further information in the manuals listed below:

FORTRAN-10 Programmer's Reference Manual (V5)	AA-0944E-TB
LINK-10 Programmer's Reference Manual (V2B)	DEC-10-ULKMA-B-D
DECsystem-10 Operating System Commands Manual (V6.03)	AA-0916C-TB
DECSYSTEM-20 FORTRAN Reference Manual (V5)	AA-4158B-TM
DECSYSTEM-20 LINK Reference Manual (V3)	DEC-20-ULRMA-A-D
DECSYSTEM-20 User's Guide (V2)	AA-4179B-TM

CHAPTER 1

INTRODUCTION

COGO is designed to solve problems in plane coordinate geometry. Complex problems can be solved by starting from known points and progressing through a series of simple COGO commands; the required series of COGO commands are similar in practice to the user's step-by-step solution to the problem if solved by hand.

COGO commands input and determine:

- points
- lines
- areas
- tangents
- circular and spiral horizontal curves

Major applications for COGO include:

- land surveying
- highway design
- right-of-way surveys
- bridge geometry
- subdivision work

1.1 OVERVIEW AND HIGHLIGHTS

COGO can be run in an interactive mode or in a batch mode.

The I/O mode (input/output mode) can be changed during a run. The input can come from a card reader, a file stored on a magnetic disk, or a terminal keyboard during the same run. The output can be sent to a terminal or line printer.

COGO offers a user much flexibility in the input format. Input to COGO consists of commands and comments. A comment can be on a COGO input record by itself or after the last datum of any COGO command. A COGO command is made up of a command name and perhaps some data. Each command can be abbreviated in almost any reasonable way. Distances, angles, azimuths, and bearings can be entered as the sum or difference of several terms. The input is in free format, meaning that the command names (or their abbreviations) and the data do not need to be in certain positions of an input record.

Errors in the input are handled to make the most use of the input already read and to be helpful to the user in identifying the error.

Error messages are printed near the left margin of terminal output for speed of printing, but end in column 72 of line printer output for ease of reading.

In practical problems a user is not dealing simply with points but with groups of points that define such things as property boundaries, streets, right-of-ways, traverses, and so forth. COGO-10/20 allows the user to define and refer to lists of related points. These lists are called FIGURES. Up to 9999 figures can be defined and referred to by number to compute intersections, parallel figures, areas, etc. The ability to operate on an entire group of points simply by specifying a figure number greatly reduces the amount of human effort required to do a job.

The coordinates and figures are stored by the computer in an area called a COGO table. This table is saved by the computer so that the COGO user can refer to the points and figures in future runs. Each table has a variable length up to 9999 points. The number of tables is limited only by the amount of disk space available. People working on several jobs can use separate COGO tables for the different jobs.

Several parameters can be easily set by a COGO site. Two examples of these parameters are the maximum number of errors COGO will accept before terminating, and the device used for output.

New commands can be added easily. COGO-10/20 command names and command codes are similar to those of previous versions. Hence, existing COGO input data can be used with at most minor changes.

COGO input consists of commands and, if the user wants to include them, comments. Most commands are composed of a command name followed by data for that command. Control commands do not require data.

Any information preceded by an asterisk (*) is reproduced as comments in the output. A comment may appear alone on a line or may follow the last data item on a line. If asterisks are in columns 1 and 2, the comment will start at the top of the next page.

1.2 DIFFERENCES BETWEEN PREVIOUS COGO VERSIONS AND COGO-10/20

COGO-10/20 is compatible with previous commonly used COGO versions except for a few changes noted in this section. COGO-10/20 does not require the local origin control record used with some versions of COGO. However, the START OF JOB, END OF JOB, and END OF RUN records must be inserted, and changes must be made to the commands listed below.

In earlier versions of COGO, the commands AREA, AREA AZIMUTHS, and AREA BEARINGS allowed the listing of points that are input to them to be continued on more than one card. COGO-10/20 also allows this with the restriction that the list, if continued to another card, must be enclosed in parentheses; a left parenthesis must precede the first point on the first card, and a right parenthesis must follow the last point on the last card.

The vertical alignment commands are deleted, because they did not communicate with any COGO table and, therefore, need not be part of COGO.

The DIVIDE/AREA command is replaced by the more useful ADJUST AREA command.

The POINTS INTERSECT, AZIMUTH INTERSECT, and BEARING INTERSECT commands no longer have the optional tolerance value. Instead, these commands have optional offsets. When mixing old input with new input, remove any tolerance values from old input because, if left in, the tolerances offset the intersection causing small errors. The degree of curvature is entered in the same format as an angle. This means that old input is acceptable if the degree of curvature was specified in decimal degrees with no minutes or seconds.

Commands can be abbreviated by omitting any characters after the first character. Ambiguous abbreviations are not accepted. Unacceptable abbreviations result in INVALID COMMAND error messages.

In order that COGO input and output conform to the usual notation of surveyors, angles precede distances when angles and distances appear in a command together. This change affects the input format of four commands:

```
LOCATE AZIMUTH
LOCATE BEARING
LOCATE ANGLE
LOCATE DEFLECTION
```

COGO is augmented by the following COGO-10/20 commands:

```
ADJUST ANGULAR ERROR
ADJUST TRAVERSE COMPASS
ADJUST TRAVERSE CRANDALL
ADJUST TRAVERSE TRANSIT
ANGLES INTERSECT
FORESECTION
LIST POINT NUMBERS
SET ERROR LIMIT
SET OUTPUT DSK:
SET OUTPUT LPT:
SET OUTPUT TTY:
STAKING NOTES
```

1.3 OPERATIONAL MODES

Users of COGO-10/20 have the option of working in the on-line or off-line mode.

1.3.1 On-Line Mode

In the on-line mode, users work directly with the computer by using a terminal keyboard to type the commands. The unit of processing is the single command, and users are expected to interact with the computer after each command. If a mistake is made, an error message is printed; users can correct the mistake in the next command and then continue. Output of the on-line mode appears on the terminal or the line printer immediately after the execution of each command.

1.3.2 Off-Line Mode

In the off-line mode, users write the commands on paper and have them keypunched. In this mode of operation, the unit of processing is the job, made up of many commands, with no interaction on the part of users from start to finish of the job. If an input error is detected, an appropriate message is printed. The remaining input of the job, however, is still processed. A printout of the output is returned to the user and any mistakes can be corrected and the problem rerun. Several jobs can be batched and run at the same time. Errors in one job do not affect the execution of another job in the same run.

CHAPTER 2

CONVENTIONS AND DEFINITIONS OF TERMS

2.1 COMMAND NAME FIELD

The command name can be spelled out in full or can be abbreviated by omitting any letters other than the first of a word. The command name must be the first item of a command. If the command name is left blank, the previous name is used (automatic ditto feature).

If the first nonblank column of a COGO input record is alphabetic, it is understood by COGO to be the first character of a command name.

A command name can be entered exactly as it appears or can be abbreviated. At least the first letter of each word of the desired command name must appear. Other letters may be omitted as long as the meaning of the resulting abbreviation is not ambiguous. The words (or abbreviated words) must be separated by one or more blanks and slashes.

Examples:

Abbreviation	Meaning
STORE	STORE
STR	STORE
S	Invalid (could be store or segment or show)
ST	Invalid (could be store or segment)
SE	Invalid (could be store or segment)
SR	STORE
SG	SEGMENT
SEG	SEGMENT
SEG PLUS	SEGMENT PLUS
SG PL	SEGMENT PLUS
S/M	SEGMENT MINUS
PTS INTR	POINTS INTERSECT
BR/INTERSECT	BEARING INTERSECT
BEAR INT	BEARING INTERSECT
B I	BEARING INTERSECT
A	Invalid (could be area, angles, or alignment)
AA	Invalid (could be area azimuths or adjust area)
AA A	AREA AZIMUTHS
A AA	ADJUST AREA
AL	Invalid (could be angles or alignment)
AN	Invalid (could be angles or alignment)

Abbreviation	Meaning
AG	Invalid (could be angles or alignment)
ANG	ANGLES
AGN	ALIGNMENT
ANN	ALIGNMENT
C/PR	CARD PRINT
E OJ	Invalid (no space between O and J
STOER	Invalid (characters out of sequence)
E O J	END OF JOB

If a command that requires data is entered without any data, COGO does not process the command at that time, but reads again looking for data. This allows a record with only a command name to be placed ahead of a group of blank commands. This also allows the operator, after interrupting the normal job sequence, to enter the name of the command that was interrupted and then return to card or data file input. Commands that do not require data can be processed without data. This includes all control commands.

2.2 DATA FIELDS AND DATA TYPES

Data is in free format and can start any place on the record. All the data pertaining to one command need not fit on one card. If COGO does not find sufficient data on one record, COGO reads the next record until the required amount of data is found. A comment (starting with an asterisk) can be entered after the last data field on a record.

Distances, angles, bearings, and azimuths can be referenced by using a delimiter and the required number of identifying points. For example, if two points are known, but the unknown distance between them is required as part of a command, the delimiter D can be used as follows to define the distance:

```
D 21 22
```

Point 21 and 22 are the two known points. In this case, if the delimiter were not used, the distance would have to be calculated either by hand or by a previous command. The following delimiters are used to denote values to be calculated from known points:

DELIMITER	POINTS	DESCRIPTION
D	p1 p2	Denotes a straight-line distance from point p1 to point p2.
A	p1 p2	Denotes the azimuth or bearing from point p1 to point p2.
G	p1 p2 p3	Denotes the angle at point p2, clockwise from p1 to p3.

Input of angles, azimuths, bearings, or distances can be expressed as an expression in parentheses. The expression can include addition and subtraction of azimuths, angles, and bearings. Bearings in expressions must be expressed using N S E W delimiters (not quadrant numbers) and are converted to azimuths before being combined.

The types and amount of data required for execution of a command depend on what that command is. The basic types of data are:

integers
coordinates
angles
bearings
azimuths
figures
transposed figures

2.3 INTEGERS

Integer data are used primarily for identification of points (p1, p2, and so forth) and figures (fgn). An integer is written as an optionally signed number with no fractional part.

Examples:
1
-12
+134
5000

2.4 LOCAL ORIGIN

COGO-10/20 uses a 32-bit word that provides approximately 16 significant digits. However, coordinates are stored using a local origin to provide greater accuracy. The local origin works automatically to improve precision and is of no general concern to the user. The local origin is defined by the first point stored in a table. Points closest to this local origin have the best precision.

2.5 COORDINATES

Coordinates are entered as optionally signed numbers.

Examples:
100432.648
5000
-12469

The sequence and directions of coordinates can be selected by the installation (N and E, S and E, X and Y, and so forth). This manual assumes coordinates are expressed as north followed by east.

Quantities, such as distances, stations, and radii, are considered to be real values. They are entered as (1) optionally signed numbers or (2) a computed distance between two points indicated by the optionally signed letter D (signifying distance) followed by the numbers of the two points, or (3) parentheses enclosing any combination of sums and differences of types 1 and 2.

Code	Meaning
342.1	Distance of 342.1 feet
50	Distance of 50.0 feet
D 1 500	Distance from point 1 to 500
(50 + 100)	Distance of 150 feet
(342.1 + 50 - 100)	Distance of 292.1 feet
(D 1 500 - D 3 8 + 14)	Distance from point 1 to 500 minus the distance from point 3 to 8 plus 14.0 feet

2.6 ANGLES

Angles (ang) are entered as degrees, minutes, and seconds. Only the degrees portion carries a sign. If the minutes portion contains a decimal point, then the seconds are omitted. If the degrees portion has a decimal point, then both the minutes and seconds are omitted. An angle to the right, or clockwise, is positive. An angle to the left, or counterclockwise, is negative.

The following are valid angles:

Code	Meaning
75 0 5.1	75 degrees, zero minutes, 5.1 seconds right
-75 0 5.1	75 degrees, zero minutes, 5.1 seconds left
075-00-05.1	75 degrees, zero minutes, 5.1 seconds right
90-00-00	90 degrees right
90 0 0	90 degrees right
+90 0 0	90 degrees right
90.	90 degrees right
G 7 8 9	Angle at point 8 from point 7 to point 9 right
230 20 0	230 degrees, 20 minutes right
-0 30 0	30 minutes left
-G25 30 6	Angle at point 30 from point 25 to point 6 left
-70-30.5	70 degrees, 30 minutes, 30 seconds
(30-0-0)	left 30 degrees, 0 minutes, 0 seconds right
(A 5 6 - 30.)	Azimuth from 5 to 6 minus angle of 30 degrees, 0 minutes, 0 seconds

2.7 BEARINGS

Bearings (br) are entered by either the quadrant method or the N, S, E, W delimiter method. In the former, a bearing is entered as quadrant, degrees, minutes, and seconds. The quadrant is coded as follows: NE = 1, SE = 2, SW = 3, NW = 4. In the delimiter method, the angle must be bracketed by the characters N or S, and E or W. The following are valid bearings:

Code	Meaning
N 30 5 58.0W	North 30 degrees, 5 minutes, 58 seconds west
N30 5 58W	North 30 degrees, 5 minutes, 58 seconds west
4 30 5 58	North 30 degrees, 5 minutes, 58 seconds west
4 30 5 58	North 30 degrees, 5 minutes, 58 seconds west
N 30-05-58.0W	North 30 degrees, 5 minutes, 58 seconds west

Code	Meaning
4 30-5-58	North 30 degrees, 5 minutes, 58 seconds west
S90 0 0 E	Due east
1 90 0 0.0	Due east
S 0 0 0 W	Due south
A 7 30	Bearings from point 7 to point 30
(S50-0.W+A7 8-A12 3)	Bearing S 50-0-0W plus angle between course 7 8 and course 12 3

2.8 AZIMUTHS

Azimuths (az) are entered as degrees, minutes, and seconds, and are measured clockwise from north. (If the degrees portion carries a negative sign, the azimuth is measured counter-clockwise.) Bearings entered by the N, S, E, W delimiter method are also acceptable.

The following are valid azimuths:

Code	Meaning
253-42-30	253 degrees, 42 minutes, 30 seconds
253 42 30.0	253 degrees, 42 minutes, 30 seconds
0 0 0	Due north
00 00 00.0	Due north
270 00 00	Due west
-90 0 0	Due west
90-0-0	Due east
S 12 30 0 W	192 degrees, 30 minutes west
192 30 00	192 degrees, 30 minutes
A482 483	Azimuth from point 482 to point 483
(S50.0 W+A 7 8 - A 12 3)	Bearing S 50-0-0W plus AZ from 7 to 8 minus AZ from 12 to 3

2.9 FIGURES

A figure (fgn) is a list of positive whole numbers enclosed in parentheses. It is usually used as a list of point numbers. A hyphen can be used to indicate a range of numbers (for example, 1 2 3 4 can be expressed as 1-4). The significance of the figure depends on what command uses it. A figure can be a list of points that make up a boundary or alignment containing curves. In this case, the curves are indicated in proper sequence by the letter C, the point number of the CC, (center of curvature) and the letter L or R (signifying left or right as the direction of curvature).

Instead of the list, the number of the stored figure can be given. This feature eliminates the need to write out the description each time it is needed. Instead, the figure is stored once and thereafter referred to by number.

Figure	Meaning
(1 2 3 4)	Numbers 1, 2, 3, and 4
(1 - 4)	Same as above
(1 - 4 1)	Description of a closed loop (first point same as last)
(904 C12R 61)	Curve from 904 to 61 centered at 12, to the right

Figure	Meaning
(500 - 264 480 - 486)	A range may run down or up
364	This refers to stored figure number 364
(1 2) T	Same as (2 1), the T means transposed
(175 - C178R 179) T	Same as (179 C178L - 175)
364 T	The transpose of a stored figure

Parentheses are not required if the figure is the only input item for the command. However, if the description consists of only one number, the description must be enclosed in parentheses to indicate that the number is a description and not a reference to a figure.

For example, the following format means that figure 35 contains the description to be used:

```
35
```

where as the following format means the number itself is the description:

```
(35)
```

If a description does not fit on one record, the description must be enclosed in parentheses. For example, the following is interpreted as one description (5 6) followed by another description (7 8):

```
5 6
7 8
```

However, the following is interpreted as one description (5 6 7 8):

```
(5 6
 7 8)
```

Valid examples are as follows:

Command	Meaning
INVERSE BEARING 1	Compute bearings and distances between all points in figure 1
INVERSE BEARING 1 2	Compute bearing and distance between points 1 and 2
IN BRG (1 2)	(Same as above)
17 (1 C2R 3)	Compute bearings and distances from point 1 to 2 and from point 2 to 3
INVERSE B 1 C2R 3	(Same as above)

2.10 TRANSPOSED FIGURES

A transposed figure is derived from another figure by describing that figure in reverse order. For all commands that use figures as input data, COGO allows the user to follow the figure number or description with the letter T to indicate that the figure should be transposed before use. This procedure can reduce execution time on certain commands.

2.11 COGO TABLES

COGO commands define and use points and figures to solve the problems of a civil engineer and surveyor. These points and figures are kept in a file referred to as a COGO table.

As a COGO job proceeds, data needed by COGO is read from the COGO table into memory. When a new portion of the COGO table is needed, the previously used portion is written back onto the disk to make room in memory for the new portion.

A COGO table can have as many as 9999 sets of coordinates and 9999 figures. The number of tables allowed is limited only by the disk space available.

All COGO jobs must begin with the START OF JOB command. If the name of a COGO table is specified with the START OF JOB command, the START OF JOB command searches for that table. If no COGO table name is given, the START OF JOB command opens a temporary table called COGTAB.TMP.

All COGO jobs should end with either the END OF RUN command or the END OF JOB command. Either of these commands writes on the disk the portions of the COGO table still in memory.

2.12 INFINITE EXTENSION COMMANDS

Certain commands in the COGO system consider the ends of figures to be extended along straight lines to infinity. Users should keep this in mind when using these commands, because unwanted intersections can be encountered. Unwanted possible intersections are illustrated by the FIGURE ARC INTERSECT command in Section 3.6. Other commands where unwanted possible intersections can occur are as follows:

STATIONS AND OFFSETS
POINTS ON ALIGNMENT
LOCATE FROM ALIGNMENT
ALIGNMENT OFFSET
FIGURE LINE INTERSECT
FIGURE ARC INTERSECT
FIGURE FIGURE INTERSECT
STREETS INTERSECT
DESCRIBE ALIGNMENT AZIMUTHS
DESCRIBE ALIGNMENT BEARINGS

CHAPTER 3

COGO COMMANDS

A command consists of a command field followed, in general, by a data field. A command field contains the name of a command, an abbreviation of that name, or a numeric code. A data field contains data to be processed. The control commands are the only commands that do not require any data.

The rules for abbreviating a one-word command name are:

1. The first letter of the abbreviation must be the first letter of the command name.

2. Each of the letters in the abbreviation must appear in the command name in the same order.

3. The abbreviation must be unique in the sense that it cannot be derived from any other command name.

The rules for abbreviating a command name consisting of more than one word are similar--rules 1 and 2 apply to the abbreviation of each word in the command name, rule 3 applies to the command name as a whole, and at least one space or exactly one slash must be between the abbreviation of each word in the command name.

A command name or its abbreviation can start in any column of an input record. At least one space must separate adjacent data in the data field. No space is required between the command name and the first datum. If a series of consecutive input records uses the same command name, the name need not be repeated. The name or an abbreviation must appear in the first record. This capability is called the automatic ditto feature.

An asterisk in column 1 of an input record causes the contents of columns 2 to 67, inclusive, to be included in the output as a comment. If both columns 1 and 2 contain an asterisk, then the comment is printed at the top of a new page. A comment can follow an asterisk placed after the last datum on any input record that is not continued onto another card. An asterisk in column 1 or in columns 1 and 2 does not nullify the automatic ditto feature.

A blank input record can be included in an input file to cause blank lines to appear in the output. This does not nullify the automatic ditto feature.

In this chapter, the following abbreviations are used in data descriptions and in diagrams:

Notation	Meaning
ang	Angle
arc	Distance measured along an arc
az	Azimuth
br	Bearing
cid	Curve identification
d	Linear distance
def	Deflection
fgn	Number of a stored figure
n,nl,nc	Points to be computed (c = center of a circle)
off,offl,offfg	Offsets (fg = figure)
p,pl,pc	Known points (c = center of a circle)
pi	Known point used to pick desired intersection
r	Radius of circle
s,ps,ns	Station (p=known and n=unknown)

All other data names are defined in the description of the corresponding command.

3.1 CONTROL COMMANDS

Control commands control the flow of a COGO run. These commands specify the COGO table for a particular job, the input and output devices, the beginning and end of a job, and the end of a run.

- - - - - - - - - - -

START OF JOB [/name]

This command should be the first record in any COGO job. It resets all indicators, selects the proper coordinate table, and heads the first page with the comment heading. If the switch /name is used, COGO opens a permanent file for the COGO table called name.CGT. If this switch is omitted, COGO opens a temporary file for the COGO table called COGTAB.TMP.

Example:

```
START OF JOB /SURVEY * Opens a file named SURVEY.CGT
```

- - - - - - - - - - -

END OF JOB

This command should be the last statement in every COGO job that is not the last job in a run. It updates the COGO table, types the message END OF JOB, and pauses. If COGO is resumed, it expects a control command to be entered at a terminal.

- - - - - - - - - - -

END OF RUN

This command is identical to the END OF JOB command except that COGO is terminated after an END OF RUN command. This command should be the last statement in the last job in a run.

- - - - - - - - - - -

SET ERROR LIMIT num

This command sets the number of errors that are allowed before the job is aborted. The default number is 100. NUM is the number of errors allowed.

Example:

```
SET ERROR LIMIT 20      * Sets maximum errors to 20.
```

- - - - - - - - - - -

SET OUTPUT LPT:

This command changes the output device for printed output to the line printer.

- - - - - - - - - - -

SET OUTPUT TTY:

This command changes the output device for printed output to the user's terminal.

- - - - - - - - - -

SET OUTPUT DSK: [/name.ext]

This command specifies that a disk file called name.ext will be used for output. If no name.ext is specified, then COGO defaults to a file of COGO.DSK.

CAUTION: Specification of the same name.ext within the same run results in overwriting the file's previous contents.

Example:

```
SET OUTPUT DSK: /MONTCO.OUT
```

This command changes the output device to a disk file called MONTCO.OUT.

3.2 MAINTAINING POINTS AND FIGURES

Points are maintained by direct input and deletion. FIGURE commands provide the user with the ability to transfer figures from one storage medium to another, for example, input terminal to table. FIGURE commands deal only with the figure description and do not use the coordinates of the points that the figure may specify. The user should keep in mind that a figure can describe not only a group of points but also a group of figures. In fact, a figure is no more than a list of numbers and delimiters and in the following commands should be thought of as such.

- - - - - - - - - -

STORE n nor eas

Store as point number n, the coordinates nor feet north and eas feet east.

Output: Coordinates of point n.

Example:

```
STORE 8 5313.  4993.
      8
```

- - - - - - - - - -

REDEFINE p n

Use the present coordinates of known point p to define point n.

Example:

```
REDEFINE 8 16
```

- - - - - - - - - -

PUNCH COORDINATES desc

The coordinates specified in desc are copied to an output file named FOR20.DAT. Undefined points are omitted. The output is in the form of STORE commands, which can later be used as input.

```
PUNCH COORDINATES 1-50          *Punch points 1 through 50

PCH COOR (1)                    *Punch point 1

PUNCH COR 1                     *Punch the points
                                *specified by figure 1

PUNCH COOR 1 50                 *Punch points 1 and 50
```

- - - - - - - - - -

DELETE COORDINATES desc

This command deletes the coordinates specified in desc. If an attempt is made to use any of the deleted coordinates, an UNDEFINED POINT error message is printed.

Examples:

```
STORE FIGURE 5 (7 12 3)
DELETE COORD (5)                    *Delete coordinates of
                                    *point 5

DELETE COORD 5                      *Delete the coordinates of
                                    *figure 5 (points 7, 12,
                                    *and 3)

DELETE COORDS (10 20 30)            *Delete coordinates 10,
                                    *20, and 30

DELETE COORDS 10 20 30              *Same as above

DEL COORDS 1-9999                   *Delete all coordinates
```

- - - - - - - - - - -

STORE FIGURE fgn desc

Store in the figure table, as figure fgn, the description desc. The figure fgn must be in the range 1 to 9999. The numbers in the description (not the coordinates) are stored in the figure area of a COGO table. The description can be continued on the following input records if desired. If figure fgn was previously defined, the new description will replace the old one. A figure can be used to specify a base line, property line, or any group of points or numbers.

If a figure, or portion of a figure, is made up of all numbers p1 through p2, the numbers can be listed or p1 and p2 separated by - will automatically fill them in. For example, the following two commands do the same thing:

```
STORE FIGURE 8 ( 34 18 3 4 5 6 8 9 10 11 12 25 39 38 37 36
35 34)

STORE FIGURE 8 ( 34 18 3 - 6 8-12 25 39-34 )
```

If another figure number is entered as desc, then figure fgn is stored as a duplicate of the figure entered as desc. For instance, in the following command, figure 60 is identical to figure 8:

```
STORE FIGURE 60 8
```

To indicate in a STORE FIGURE command that two points are connected by a circular curve rather than a straight line, the two points should be separated by the letter C, the point number of the CC (center of curvature), and the letter L or R to indicate left or right.

Example:

```
STORE FIGURE 1 ( 25 26 27 28 C29R 30 31 31 C 75 L 33 8 9 15
25)

STORE FIGURE 1 (25-C29R-32 C75L 33 8 9 15 25 )
```

- - - - - - - - - - -

PUNCH FIGURES desc

The figures specified are written to the output file named FOR20.DAT. Undefined figures are not punched. The output is in the form of STORE FIGURE commands, which can be used as input later.

Examples:

```
PUNCH FIGS (1 3)              *Punch figures 1 and 3

          (10 10)             *Punch figure 10 twice

          (1-20 600-650)      *Punch figures 1 through
                              *20 and 600 through 650

PUNCH FIGURES 77              *Punch the figures whose
                              *numbers appear in figure
                              *77

PUNCH FIGURE (77)             *Punch figure 77
```

- - - - - - - - - - -

DELETE FIGURES desc

This command deletes the figures specified in desc from the COGO table. If an attempt is made to use any of the deleted figures, an UNDEFINED FIGURE error message is printed.

Examples:

```
DELETE FIGURE (5)
STORE FIGURE 5 (1-20)         *Delete figure 5
DELETE FIGURES 5              *Delete the figures whose
                              *numbers appear in figure
                              *5 (figures 1 through 20)

DELETE FIGURES (1-20)         *Delete figures 1 through
                              *20
```

This page intentionally left blank.

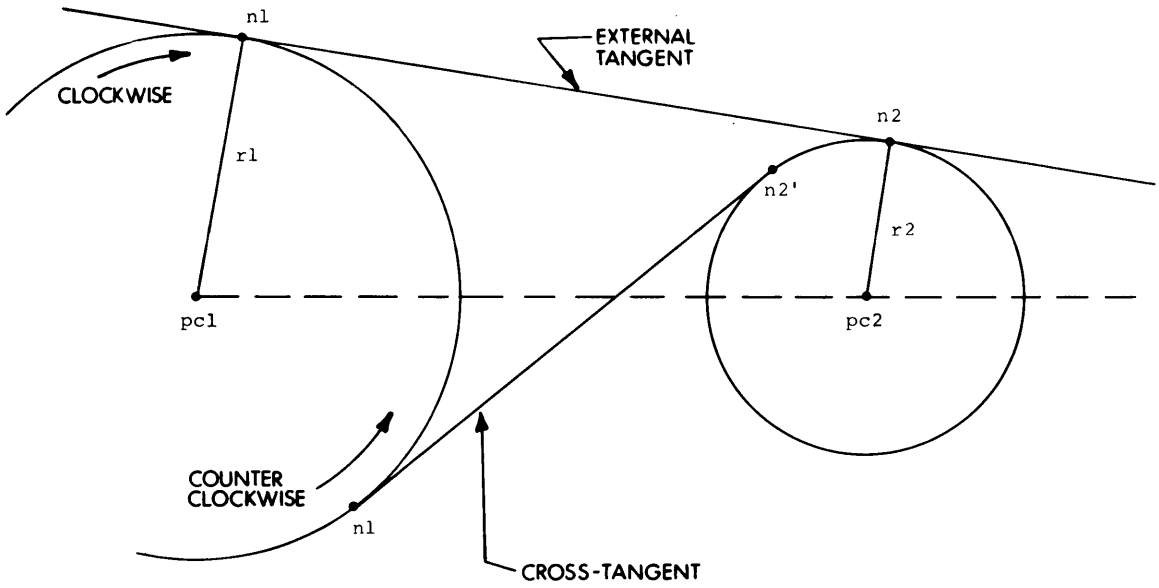
3.3 COORDINATE LOCATIONS

The following commands provide for geometric calculations that are composed of combinations of line extensions and intersections. Users can accomplish the same function by using combinations of other commands. However, these commands often provide a more direct solution.

- - - - - - - - - -

TANGENT n1 pc1 r1 n2 pc2 r2 [sign ext]

Find the tangent to two circles by locating the point of tangency n1 on the circle with center pc1 and radius r1, and the point of tangency n2 on the circle with center pc2 and radius r2. The larger circle must be entered first. The variables sign and ext are used to indicate which of the four possible tangents is described. If the tangent connecting the two circles leaves the first circle in a clockwise direction, as shown in the figure below, sign should be 1. Otherwise, it should be -1. Stated differently, the variable sign is 1 when the angle formed by the extension of the line connecting the two centers and the extension of the tangent is clockwise measured from the connecting line. The argument sign is -1 when the angle is counterclockwise. When an external tangent is being described, ext should be 1; for a cross or internal tangent, ext should be -1.



The arguments sign and ext are optional. If they are not entered, the larger circle need not be entered first. The user describes the desired tangent as follows: when looking from the center of the first circle toward the center of the second circle, if the point of tangency is on the left, the radius of the circle or circles should be entered as negative; otherwise, they should be entered as positive.

Output: Coordinates of n1 and n2, azimuth of the tangent connecting n1 and n2, and the distance from n1 to n2.

Example 1 for TANGENT:

```
DELETE COORDINATES 1-100
STORE 1 200 100
      1
2 200 300
      2
TANGENT 9 1 -50 10 2 -25
      9        249.6078        106.2500

     10        224.8039        303.1250

                    FROM    9  TO   10        97-10-50.7        198.4313

7 1 -50 6 2 25
      7        246.3512        118.7500

      6        176.8244        290.6250

                    FROM    7  TO    6       112- 1-27.5        185.4050

5 1 50 8 2 -25
      5        153.6488        118.7500

      8        223.1756        290.6250

                    FROM    5  TO    8        67-58-32.5        185.4050

3 1 50 4 2 25
      3        150.3922        106.2500

      4        175.1961        303.1250

                    FROM    3  TO    4        82-49- 9.3        198.4313
```

Example 2 for TANGENT with alternate form of data input:

```
DELETE COORDINATES 3-10
TANGENT 10 2 25 9 1 50
      10          224.8039          303.1250

       9          249.6078          106.2500

                        FROM   10  TO     9        277-10-50.7          198.4313

6 2 -25 7 1 50
       6          176.8244          290.6250

       7          246.3512          118.7500

                        FROM    6  TO     7        292- 1-27.5          185.4050

8 2 25 5 1 -50
       8          223.1756          290.6250

       5          153.6488          118.7500

                        FROM    8  TO     5        247-58-32.5          185.4050

4 2 -25 3 1 -50
       4          175.1961          303.1250

       3          150.3922          106.2500

                        FROM    4  TO     3        262-49- 9.3          198.4313
```

Example 3 for TANGENT with alternate form of data input:

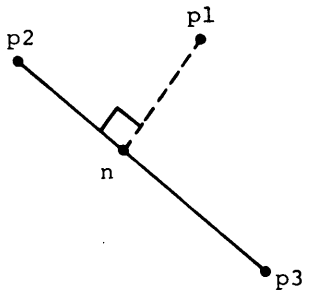
```
DELETE COORDINATES
        1-9999
DELETE FIGURES
        1-99
STORE 1 200 400
      1
6 200 1000
      6
TANGENT 7 6 200, 2 1 100, -1 1
      7          397.2027          966.6667

      2          298.6013          383.3333

                      FROM    7  TO     2        260-24-21.4          591.6080

8 6 200, 4 1 100, 1 -1
      8           26.7949          900.0000

      4          286.6025          450.0000

                      FROM    8  TO     4        300- 0- 0.0          519.6152

9 6 200, 5 1 100, 1 1
      9            2.7973          966.6667

      5          101.3987          383.3333

                      FROM    9  TO     5        279-35-38.6          591.6080

10 6 200, 3 1 100, -1 -1
      10         373.2051          900.0000

      3          113.3975          450.0000

                      FROM   10  TO     3        240- 0- 0.0          519.6152
```

- - - - - - - - - - -

TANGENT OFFSET n p1 p2 p3

Find the point n by intersecting the line connecting p2 and p3 with the perpendicular offset from p1.

Output: Coordinates of n, distance from p2 to n, and distance from n to p1.

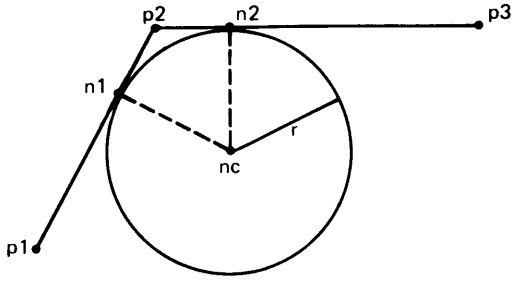


Example:

```
DELETE COORDINATES 1-10
STORE 2 100 100
      2
3 500 600
      3
1 400 150
      1
TANGENT OFFSET 10 1 2 3
      10         241.4634          276.8293

                      FROM    2  TO    10        226.4520 FT.

                      FROM   10  TO     1        203.0259 FT. LEFT
```

- - - - - - - - - -

FIT CURVE pl p2 p3 nl nc n2 [r]



Fit a circular curve of radius r tangent to the lines from pl to p2 and from p2 to p3. The points nl and n2 are the two points of tangency. The radius r may be omitted if pl = nl or p3 = n2.

Output: Coordinates of nl, nc, and n2.

Example 1 for FIT CURVE:

```
DELETE COORDINATES 1-100
STORE 1 200 200
      1
2 400 200
      2
3 500 400
      3
FIT CURVE 1 2 3 5 6 7 100.
      5        338.1966        200.0000

      6        338.1966        300.0000

      7        427.6393        255.2786

DIVIDE LINE 1 5 2 6
      8        269.0983        200.0000
```

Example 2 for a FIT CURVE with radius unknown:

```
FIT CURVE 8 2 3 8 9 10
      8        269.0983        200.0000

      9        269.0983        411.8034

     10        458.5410        317.0820
```

- - - - - - - - - -

SIMPLE CURVE cid pb ptc ntt nct dc cang sign

This command requires the following information about a simple circular curve:

cid — Curve identification number. A number (0-999) that uniquely defines a curve.

pb — A predefined point anywhere on the back tangent.

ptc — The predefined point of transition from tangent to curve. The curvature starts at this point.

ntt — The new point number assigned to the intersection of the back tangent and the ahead tangent.

nct The new point number assigned to the point of transition between curve and tangent. The curvature ends at this point.

dc The degree of curvature of the curve. The angle, in decimal degrees, that subtends a 100-foot arc. For a circular arc, dc = 100*180/(3.14*r), where r is the radius of the arc.

cang The central angle of the curve.

sign Equals 1 for a clockwise curve and -1 otherwise.

Output: cid, coordinates of tc, tt, and ct, tangent length (from tc to tt), arc length, back and forward azimuths.

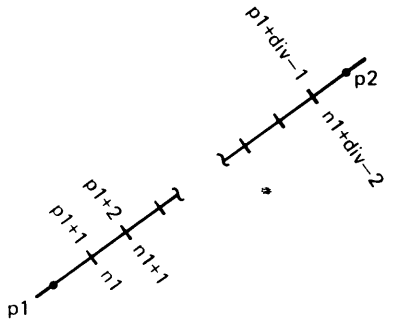
NOTE

SIMPLE CURVE must be used independently of any other commands. The commands COORD POA, COORD OFFSET, STATION FROM COORD, and OFFSET ALIGN must be used on a curve entered by the ALIGNMENT or DEFINE CURVE command and not on a curve entered by the SIMPLE CURVE command.

- - - - - - - - - - -

DIVIDE LINE p1 p2 div [n1]

Divide the line joining the known points p1 and p2 into div equal parts. If n1 is omitted, the intermediate points are assigned the numbers p1+1,..., p1+(div-1). If n1 is entered, the intermediate points are given the values n1, n1+1,..., n1+(div-2). Since point numbers are assigned successively from p1 when n1 is omitted, users must exercise care to avoid destroying known points.

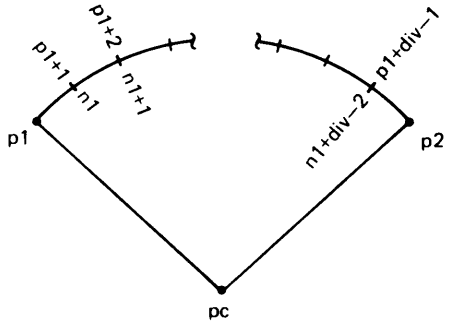


Output: Coordinates of each intermediate point.

- - - - - - - - - - -

DIVIDE ARC p1 p2 pc div [n1]

Divide the clockwise arc from p1 to p2 around the known center of the circle pc into div equal parts. If n1 is omitted, the div-1 intermediate points are assigned the numbers p1+1, p1+2,...,p1+(div-1). If n1 is entered, then the intermediate points are given the point numbers n1, n1+1,..., n1+(div-2). When n1 is omitted, care must be taken to avoid destroying known points.



Output: Coordinates at each intermediate point.

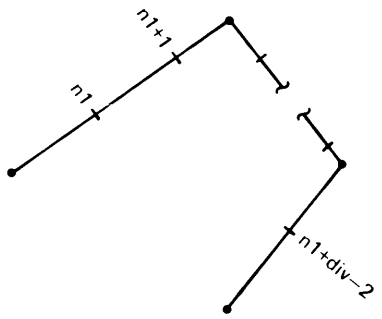
Examples for DIVIDE LINE and DIVIDE ARC:

```
STORE 15 0 0
     15
20 1000 0
     20
30 0 1000
     30
DIVIDE LINE 15 20 3
     16        333.3333        -0.0000

     17        666.6667        -0.0000

DIVIDE ARC 20 30 15 4
     21        923.8795        382.6834

     22        707.1068        707.1068

     23        382.6834        923.8795
```

- - - - - - - - - - -

DIVIDE FIGURE desc div nl

Divide the figure described by desc into div equal parts. The div-1 intermediate points are assigned point numbers nl, nl+1, ..., nl+(div-2). Some care must be exercised in the selection of nl so that no known points are destroyed by assigning the intermediate points numbers ranging from nl to nl+(div-2).

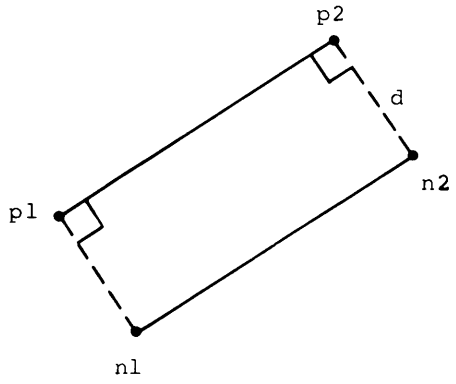


Output: Coordinates of each intermediate point.

- - - - - - - - - - -

PARALLEL LINE p1 p2 off n1 n2

Compute points n1 and n2 to define a line parallel to the line joining the known points p1 and p2 that is at an offset distance off to the left or right depending on whether off is negative or positive, respectively.



Output: Coordinates of n1 and n2.

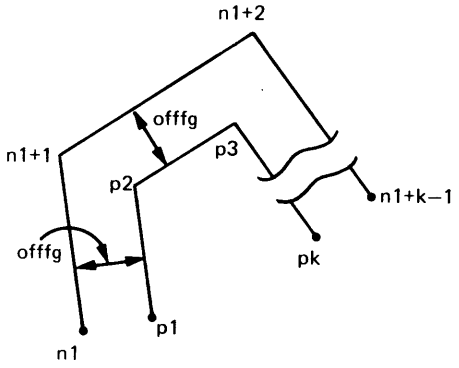
Examples for PARALLEL LINE:

```
STORE 31 200 200
      31
32 400 100
      32
PARALLEL LINE 31 32 150. 33 34
          31 32 150. 33 34
      33        267.0820        334.1641

      34        467.0820        234.1641

31 32 -100. 35 36
      35        155.2786        110.5573

      36        355.2786         10.5573

34 36 100. 37 38
      37        556.5248        189.4427

      38        444.7214        -34.1641
```

- - - - - - - - - - -

PARALLEL FIGURE fgn offfg nl

Let k denote the number of points in the figure fgn. Then locate a line parallel to the line described by the figure fgn by computing points nl, nl+1,..., nl+(k-1) offset at a distance offfg. The new line will be inside or outside depending on whether offfg is negative or positive, respectively.



The centers of circular curves are not computed, since they are the same as those of the figure fgn. Care must be exercised in the selection of nl to avoid destroying known points by assigning point numbers from nl to nl+(k-1).

Output: Coordinates of the points in the parallel lines.

Example:

```
DELETE COORDINATES 1-100
STORE 11 982.8517 1140.9693
      11
14 907.4955 1177.3215
      14
23 1000.0000 1000.0000
      23
24 1044.3302 1023.1214
      24
25 980.3919 1181.7215
      25
26 942.9598 1195.8224
      26
27 1027.1821 1164.0954
      27
STORE FIGURE 1 (23 14 C26L 25 C27L 11 24)
                1
PARALLEL FIGURE 1 10. 30
      30        991.1339        995.3748

      31        898.6294       1172.6963

      32        989.7499       1178.1963

      33        973.9856       1136.3441

      34       1053.1963       1027.7466

1 -20. 35
      35       1017.7322       1009.2505

      36        925.2277       1186.5720

      37        961.6758       1188.7720

      38       1000.5839       1150.2197

      39       1026.5980       1013.8710

STORE FIGURE 2 (30 31 C26L 32 C27R 33 34)
                2
3 (35 36 C26L 37 C27R 38 39)
                3
```

- - - - - - - - - -

STREETS INTERSECT r fg1 w1 fg2 w2 pi ncls nll nlr nrl nrr

Locate the intersection of two streets, including PCs, CCs, and PTs of street returns with radius r if r is positive, or only PLs, if r is zero. The center line and width of the streets are described by figure fg1 and width w1 for one, and figure fg2 and width w2 for the other, with their intersection being the one closest to point pi or farthest if pi is negative.

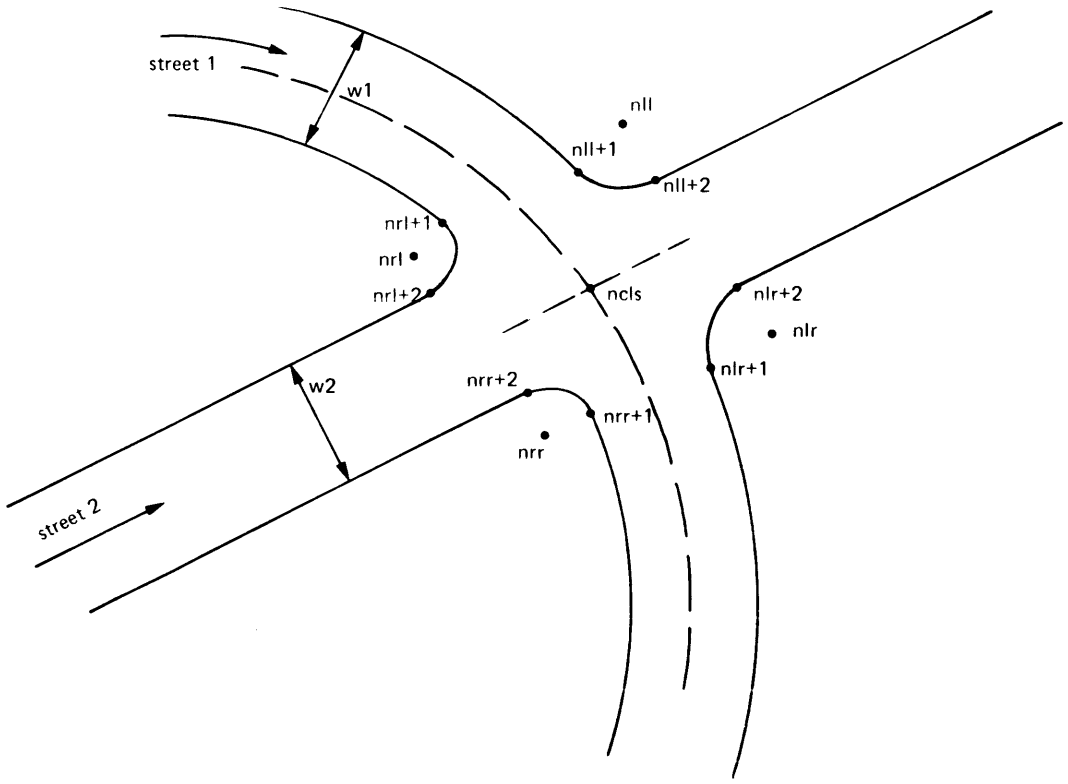
ncls	Point of intersection of the center lines.
nll	Center of curve of return, or PL of intersection at the left side of both streets.
nll+1	PC of corresponding return (if any) on street 1.
nll+2	PT of corresponding return (if any) on street 2.

nlr	Center of curve of return or PL of intersection at the left of street 1 and right of street 2.
nlr+1	PC of corresponding return (if any) on street 1.
nlr+2	PT of corresponding return (if any) on street 2.
nrl	Center of curve of return or PL of intersection at right of street 1 and left of street 2.
nrl+1	PC of corresponding return (if any) on street 2.
nrl+2	PT of corresponding return (if any) on street 2.
nrr	Center of curve of return or PL of intersection at right of both streets.
nrr+1	PC of corresponding return (if any) on street 1.
nrr+2	PT of corresponding return (if any) on street 2.

The radius r can be entered as zero to indicate that the intersection is not to have returns. The figures fg1 and fg2 may be specified as edge of street by making w1 and w2 zero, respectively. The points nll, nlr, nrl, or nrr may be zero, thereby indicating no operation in corresponding sector. PCs are considered to be on street 1, and PTs on street 2. The argument ncls can also be zero to indicate the intersection of the CLs is not to be stored.

This command also can be used for intersecting a right-of-way with a property line.

Output: Coordinates of ncls; nll, nll+1, nll+2; nlr, nlr+1, nlr+2; nrl, nrl+1, nrl+2; nrr, nrr+1, nrr+2.



Example:

```
DELETE COORDINATES 1-100
DELETE FIGURES 1-10
STORE 1 0 1000
       1
2 100 1000
       2
LOCATE ANGLE
         1 2 3 90. 500
       3         100.0000         500.0000

2 3 4 -30. 500
       4         350.0000         933.0127

2 3 5 -30. (500+600)
       5         650.0000        1452.6279

3 5 6 20. 600
       6         545.8111         861.7433

5 6 7 -90. 100
       7         644.2919         844.3785

STORE FIGURE 1 ( 1 2 C3L 4 C5R 6 7)  *C/L OF STREET A
                1
STORE 8 250 700
       8
9 750 1200
       9
STORE FIGURE 2 (8 9)  *C/L OF STREET B
                2
STREETS INTERSECT 40. 1 80. 2 100. 4 10 15 20 25 30
           CL INTERSECTION IS AT SEGMENTS     4     6 AND     8     9
      10         440.4210         890.4210

           LL INTERSECTION IS AT SEGMENTS     4     6 AND     8     9
      15         473.2735         795.9943

      16         483.6692         834.6198

      17         444.9892         824.2786

           LR INTERSECTION IS AT SEGMENTS     2     4 AND     8     9
      20         294.8088         872.0881

      21         313.3621         907.5250

      22         323.0931         843.8038

           RL INTERSECTION IS AT SEGMENTS     6     7 AND     8     9
      25         609.0975         931.8183

      26         602.1516         892.4260

      27         580.8133         960.1026

           RR INTERSECTION IS AT SEGMENTS     4     6 AND     8     9
      30         412.6680         989.9472

      31         394.4116         954.3563

      32         440.9522         961.6629
```

- - - - - - - - - -

This page intentionally left blank.

3.4 ADJUST COMMANDS

The following commands use the present coordinates of a point to compute new coordinates of the same point. Care must be taken when using these commands to be sure to "do it right the first time", since the original coordinates of points changed by the command are destroyed (replaced by the computed results) during execution of the commands.

- - - - - - - - - -

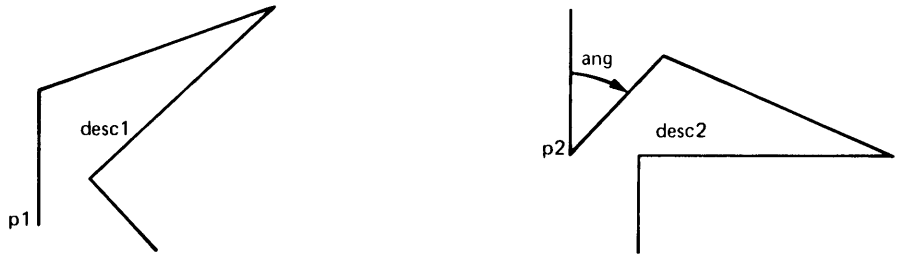
CONVERT MERIDIAN descl desc2 pl p2 ang [scale]

Rotate and translate the known coordinates of description descl with the resulting coordinates corresponding to the point numbers listed in description desc2. Point pl is a point in the description descl, and point p2 is the corresponding point with known coordinates in the converted description desc2. The rotation angle is ang. Coordinates in descl are multiplied by the optional argument scale.

Description desc2 must have as many or more points than description descl. When the end of either figure is reached, the list continues with the first point of that figure and procedes until all the points of descl have been converted.

The CONVERT MERIDIAN command can be used to change the coordinate axes orgin and azimuth from a known arbitrary reference to any specified reference. Dimension conversions can also be made by using the optional argument scale.

Output: Coordinates of each point in desc2.



Example 1 for CONVERT MERIDIAN:

```
STORE FIGURE 20 (500-502)
                20
3 (888 700 750 888)
                  3
STORE 700 25000 30000
    700
750 24962.6178 30033.2050
    750
500 1000 500
    500
888 21910.4065 29821.1900
    888
LOCATE BEARING 500 501 N60 0 0E 200.0
    501       1100.0000        673.2051

501 502 S45 0 0E 50
    502       1064.6447        708.5604
```

```
INVERSE BEARINGS
        502 500
                    FROM  502  TO  500       S 72-46-43.8 W        218.3492

CONVERT MERIDIAN 20 3 501 700 3-23-11.9
                                      SHIFT =   50-49-17.5        37832.1412
                               CONVERSION ANGLE =    3-23-11.9
                                  SCALE FACTOR = 0.100000000E+01

AREA BEARINGS 3
    888      24910.4065      29821.1900
                                          N 63-23-11.9 E        200.0000
    700      25000.0000      30000.0000
                                          S 41-36-48.1 E         50.0000
    750      24962.6178      30033.2050
                                          S 76- 9-55.7 W        218.3492
    888      24910.4065      29821.1900

             AREA=      4829.630 SQ.FT.=    0.11087303 ACRES
```

Example 2 for CONVERT MERIDIAN using closure of a parcel from two adjoining deed descriptions:

```
DELETE COORDINATES
         1-21
STORE 1 5000 5000
      1
LOCATE BEARING 1 2 S53 0 0E 52.951    * FIRST SIDE, FROM DEED A
      2       4968.1333       5042.2885

2 3 S46 16 50E 25.0    * SECOND SIDE, FROM DEED A
      3       4950.8551       5060.3569

3 4 S38 45 30W 95.354    * THIRD SIDE, FROM DEED A
      4       4876.4987       5000.6617

4 20 N47 51 30W 88.547    * FOURTH SIDE, FROM DEED B
     20       4935.9107       4935.0052

20 21 N43 54 00E 88.843    * FIFTH SIDE, FROM DEED B
     21       4999.9266       4996.6091

INVERSE BEARINGS
          21 1    * ERROR OF CLOSURE BEFORE CONVERSION
                    FROM   21  TO    1       N 88-45-37.5 E          3.3917

STORE FIGURE 85 (1-6 1)    * LOT BOUNDARY
              85
62 (4 20 21)    * 2 SIDES FROM DEED B
              62
CONVERT MERIDIAN
         62 85 4 4 G21 4 1    * ROTATE DEED B COURSES INTO DEED A
                                      SHIFT =    0- 0- 0.0           0.0001
                               CONVERSION ANGLE =    1-34-24.9
                                  SCALE FACTOR = 0.100000000E+01
```

```
AREA BEARINGS
        85 * LAST COURSE IS ERROR OF CLOSURE
    1       5000.0000       5000.0000
                                          S 53- 0- 0.0 E        52.9510
    2       4968.1333       5042.2885
                                          S 46-16-50.0 E        25.0000
    3       4950.8551       5060.3569
                                          S 38-45-30.0 W        95.3540
    4       4876.4987       5000.6617
                                          N 46-17- 5.1 W        88.5470
    5       4937.6913       4936.6615
                                          N 45-28-24.9 E        88.8430
    6       4999.9914       5000.0000
                                          N  0-18-25.2 W         0.0086
    1       5000.0000       5000.0000

            AREA=      7719.801 SQ.FT.=    0.17722223 ACRES
```

- - - - - - - - - -

ADJUST AREA descl ar p1 p2 p3 p4 [desc2]

Adjust the area described by descl to be ar square feet. Make the adjustment in the sides p1 through p2. The points p3 and p4 are additional parameters needed by some methods. The points described in desc2, although not part of descl, are moved in the same manner as the points between p1 and p2. The first and the last point in the figure description cannot be adjusted. The points p1 and p2 must be specified in the same sequence used in the description descl.

Output: Initial area and final area expressed in square feet.

Method 1:

Rotate the sides between p1 and p2 about point p3, where p3 can be any defined point. Method 1 is called by entering p4 as zero.

Method.2:

Move the sides between p1 and p2 in a direction perpendicular to a line connecting points p1 and p2. Method 2 is called by entering p3 and p4 as zero.

Method 3:

Move the sides between p1 and p2 in a direction parallel to a line connecting point p3 to point p4, where p3 and p4 are not necessarily on descl.

Example 1 for ADJUST AREA method 1 with p3 not on descl:

```
STORE FIGURE 10 (101 105 836 4 78 C12R 52-54 781 210 101)
              10
STORE 250 5844.30 1727.40
    250
101 5835.3700 1863.3100
    101
105 5862.0700 1771.6700
    105
836 5993.1500 1732.0300
    836
4 6124.7100 1757.7600
         PREVIOUS VALUE OF POINT    4 WAS
             4876.4987      5000.6617
      4
78 6239.5300 1731.4900
     78
12 6253.2100 1791.2800
     12
DELETE COORDINATES
        52-54
STORE 52 6298.6500 1750.0800
     52
53 6324.1230 1778.1762
     53
54 6339.4800 1873.8300
     54
781 6074.2600 1888.5600
    781
210 5919.0900 1836.7000
    210
101 5835.3700 1863.3100
    101
ADJUST AREA
         10 70000 836 53 250 0
    ORIGINAL AREA=     59259.062 SQ.FT.
       FINAL AREA=     70000.000 SQ.FT.

10 55000 836 53 250 0
    ORIGINAL AREA=     70000.000 SQ.FT.
       FINAL AREA=     55000.001 SQ.FT.
```

Example 2 for ADJUST AREA method 1 with p3 on descl:

```
DELETE COORDINATES
        1-5
STORE 1 1130.02 4930.98
      1
LOCATE AZIMUTH
        1 2 24-39-00 231.17
      2       1340.1241       5027.3950

2 3 119-14-00 184.18
      3       1250.1765       5188.1175

3 4 217-28-00 274.87
      4       1032.0102       5020.9141

4 5 317-27-00 133.02
      5       1130.0044       4930.9616

INVERSE AZIMUTHS
        5 1
                    FROM    5  TO    1         49-40-11.5          0.0242

STORE FIGURE
        39 (1-4 1)
              39
ADJUST AREA
         39 43560 2 3 2 0
    ORIGINAL AREA=     39224.455 SQ.FT.
       FINAL AREA=     43560.001 SQ.FT.
```

Example 3 for ADJUST AREA method 2:

```
DELETE COORDINATES
        1-8
STORE FIGURE
        25 (1-8 1)
              25
STORE 1 10580.80 10847.39
      1
2 10600.51 10994.38
      2
3 10457.15 11232.16
      3
4 10316.50 11888.02
      4
5 10315.63 11114.78
      5
6 10257.05 11066.51
      6
7 10255.75 10926.64
      7
8 10400.67 10950.82
      8
ADJUST AREA
         25 100000 3 7 0 0
    ORIGINAL AREA=    129369.617 SQ.FT.
       FINAL AREA=    100000.001 SQ.FT.
```

```
AREA BEARINGS
      25
      1      10580.8000      10847.3900
                                          N 82-21-45.8 E      148.3056
      2      10600.5100      10994.3800
                                          S 58-54-49.7 E       58.9369
      3      10570.0793      11044.8531
                                          S 77-53-46.2 E      805.2059
      4      10401.2406      11832.1587
                                          S 89-56- 7.9 W      773.2405
      5      10400.3706      11058.9187
                                          S 39-29-18.8 W       75.9053
      6      10341.7906      11010.6487
                                          S 89-28- 3.0 W       69.7639
      7      10341.1422      10940.8877
                                          N  9-28-21.3 E       60.3507
      8      10400.6700      10950.8200
                                          N 29-51-51.4 W      207.7127
      1      10580.8000      10847.3900

            AREA=    100000.001 SQ.FT.=    2.29568412 ACRES
```

Example 4 for ADJUST AREA method 3 with p3 and p4 not on descl:

```
DELETE COORDINATES
       8-27
STORE
    8 5313. 4993.
        8
    9 5355. 5032.
        9
    25 5000 5000
       25
    LOCATE AZIMUTH 25 26 192-30-00 208.44
       26       4796.5009       4954.8853

    26 27 345-20-00 194.50
       27       4984.6631       4905.6389

    27 12 15-00-00 250.
       12       5226.1446       4970.3436

    EXTEND ARC 12 27 13 120.0
       13       5168.9766       5074.5428

    AZIMUTH INTERSECT
            14 13 187-10-00 25 37-40-00
       14       5082.4733       5063.6660

    STORE FIGURE 803 (12 C27R 13 14 25-27 12)
                 803
    ADJUST AREA 803 40000 13 26 8 9
        ORIGINAL AREA=     33961.171 SQ.FT.
           FINAL AREA=     40000.001 SQ.FT.

    803 26000 13 26 8 9
        ORIGINAL AREA=     40000.001 SQ.FT.
           FINAL AREA=     26000.001 SQ.FT.
```

Example 5 for ADJUST AREA method 3 with p3 and p4 on descl:

```
DELETE COORDINATES
        101-105
STORE FIGURE 999 ( 103 105 104 102 101 103)
             999
STORE 101 10000 10000
    101
102 9949 10160
    102
103 9843.5 9972
    103
104 9870 10132
    104
105 9820 10180
    105
ADJUST AREA 999 19000 105 102 103 105
    ORIGINAL AREA=     23819.001 SQ.FT.
       FINAL AREA=     19000.000 SQ.FT.

999 15000 105 102 103 105
    ORIGINAL AREA=     19000.000 SQ.FT.
       FINAL AREA=     15000.000 SQ.FT.

AREA AZIMUTHS
         999
    103       9843.5000       9972.0000
                                          96-26-45.6          142.6484
    105       9827.4854      10113.7466
                                         316-10- 8.9           69.3109
    104       9877.4854      10065.7466
                                          19-30-57.1           96.8110
    102       9968.7344      10098.0881
                                         287-40-46.9          102.9505
    101      10000.0000      10000.0000
                                         190- 8-37.2          158.9851
    103       9843.5000       9972.0000

              AREA=     15000.000 SQ.FT.=    0.34435262 ACRES
```

- - - - - - - - - -

ADJUST TRAVERSE COMPASS desc nclos accur [nstr brglst aerr]

ADJUST TRAVERSE CRANDALL desc nclos accur [nstr brglst aerr]

ADJUST TRAVERSE TRANSIT desc nclos accur [nstr brglst aerr]

Adjust the traverse described by desc closing the last point in the description desc to point nclos. The traverse adjustment will not be made unless the error is less than 1.0 in accur.

The optional parameter nstr defines the point number at which to start storing the adjusted traverse. If nstr is omitted or is the same as the first point in desc, then the adjusted traverse overlays the original traverse. If nstr is different from the first point in desc, then COGO stores the adjusted traverse in consecutive point numbers beginning at point nstr.

The optional parameter brglst, when given, indicates the bearing of the last course of the traverse described by desc. The angular error between brlst and the last course of desc is computed and printed. However, no angular adjustment is made.

The optional parameter aerr defines the maximum allowable error between brglst and the last course of desc. If the angular error exceeds this amount, no traverse adjustment is made.

Examples:

```
AD TR COMP (1-7) 1 0
                        * Adjust the traverse defined by
                        * points 1-7 by the COMPASS
                        * method closing point 7 back
                        * to point 1.  An accuracy of
                        * 1.0 in 0 (unlimited) is
                        * allowable.  Store the adjusted
                        * traverse back in points 1-7.

ADJ TR CRNDL (1-4 8 7 12-15) 22 10000 101 A 256 300 1.

                        * Adjust the defined traverse
                        * by the CRANDALL method closing
                        * point 15 to point 22 if the
                        * accuracy is better than 1.0 in 10000.
                        * The last course in the traverse (14-15)
                        * should have the same bearing as a
                        * line between points 256 and 300.
                        * If this bearing is more than
                        * 1-DEG-00-min-00-SEC in error,
                        * then do not close the traverse.
                        * If traverse adjustment is made, store
                        * adjusted coordinates in successive
                        * point numbers starting at point 101.
                        * The adjusted traverse could be
                        * referred to as (101-110).
```

Three methods of traverse adjustment are provided. A brief description of each method and its use are given as follows:

** COMPASS **

The COMPASS rule states that the correction to be applied to the latitude/departure of any course is to the total correction in latitude/departure as the length of the course is to the length of the traverse. This rule is based upon the following assumptions:

1. The errors in traversing are accidental and, therefore, vary with the square root of the length of the sides, thus making the correction to each side proportional to its length.

2. The effects of the errors in angular measurement are equal to the effects of errors in chaining.

The compass rule is the most commonly used method of adjustment.

** CRANDALL ** (Also known as Method of Least Squares)

The CRANDALL rule assumes that accidental errors in linear measurement are likely to be greater than those in the measurement of angles, as, for example, in stadia traversing, or even careful tape measurements where some of the systematic errors are rendered accidental in nature by reason of corrections and special methods applied to field measurements. This solution meets the desired assumptions and distributes the error of closure in the lengths of the lines only.

The CRANDALL rule should be used if you do not want changes in angles.

** TRANSIT **

The TRANSIT rule states that the correction to be applied to the latitude/departure of any course is to the total correction in latitude/departure as the latitude/departure of that course is to the arithmetic sum at all the latitudes/departures in the traverse. This rule is based on the following assumptions:

1. The errors in traversing are accidental.

2. Angular measurements are more precise than those of chaining.

The transit rule is merely a rule of thumb, which does not apply successfully to many cases. In fact, it meets the assumptions upon which it is based only to the extent that each side is parallel to one or the other coordinate axes.

- - - - - - - - - -

ADJUST ANGULAR ERROR desc nclos cloaz aerr [grid desc2]

Adjust the angular error in the traverse described by desc closing the last point of desc in the direction of point nclos. If the last point in desc is the same as nclos, the unadjusted field closing azimuth is computed from the next to the last point in desc to nclos. The closing azimuth is compared to the record closing azimuth (from north) of cloaz. The error will be printed out and the raw field coordinates of desc will be replaced with coordinates adjusted for angular closure and grid factor. If the correction per angle exceeds the allowable amount of aerr, the angular misclosure will not be adjusted and execution will be terminated unless the input device was specified as the user's terminal. The correction per angle is obtained by dividing the total angular error by the number of angles to be adjusted.

The optional parameter grid is a constant scale factor that can be applied to distances in the traverse to lengthen or shorten them. If grid is not specified, a factor of 1.0 is automatically used. If desc2 is used, grid must be specified.

The optional parameter desc2 is a figure containing the point numbers along the traverse where angles that are not to be adjusted were turned.

NOTE

If the first course of the traverse is also the beginning bearing (or azimuth), then the first point in desc2 is the beginning traverse point.

This page intentionally left blank.

3.5 LOCATE COMMANDS

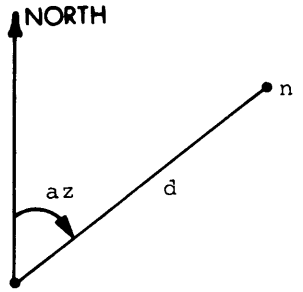
The commands in this section are the simplest of those that compute points. Each command provides a method for locating a single point by extending a given line or curve a given distance.

- - - - - - - - - - -

LOCATE AZIMUTH p n az d

Starting at point p, locate the point n that is at azimuth az and a distance d.

Output: Coordinates of n.

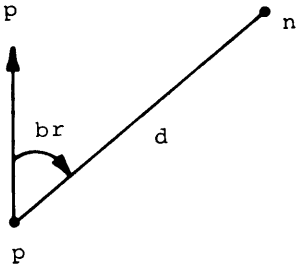


- - - - - - - - - - -

LOCATE BEARING p n br d

Starting at point p, locate the point n that is at bearing br and at a distance d.

Output: Coordinates of n.



Examples for LOCATE AZIMUTH and LOCATE BEARING:

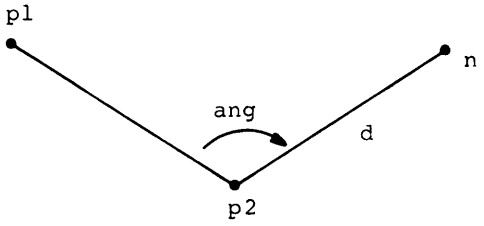
```
DELETE COORDINATES
        1-100
STORE 40 10000 20000
     40
LOCATE AZIMUTH 40 41 45-00-00 100.
     41      10070.7107      20070.7107

41 42 S 45 00 00E 100
     42      10000.0000      20141.4214

42 43 A 41 40 100
     43       9929.2893      20070.7107

LOCATE BEARING 43 44 4 45-00-00 100
     44      10000.0000      20000.0000

40 50 N45 0 0E 200
     50      10141.4214      20141.4214

50 51 A41 43 D41 43
     51      10000.0000      20141.4214

DISTANCE
     51 42
                    FROM   51  TO   42        0.0000 FT.
```

- - - - - - - - - - -

LOCATE ANGLE p1 p2 n ang d

Backsight on p1 while at p2 and locate n at distance d and angle ang. The angle ang can be negative (counterclockwise) or positive (clockwise).

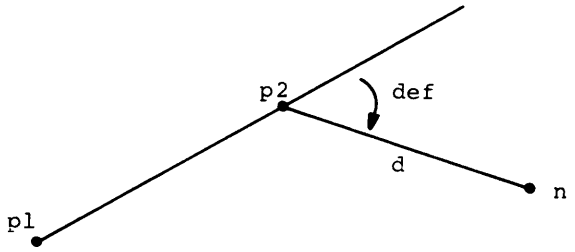
Output: Coordinates of n.



- - - - - - - - - -

LOCATE DEFLECTION pl p2 n def d

From the extension of the path from pl through p2, locate n at distance d and deflection angle def. The deflection def can be negative (to the left) or positive (to the right).

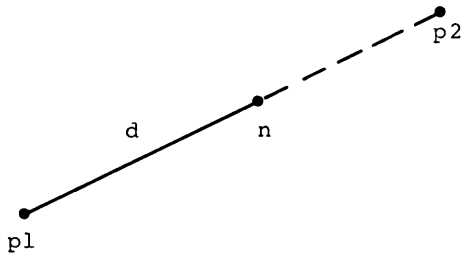


Output: Coordinates of n.

- - - - - - - - - -

LOCATE LINE pl p2 n d

From pl, in the direction p2, locate n at a distance d. A negative value of d locates point n in the direction away from p2.

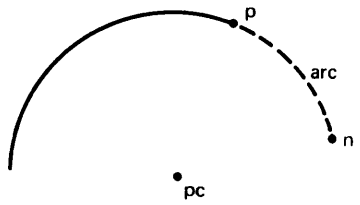


Output: Coordinates of n.

- - - - - - - - - -

EXTEND ARC p pc n arc

Starting from known point p, extend the arc centered at known point pc in a clockwise direction to locate a point n at a distance arc along the arc from p. A negative value for the distance arc causes n to be located in a counterclockwise from p.



Output: Coordinates of n.

Examples for EXTEND ARC, LOCATE ANGLE, LOCATE DEFLECTION, and LOCATE LINE:

```
DELETE COORDINATES
        1-100
STORE 1 200 600
      1
50 1200 600
     50
EXTEND ARC 50 1 200 1570.8
    200        199.9963       1600.0000

50 1 201 -785.4
    201        907.1055       -107.1081

STORE 3 8000 5000
      3
4 7700 5000
      4
LOCATE ANGLE 3 4 8 90 0 0 250
      8       7700.0000       5250.0000

4 8 9 90 0 0 300
```

```
      9       8000.0000       5250.0000

8 9 10  G4 8 9 250
     10       8000.0000       5000.0000

LOCATE DEFLECTION 4 3 12 045 00-00.0 100.0
     12       8070.7107       5070.7107

4 3 13 -045-00-00.0 100.0
     13       8070.7107       4929.2893

4 3 22 60-0-0 200
     22       8100.0000       5173.2051

4 3 23 -G 4 3 22 D 3 22
     23       7900.0000       5173.2051

LOCATE LINE
        3 4 5 500
      5       7500.0000       5000.0000

3 4 6  500
      6       8500.0000       5000.0000
```

Reduction of Slope Distances

A slope distance can be entered instead of a horizontal distance in each of the locate commands if it is followed by the vertical angle. The nearest multiple of 90 degrees is taken as horizontal, so that angles measured from horizon, zenith, or nadir can be used. Thus, the formats of these commands become:

LOCATE AZIMUTH	p	n	az	sd	vang
LOCATE BEARING	p	n	br	sd	vang
LOCATE ANGLE	p1	p2	n	ang	sd vang
LOCATE DIRECTION	p1	p2	n	def	sd vang
LOCATE LINE	p1	p2	n	sd	vang

Output: Coordinates of n.

Examples:

```
DELETE COORDINATES 1-100
STORE 1 0 0
      1
LOCATE AZIMUTH 1 2 0-0-0 100 30-00-00
      2         86.6025          -0.0000

LOCATE ANGLE 1 2 3 90-0-0 100 75-8-30
      3         86.6025         -96.6563

LOCATE LINE 2 3 4 -100 284 51 30
      4         86.6025          96.6563

LOCATE DEFLECTION 2 1 4 45 0 0 100 45 0 0
         PREVIOUS VALUE OF POINT    4 WAS
                86.6025          96.6563
      4        -50.0000         -50.0000
```

This page intentionally left blank.

3.6 INTERSECT COMMANDS

These commands compute the intersections of two lines, a line and a curve, two curves, a line and a figure, a curve and a figure, or two figures. Each time one of these commands is executed, a single point is defined (the point of intersection). If more than one intersection is possible, as in the case of intersecting a line and a curve, COGO computes all possible intersections and the distance from each possible intersection to a point specified by the user. The intersection found closest to or farthest from that point is chosen as the point of intersection. Optimal offset distances can be positive or negative. A positive offset is to the right in a forward direction; a negative offset is to the left in a forward direction.

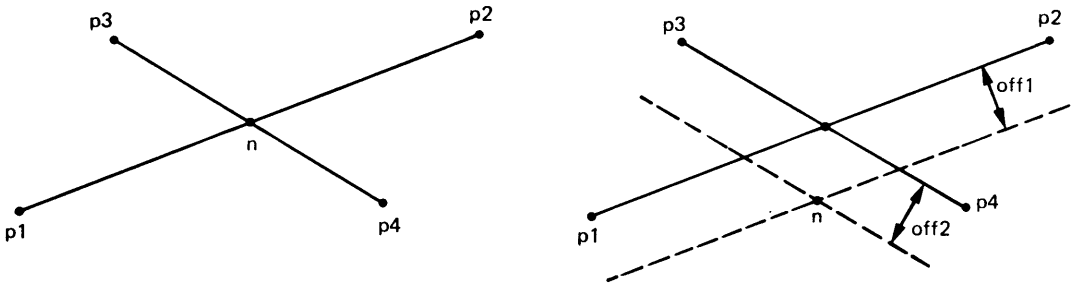
In this section, the commands that work with straight lines or figures assume that all straight line segments extend to infinity. This fact should be kept in mind whenever any of these commands is being used, because unexpected intersections may result.

- - - - - - - - - -

POINTS INTERSECT n p1 p2 p3 p4 [off1 off2]

Find the point n by intersecting the line through p1 and p2 with the line through p3 and p4. A warning message is printed if the angle of intersection is less than 6 degrees. The distances off1 and off2 are optional offset distances.

Output: Coordinates of n



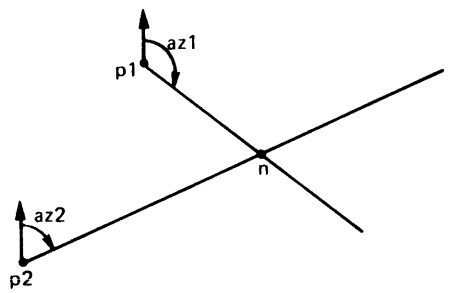
- - - - - - - - - -

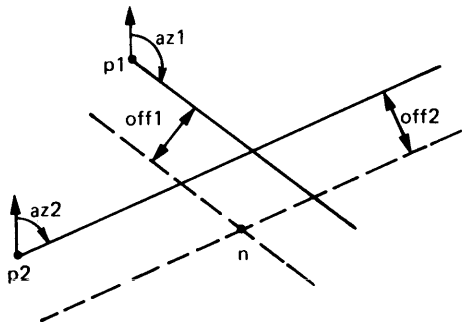
AZIMUTH INTERSECT n p1 az1 p2 az2 [off1 off2]

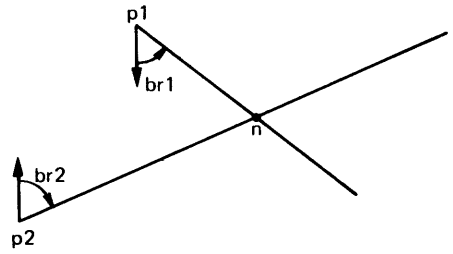
BEARING INTERSECT n p1 br1 p2 br2 [off1 off2]

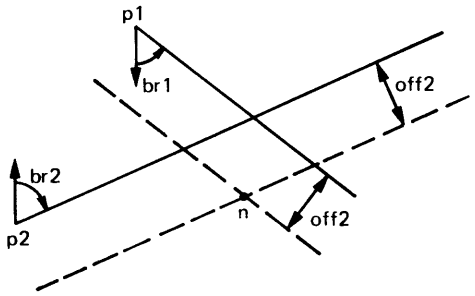
Find the point n by intersecting the line through point p1 at azimuth az1 or bearing br1 with the line through point p2 at azimuth az2 or bearing br2, respectively. A warning message is printed if the angle of intersection is less than 6 degrees. The distances off1 and off2 are optional offset distances.

Output: Coordinates of n.









Examples for AZIMUTH INTERSECT, BEARING INTERSECT, and POINTS INTERSECT:

```
DELETE COORDINATES 1-100
STORE 1 5000 5000
      1
2 5000 4500
      2
AZIMUTH INTERSECT 3 1 10 1 30.1 2 30.0
      3       6248.2124       5220.6558

BEARING INTERSECT 4 1 N 20 W 2 N60 E
      4       5238.5472       4913.1759

POINTS INTERSECT 5 2 1 3 4
      5       5000.0000       4840.5296
```

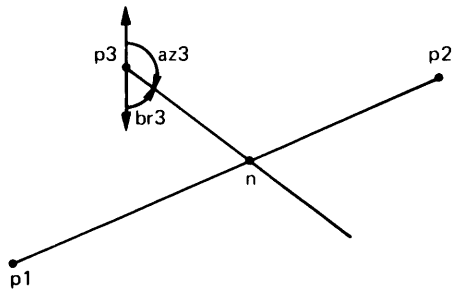
- - - - - - - - - - -

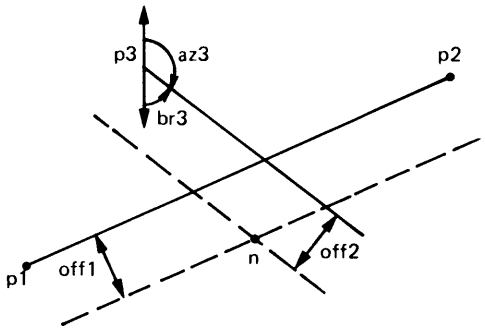
POINTS AZIMUTH INTERSECT n p1 p2 p3 az3 [off1 off2]

POINTS BEARING INTERSECT n p1 p2 p3 br3 [off1 off2]

Find the point n by intersecting the line through p1 and p2 with a line from p3 at azimuth az3 or bearing br3. The distances off1 and off2 are optional offsets.

Output: Coordinates of n.





Examples for POINTS AZIMUTH INTERSECT and POINTS BEARING INTERSECT:

```
POINTS AZIMUTH INTERSECT 6 1 3 2 45.
      6       5607.3691       5107.3691

POINTS BEARING INTERSECT 7 1 3 3 S45W -400
      7       5754.7833       4727.2267
```

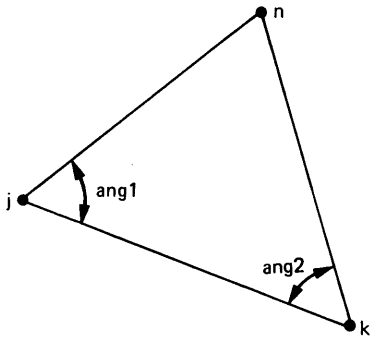
- - - - - - - - - - -

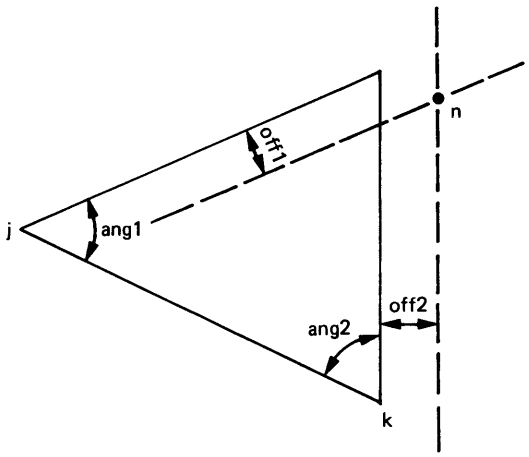
ANGLES INTERSECT n j angl k ang2 [offl off2]

FORESECTION n j angl k ang2 [offl off2]

The ANGLES INTERSECT, or FORESECTION, command is used to locate point n by intersecting a line defined by turning angle angl from base-line point j to point n with a line defined by turning angle ang2 from base-line point k to point n. The optional offset distances, offl and off2 are from line j n and k n, respectively.

Output: Coordinates of the point n.





Examples:

```
STORE 60 100 100
      60
20 550 600
      20
30 890 250
      30
40 1000 500
      40
ANGLES INTERSECT 50 60 25.00 20 75.00
      50        347.8091        911.1946

29 40 36-23-4 20 90-0-0
      29        476.3155        268.4199

FORESECTION 100 30 50-0-0 60 15-0-0
     100       1067.2891        566.5850

95 40 145.5 60 34.95
      95        407.6164        610.1930
```

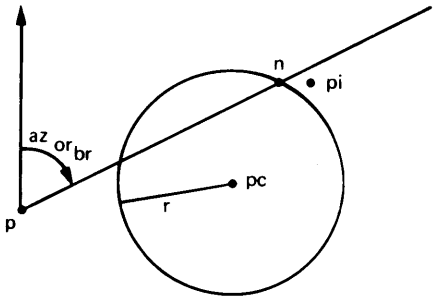
- - - - - - - - - - -

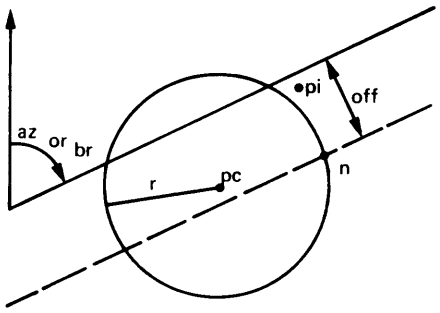
ARC LINE AZIMUTH n pc r p az pi [off]

ARC LINE BEARING n pc r p br pi [off]

Find the point n by intersecting the circle centered at pc having radius r with the line through point p at azimuth az or bearing br. The point of intersection that is closer to or farther from pi depending on whether pi is positive or negative, respectively, is picked as the desired intersection. A warning message is printed if the intersection angle is less than 6 degrees. The distance off is the optional offset distance from the line through p.

Output: Coordinates of n.



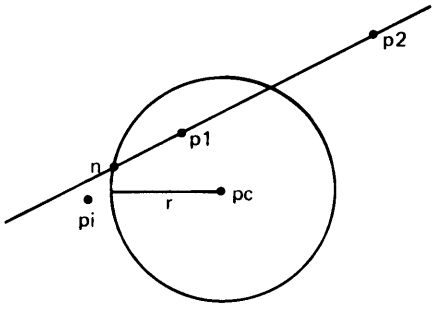


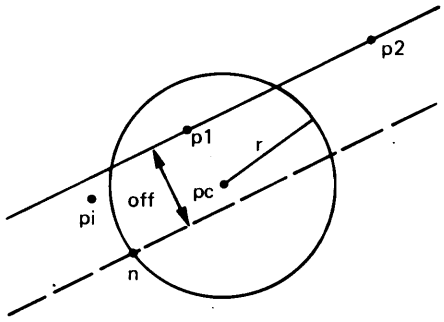
- - - - - - - - - - -

ARC LINE POINTS n pc r pl p2 pi [off]

Find the point n by intersecting the circle centered at point pc having radius r with the line through pl and p2. The intersection that is picked as the desired point of intersection is the one closer to or farther from the point pi depending on whether pi is positive or negative, respectively. The distance off is the optional offset distance.

Output: Coordinates of n.



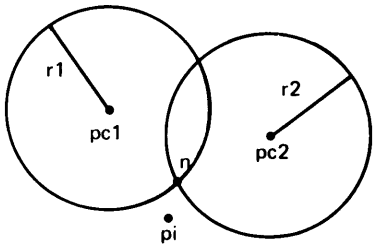


- - - - - - - - - - -

ARC ARC INTERSECT n pcl rl pc2 r2 pi

Find the point n by intersecting the circle centered at pcl and having radius rl with the circle centered at pc2 and having radius r2. The desired intersection is the one closer to or farther from point pi depending on whether pi is positive or negative, respectively. A warning message is printed if the angle of intersection is less than 6 degrees.

Output: Coordinates of n.



Examples for ARC LINE AZIMUTH, ARC LINE BEARING, ARC LINE POINTS, and ARC ARC INTERSECT:

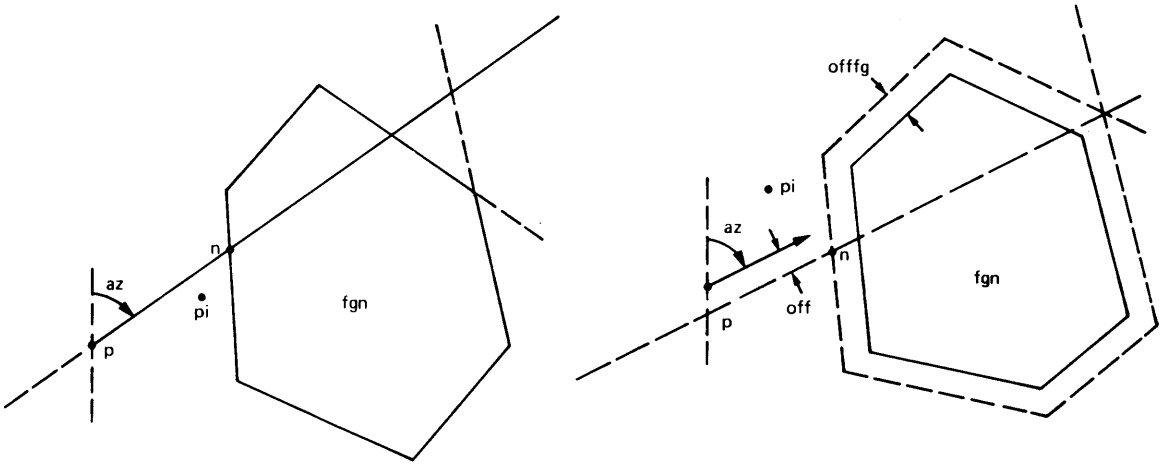
```
STORE 10 0 0
     10
11 900 900
     11
ARC LINE AZIMUTH 15 10 1500 11 -90. 11
     15        900.0000       1200.0000

ARC LINE BEARING 16 10 1500 10 N15E 11
     16       1448.8887        388.2286

ARC LINE POINTS 17 11 1000. 10 16 10
     17        319.6325         85.6453

ARC ARC INTERSECT 18 10 D 10 15 11 500 15
     18        668.1270       1342.9841
```

- - - - - - - - - - -

FIGURE LINE INTERSECT n fgn p az pi [offfq off]

Find the point n by intersecting the line described by figure fgn with a line through point p at azimuth az. The intersection is the one closer to or farther from the point pi depending on whether pi is positive or negative, respectively. The intersection can be made with a line parallel to figure fgn by specifying the offset distance offfg. The intersection can be made with a line parallel to the line through point p by specifying the offset distance off. The user should keep in mind that both ends of both the line and figure extend to infinity.

Output: Coordinates of n.

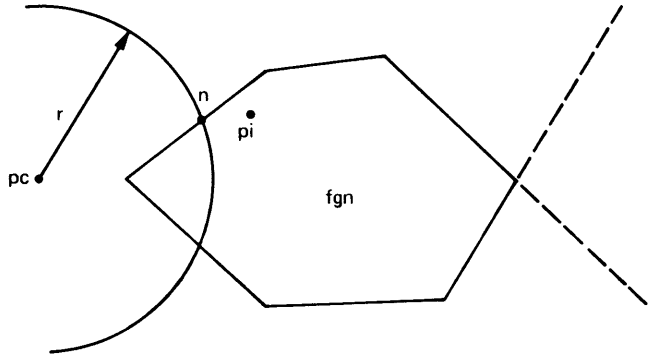


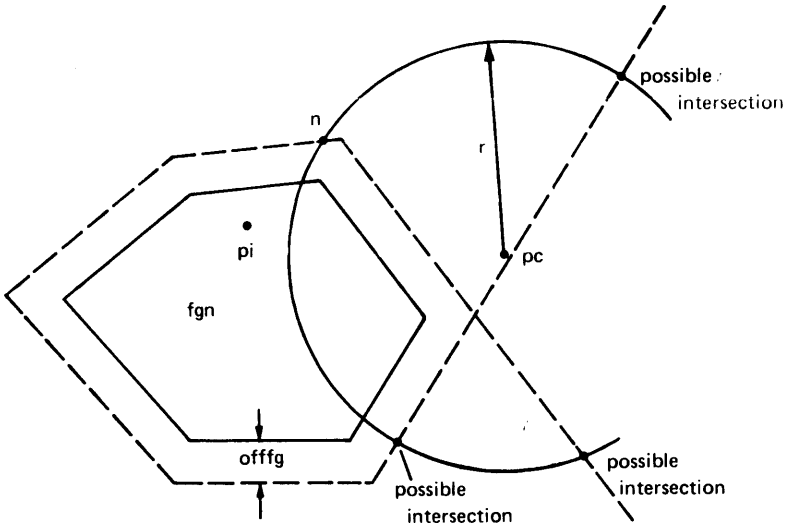
- - - - - - - - - - -

FIGURE ARC INTERSECT n fgn pc r pi [offfq]

Find the point n by intersecting the line described by figure fgn with the circle having center at pc and radius r. The intersection is the one closer to or farther from the point pi depending on whether pi is positive or negative. Users should keep in mind that both ends of the figure extend to infinity.

Output: Coordinates of n.



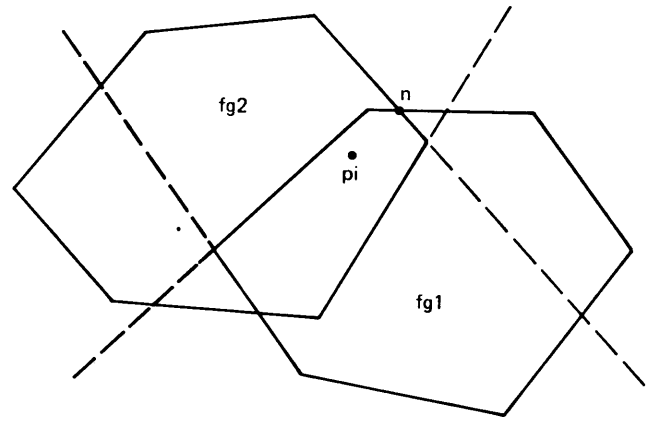


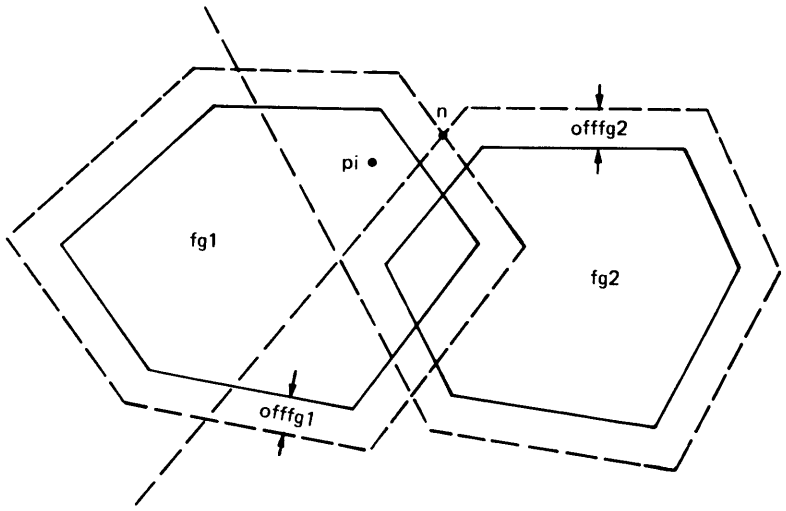
- - - - - - - - - - -

FIGURE FIGURE INTERSECT n fg1 fg2 pi [offfg1 offfg2]

Find the point n by intersecting the line described by figure fg1 with the line described by figure fg2. The intersection is the one closer to or farther from the point pi depending on whether pi is positive or negative. The intersection can be made with a line parallel to figure fg1 by specifying the offset distance offfg1 and also with a line parallel to figure fg2 by specifying the offset distance offfg2. Users should keep in mind that both ends of both figures extend to infinity.

Output: Coordinates of n.





Examples for FIGURE LINE INTERSECT, FIGURE ARC INTERSECT, and FIGURE FIGURE INTERSECT:

```
DELETE FIGURES
        1-99
DELETE COORDINATES
        1-9999
STORE FIGURE
        1 (1-4)
                  1
STORE 1  400  400
        1
2  800  100
        2
3  800  800
        3
4  400  800
        4
6  300  600
        6
STORE FIGURE
        2(5-8)
                  2
 CONVERT MERIDIAN 1  2  2  6  -90.
                                    SHIFT = 135- 0- 0.0          707.1068
                            CONVERSION ANGLE = 270- 0- 0.0
                                SCALE FACTOR = 0.100000000E+01

FIGURE LINE INTERSECT 11 1 4 90. 4
          INTERSECTION IS AT SEGMENT     3     4
     11        300.0000        800.0000

FIGURE ARC INTERSECT 12 1 5 400. 3
          INTERSECTION IS AT SEGMENT     2     3
     12        800.0000        653.5898

FIGURE FIGURE INTERSECT 13 1 2 4 0. 100.
              INTERSECTION IS AT SEGMENTS     3     4 AND     5     6
     13        575.0000        800.0000
```

This page intentionally left blank.

3.7 ALIGNMENT COMMANDS

The commands in this section provide the capability of working with an entire alignment (including any number of straight lines and circular curves). In each of the commands the alignment is specified by the first three items of data.

The first item is the figure number or description to be used as the alignment. The alignment must be described in ascending order of stations, (for example, the second point in the description must have a higher station than the first, and so forth). If this is not true, then the figure number or description should be followed by the letter T indicating that the description is the transposed description of the one specified.

The second item is the point number used to define the stationing. This may be any point.

The third item is the station of the point defined in the second item.

These commands consider the alignment to be extended along a tangent of each end when stations are outside the range of the alignment as specified.

- - - - - - - - - -

STATIONS AND OFFSETS descl p ps desc2

This command computes and prints the station and offsets to alignment descl whose stationing is defined by point p, which has known station ps. The station and offset of each point of desc2 are computed.

Examples:

```
STAS AND OFFS (1 2) 1 1000 (5)

                                *Print the station and offset
                                *to the alignment from
                                *point 1 to 2 of
                                *point 5. (The station of
                                *point 1 is 1000.)

STORE FIGURE 465 (70 75 C3R-9)
STORE FIGURE 8 (200-256)
STATION AND OFFSETS 465 32 1423.6 8

                                *Print the station and offset
                                *to the alignment described
                                *in figure 456,
                                *whose stationing is defined by
                                *point 32 having station
                                *1423.6, of each point in
                                *figure 8.

STATIONS AND OFFS 465 32 1423.6 (490 2)

                                *print the stations and
                                *off-sets to the same alignment
                                *as above of points 490 and 2.
```

```
DELETE COORDINATES
         1-9999
DELETE FIGURES
         1-99
STORE 1 1000 1000
       1
2 400 600
       2
5 750 750
       5
STATIONS AND OFFSETS (1 2) 1 1000 (5)
         SEG    1    2     PNT,STA,OFF    5     1346.6877        69.3376
```

- - - - - - - - - -

POINTS ON ALIGNMENT desc p psl d nl [off ps2 ps3]

This command computes points at fixed intervals along an alignment. The alignment is specified by figure number or description descl. The stationing is specified by point p (any point with known station) and psl (the station of point p). Points are computed along the alignment at each position that has a station that is evenly divisible by the station interval d (feet) and optional offset off. The computed points are stored in the table as point numbers nl, nl+1..., nl+2,..., and so forth. Optional information can be entered to specify that points should be computed only on that portion of the alignment that falls between stations ps2 and ps3.

NOTE

> Choose nl carefully so that no known points are destroyed by the assignment of point numbers nl, nl+1, etc. to the computed points.

Output: Coordinates of the intermediate points.

Examples:

```
STORE FIGURE 1 (1-5)
              1
POINTS ON ALIGNMENT (1 2) 5 1000 100. 11
     11        961.1535        974.1024    SEG    1    2     STA    700.

     12        877.9485        918.6323    SEG    1    2     STA    800.

     13        794.7435        863.1623    SEG    1    2     STA    900.

     14        711.5385        807.6923    SEG    1    2     STA   1000.

     15        628.3334        752.2223    SEG    1    2     STA   1100.

     16        545.1284        696.7523    SEG    1    2     STA   1200.

     17        461.9234        641.2822    SEG    1    2     STA   1300.
```

- - - - - - - - - -

LOCATE FROM ALIGNMENT desc p psl n ps2 [off]

The alignment is specified by figure number, or description desc with stationing defined by point p having station psl. Point n is located from the alignment at station ps2 and optional offset off.

Output: Coordinates of n.

Examples:

```
DELETE FIGURES 1-5
STORE FIGURE 1 (1 2 5)
               1
LOCATE FROM ALIGNMENT
            1 1 0 18 157.3
     18       869.1185       912.7457
```

- - - - - - - - - - -

ALIGNMENT OFFSET desc pl psl n ps2

This command is used to locate a point on an alignment from an offset point. The alignment is specified by figure number or description desc with stationing defined by point pl having station psl. Point n is located on the alignment by drawing a perpendicular from point ps2 to the alignment.

Output: Coordinates of n.

```
DELETE FIGURES 1
STORE 3 1400 600
      3
4 800 100
      4
6 800 600
      6
STORE FIGURE
        1(3 1 C6L 2 4)
                1
ALIGNMENT OFFSET
        1 1 1000. 20 5
         SEG    3    1     PNT,STA,OFF    5     1000.0000      353.5534

     20      1000.0000      1000.0000
```

- - - - - - - - - -

DESCRIBE ALIGNMENT AZIMUTHS desc pl psl [pol ps2 ps3]

DESCRIBE ALIGNMENT BEARINGS desc pl psl [pol ps2 ps3]

Describe the alignment specified by figure number or description desc. The stationing is specified by point pl and its station psl. Coordinates and stations of all PI's, PC's, PT's, coordinates of CC'S, azimuths or bearings of all tangents, and deflection angles at PI's are listed. Curve data for curves (radius, degree, tangent lengths, external, arc length, and so forth) are also listed.

If pol is not zero, compute and print coordinates and station for every point that has a station evenly divisible by pol.

If stations ps2 and ps3 are specified, describe only the part of the alignment between stations ps2 and ps3.

Example for DESCRIBE ALIGNMENT AZIMUTHS:

```
DESCRIBE ALIGNMENT AZIMUTHS 1 1 1000
      3       1400.0000        600.0000          434.3146
                                                          135- 0- 0.0

PC    1       1000.0000       1000.0000         1000.0001   161-33-54.2 L
                                                          333-26- 5.8
CC    6        800.0000        600.0000    RAD=  400.0000   243-26- 5.8 L

                  DEGREE= 14-19-26.2        L= 1699.4966

                                                           90- 0- 0.0
PT    2        400.0000        600.0000         2699.4966   141-20-24.7 L
                                                          308-39-35.3

      4        800.0000        100.0000         3339.8090

1 1 500
      3       1400.0000        600.0000          -65.6855
                                                          135- 0- 0.0

PC    1       1000.0000       1000.0000          500.0000   161-33-54.2 L
                                                          333-26- 5.8
CC    6        800.0000        600.0000    RAD=  400.0000   243-26- 5.8 L

                  DEGREE= 14-19-26.2        L= 1699.4966

                                                           90- 0- 0.0
PT    2        400.0000        600.0000         2199.4966   141-20-24.7 L
                                                          308-39-35.3

      4        800.0000        100.0000         2839.8090
```

Example for DESCRIBE ALIGNMENT BEARINGS:

```
DESCRIBE ALIGNMENT BEARINGS 1 1 1000
      3       1400.0000        600.0000          434.3146
                                                          S 45- 0- 0.0 E

PC    1       1000.0000       1000.0000         1000.0001    161-33-54.2 L
                                                          N 26-33-54.2 W
CC    6        800.0000        600.0000    RAD=  400.0000    243-26- 5.8 L

                  DEGREE= 14-19-26.2        L= 1699.4966

                                                          S 90- 0- 0.0 E
PT    2        400.0000        600.0000         2699.4966    141-20-24.7 L
                                                          N 51-20-24.7 W

      4        800.0000        100.0000         3339.8090

1 1 500
      3       1400.0000        600.0000          -65.6855
                                                          S 45- 0- 0.0 E

PC    1       1000.0000       1000.0000          500.0000    161-33-54.2 L
                                                          N 26-33-54.2 W
CC    6        800.0000        600.0000    RAD=  400.0000    243-26- 5.8 L

                  DEGREE= 14-19-26.2        L= 1699.4966

                                                          S 90- 0- 0.0 E
PT    2        400.0000        600.0000         2199.4966    141-20-24.7 L
                                                          N 51-20-24.7 W

      4        800.0000        100.0000         2839.8090
```

The following commands are used to define and solve the geometry associated with an alignment that includes simple curves, tangents, offsets, and stationing along the line. All curves are circular, and stations are expressed in decimal feet (station 41 + 23.67 is entered as 4123.67).

An ALIGNMENT or DEFINE CURVE command must precede any of the following commands:

COORD POA

COORD OFFSET

STATION FROM COORD

OFFSET ALIGN

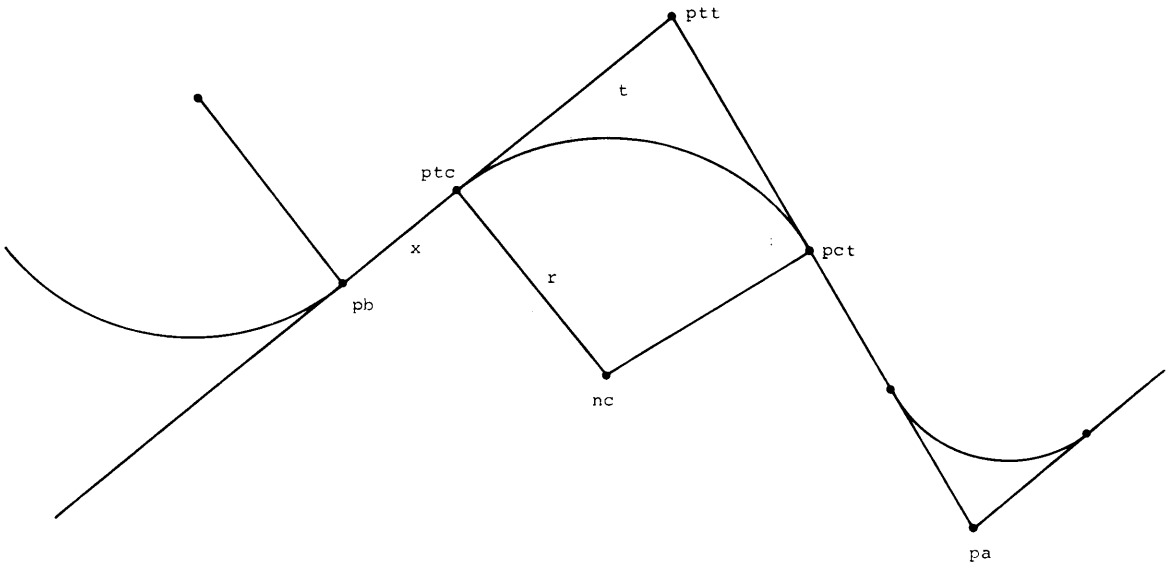
The ALIGNMENT or DEFINE CURVE command establishes a circular curve in the computer memory by storing its parameters. Only one set of parameters can be stored at one time. All of the above commands then pertain to that curve. If another ALIGNMENT or DEFINE CURVE command is given, a new curve is established and the above commands refer to the new curve; the original curve is removed from the computer memory and must be reentered for further calculation.

If all of the curve data are known, the DEFINE CURVE command should be used. If the curve in question has unknown quantities, the ALIGNMENT command must be used.

The established curve remains in memory until replaced by a subsequent ALIGNMENT or DEFINE CURVE command.

- - - - - - - - - -

ALIGNMENT cid pb ptt pa ntc nc nct r t sb x



Compute the curve, given the following:

cid Curve identification (0-999).

pb Any known point on the back tangent.

ptt Predefined point of intersection of the tangents.

pa Predefined point anywhere on the ahead tangent.

ntc New point number assigned to the beginning of the curve, that is, the transition from tangent to curve.

nc Number assigned to center of curve.

nct New point number assigned to the end of the curve, that is, the transition from curve to tangent.

r Radius of curve (if unknown, use 0).

t Tangent length of curve (if unknown, use 0).

sb Station at pb. If entered as -1., pb is taken to be and must be entered as ct of the previous curve, and sb is taken as the station of the previous ct (that is, x = 0). This allows stationing to be automatically carried forward (see example).

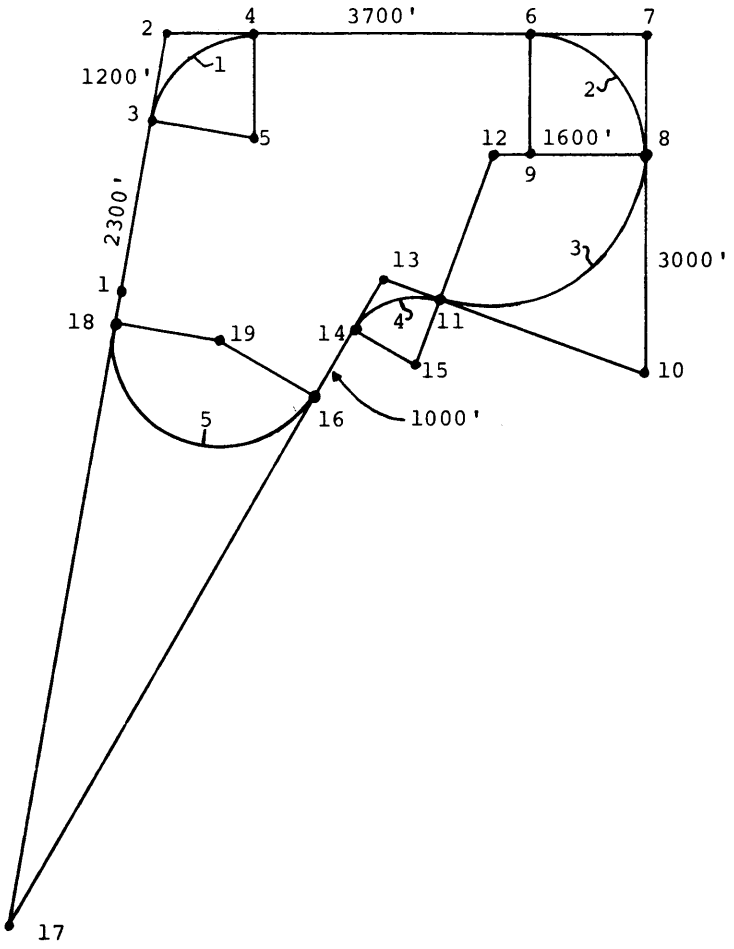
x Fixed distance from pb to tc. If r and t are unknown (0.) and x = 0., the curve is compounded or reversed with the previous curve. If r and t are unknown (0.) and x = 150., the curve is computed such that the tc is 150. feet from pb (usually but not necessarily the ct of the previous curve).

All tangent intersections (designated tt) should be located by using LOCATE or other commands before using ALIGNMENT command to compute and station the alignment. If x = 0., t = 0., and r = 0., the pb must be the same point number as ntc.

NOTES

- Where a distance or length is unknown, the value 0. must be used.
- Any one of the four values r, t, sb, and x specifies the curve completely. To prevent contradictory overspecification of a curve, COGO uses the first of these values that is nonzero, and disregards the others.

Output: cid, sign (1. if to right, -1. if to left), radius, tangent length, deflection angle, x (distance pb to tc), station of tc, curve length, station of ct, coordinates of tc, ct, and c (center of curve).



	cid	pb	ptt	pa	ntc	nc	nct	r	t	sb	x
ALIGNMENT	1	1	2	6	3	5	4	0.	1200.	0.	2300.
	2	4	7	10	6	9	8	1600.	0.	-1.	0.
	3	8	10	13	8	12	11	0.	0.	-1.	0.
	4	11	13	16	11	15	14	0.	0.	-1.	0.
	5	14	17	1	16	19	18	0.	0.	-1.	1000.

Curve 1 is determined by defining its tangent length and distance from tc to point 1 on the back tangent.

Curve 2 is determined by defining its radius.

Curve 3 is a compound curve.

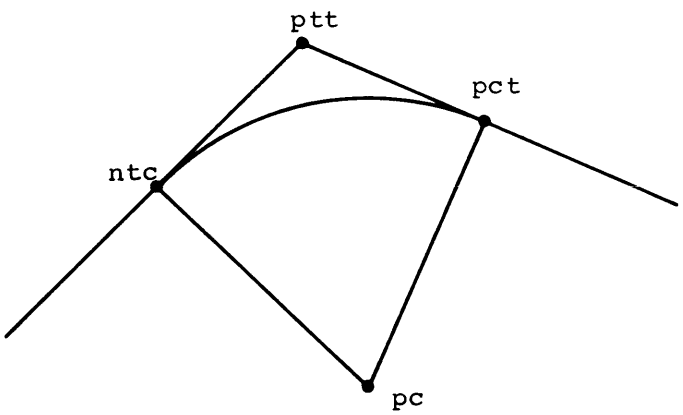
Curve 4 is a reverse curve.

Curve 5 is determined by defining the distance from tc to the pb of the previous curve.

- - - - - - - - - -

DEFINE CURVE cid ntc stc ptt pct sct pc sign

Output: None.



cid Curve identification number (0-999).

ntc New point number assigned to the beginning of the curve, that is, the transition from tangent to curve.

stc Station of the tc.

ptt Predefined point at the point of intersection of the tangents.

pct Predefined point at the end of the curve, that is, the transition from curve to tangent.

sct Station of the ct.

pc Number of the predefined center of the curve.

sign 1. for clockwise curve (from tc to tt).
-1. for counterclockwise curve (from tc to tt).

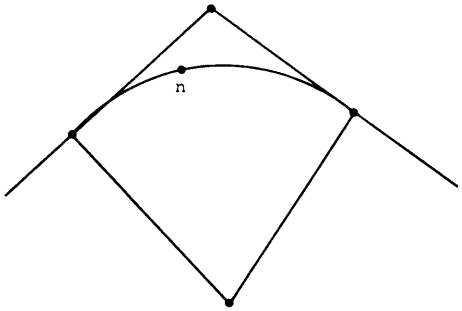
Each of the four following routines automatically selects the back tangent, curve section, or forward tangent, whichever is appropriate.

- - - - - - - - - -

COORDINATE POA n s

Compute the COORDINATEs of Point n On the Alignment at station s.

Output: Coordinates of n.



NOTE

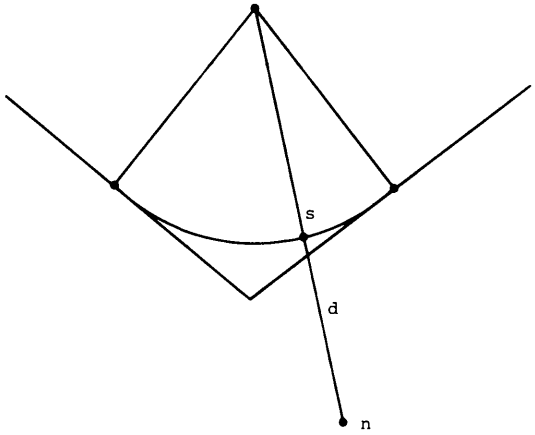
The curve must have been previously defined by a DEFINE CURVE or ALIGNMENT command.

- - - - - - - - - -

COORDINATE OFFSET n s d

Compute the coordinates of point n at station s and offset distance d. If d is positive, n is to the right of the curve when looking from the back to the ahead tangent.

Output: Coordinates of n.



NOTE

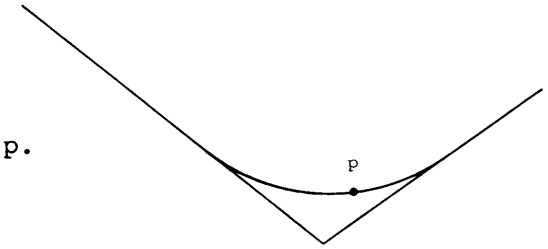
The curve must have been previously defined by a DEFINE CURVE or ALIGNMENT command.

- - - - - - - - - -

STATION FROM COORDINATES p

Compute the station of previously defined point p on the alignment.

Output: Point number and station of p.



NOTES

1. This routine is useful in stationing any number of points located by intersections with the center line.

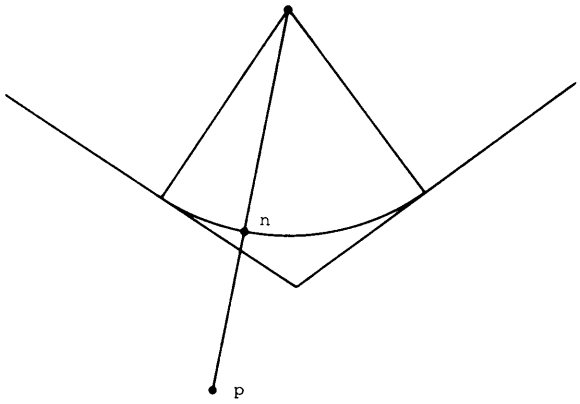
2. The curve must have been previously defined by a DEFINE CURVE or ALIGNMENT command.

- - - - - - - - - - -

OFFSET ALIGNMENT n p

Locate point n at the intersection with the alignment of the radial offset from predefined point p.

Output: n, p, station, offset (negative if to the left) and coordinates of n.



NOTES

1. If a line segment or its extension drawn from the center of the circle to p does not intersect the alignment in the circular portion, the offset is from p perpendicular to the line of the back tangent.

2. The curve must be previously defined by a DEFINE CURVE or ALIGNMENT command.

3.8 SPIRAL COMMANDS

The following group of commands introduces spirals to the geometry (that is, Normal Highway Transition Spiral[1]). As in the preceding group, these commands are used to define and solve the geometry with an alignment. In all of the following commands, this alignment contains a spiral. The transition from a straight line to a circle by use of a spiral is called "spiral in" (to the circle). The transition spiral from the circle to the straight line is called "spiral out" (from the circle).

- - - - - - - - - -

SIMPLE SPIRAL cid pb pts ntt nsc +ls dc sign

SIMPLE SPIRAL cid pb nts ntt psc -ls dc sign

From the given set of defining parameters, calculate the other parameters associated with a simple spiral.

The defining parameters are as follows:

cid Curve identification (0-999).

pb Predefined point anywhere on the back tangent.

pts Predefined point at ts, the point of change from tangent to spiral.

nts New point number assigned to the transition from tangent to spiral for spiral out.

ntt New point number assigned to the pi, the point of intersection of the spiral tangents.

nsc New point number assigned to the point of transition from spiral to circular curve for spiral in.

psc Predefined point at the point of transition from spiral to circular for spiral out.

ls Length of spiral (negative for spiral out), measured along the spiral from ts to sc.

dc Degrees of curvature of circular curve, defined as the central angle (in decimal degrees) that subtends a 100-foot arc.

sign 1. for spiral clockwise.
-1. for spiral counterclockwise.

NOTE

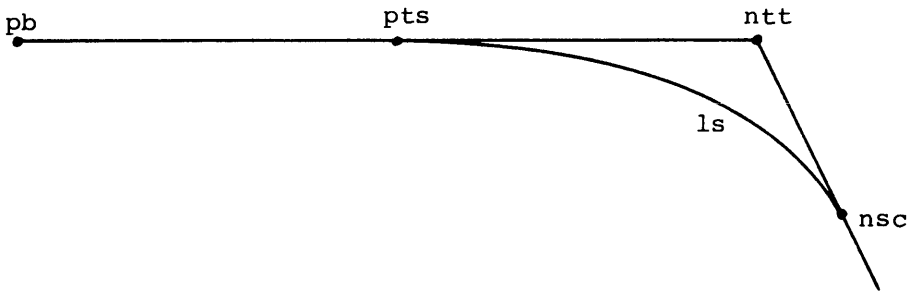
Direction is taken from ts to sc for spiral in and from sc to ts for spiral out.

---

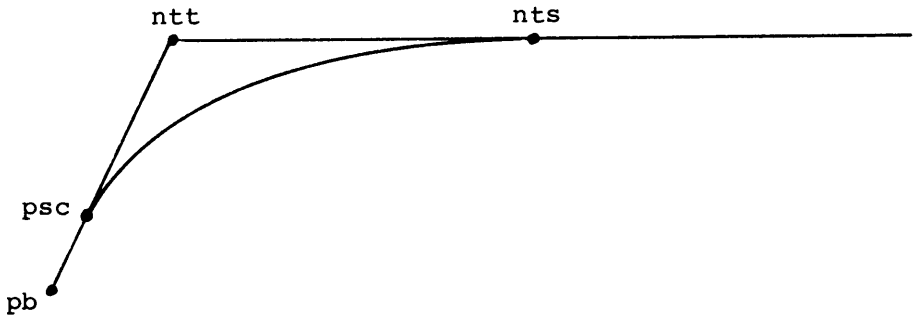
[1] For equation see Thomas F. Hickerson, ROUTE, SURVEYS & DESIGN, McGraw-Hill.

Output: Curve number, sign, spiral length, degree of curvature of circular curve, deflection angle at spiral tt, long tangent length and azimuth, short tangent length and azimuth, coordinates of ts, tt, and sc.

For spiral in:



For spiral out:



Examples for clockwise spiral in and spiral out:

```
DELETE COORDINATES
         1-9999
DELETE FIGURES
        1-99
*                                                          SIMPLE SPIRAL
STORE 1 150 180  * PB
      1
2 200 200  * PTS
      2
SIMPLE SPIRAL 0 1 2 3 4 200. 120. 1.  * SPIRAL IN
  SPIRAL    0 SIGN= 1. L=       200.0000 DC=120- 0- 0.0 DEFLN ANG=120- 0- 0.0

  LONG TAN L=        186.8935 AZ=  21- 48-  5.1

  SHORT TAN L =       117.2215 AZ= 141- 48-  5.1

      2        200.0000        200.0000

      3        373.5262        269.4105

      4        281.4052        341.8990

1 1 6 5 4 -200. 120. 1.  * SPIRAL OUT
  SPIRAL    1 SIGN= 1. L=       200.0000 DC=120- 0- 0.0 DEFLN ANG=120- 0- 0.0

  LONG TAN L=        186.8935 AZ= 170- 56-  7.9

  SHORT TAN L =       117.2215 AZ=  50- 56-  7.9

      6        170.7180        462.3584

      5        355.2775        432.9140

      4        281.4052        341.8990
```

Example for counterclockwise spiral in and spiral out:

```
STORE
    7 150 560
      7
8 200 660
      8
SIMPLE SPIRAL
       2 7 8 9 12 200. 120. -1.
  SPIRAL    2 SIGN=-1. L=       200.0000 DC=120- 0- 0.0 DEFLN ANG=120- 0- 0.0

  LONG TAN L=        186.8935 AZ=  63- 26-  5.8

  SHORT TAN L =       117.2215 AZ= 303- 26-  5.8

      8        200.0000        660.0000

      9        283.5813        827.1626

     12        348.1691        729.3399

3 7 10 11 12 -200. 120.  1.
  SPIRAL    3 SIGN=-1. L=       200.0000 DC=120- 0- 0.0 DEFLN ANG=120- 0- 0.0
```

```
LONG TAN L=        186.8935 AZ= 280- 30- 52.6

SHORT TAN L =      117.2215 AZ=  40- 30- 52.6

   10        471.3911        621.7367

   11        437.2856        805.4919

   12        348.1691        729.3399
```

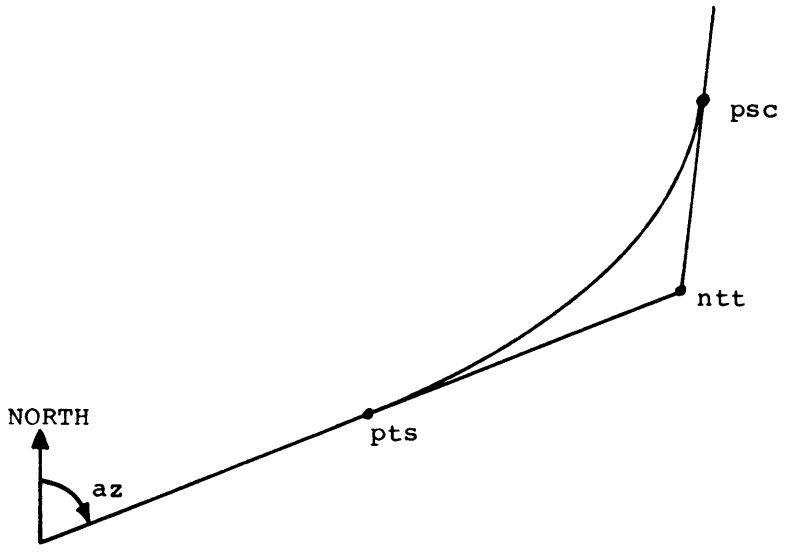
- - - - - - - - - - -

SPIRAL LENGTH cid pts psc ntt az sign

Compute a spiral in using the following parameters:

cid	Curve identification number (0-999).
pts	Predefined point at ts, the point of change from tangent to spiral.
psc	Predefined point at sc, the point of change from spiral to circle.
ntt	New point number assigned to tt, the point of intersection of spiral tangents.
az	Azimuth of tangent at ts.
sign	1. clockwise spiral -1. counterclockwise spiral.

Output: Curve number, spiral length from ts to sc, degree of curvature at sc (the central angle that subtends a 100-foot arc), tangent lengths and their azimuths, coordinates of ts, tt, and sc.



Example:

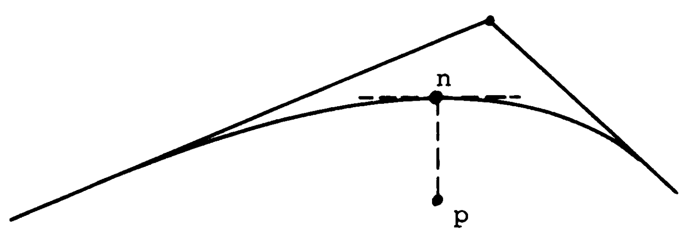
```
STORE
    13 450 180
     13
14 420 400
     14
SPIRAL LENGTH 999 13 14 15 90. 1.
  SPIRAL  999 L=       223.6797 DC= 20-51-30.4

  LONG TAN  L=       150.4345 AZ=  90-  0-  0.0

  SHORT TAN L=        75.7586 AZ= 113- 19- 40.9

     13        450.0000        180.0000

     15        450.0000        330.4345

     14        420.0000        400.0000

998 13 14 16 106. 1.
  SPIRAL  998 L=       223.8862 DC= 22- 6-12.5

LONG TAN  L=       150.7409 AZ= 106-  0-  0.0

SHORT TAN L=        75.9816 AZ=  81- 15- 24.1

   13        450.0000        180.0000

   16        408.4502        324.9014

   14        420.0000        400.0000
```

\- - - - - - - - - -

SPIRAL OFFSET n p

Find point n on the previously defined SIMPLE SPIRAL or SPIRAL LENGTH corresponding to an offset to the spiral from point p.

Output: Coordinates of point n, arc length along spiral from ts to point n, and offset distance from p to n.



Example:

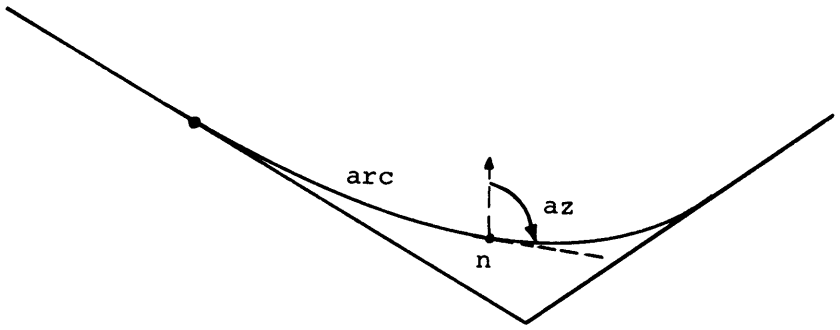
```
STORE 18 410 320
    18
SPIRAL OFFSET 17 18
    17        418.6070        320.8575

  ARC FROM  13 TO  17=       144.5213 OFFSET=          8.6496
```

- - - - - - - - - -

COORDINATE POSP n arc

Use the COORDINATE Point On SPiral to locate point n on the previously defined SIMPLE SPIRAL or SPIRAL LENGTH a distance arc from ts, measured along the curve.

Output: Point number n and its coordinates, spiral number, arc length along the spiral, and azimuth of the tangent at point n.



Example:

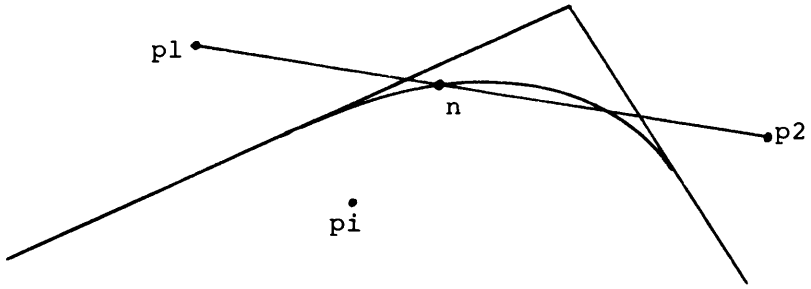
```
COORDINATE POSP
          19 100.
     19        425.2158        276.8460

  SPIRAL    998 ARC=       100.0000 AZIMUTH=101- 3-49.3
```

- - - - - - - - - -

LINE SPIRAL n p1 p2 pi

Find the intersection point n of the line defined by the points p1 and p2 and the previously defined SIMPLE SPIRAL or SPIRAL LENGTH. If two intersections are found, point n is the point closest to pi.

Output: Coordinates of point n and arc length from ts to n.



Example:

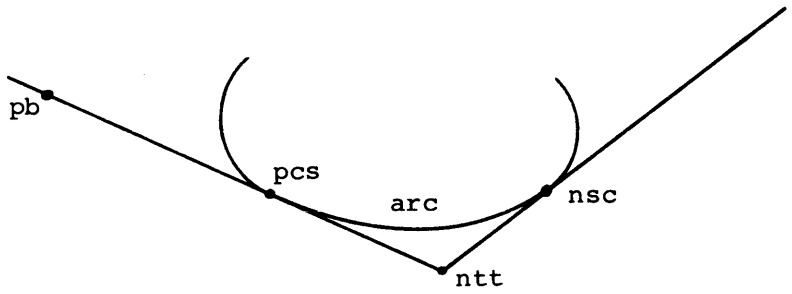
```
STORE 20 500 400
     20
21 460 220
     21
LINE SPIRAL
      22 18 20  21
     22        418.0316        327.1393

  ARC FROM PT.  13 TO PT.  22=       150.8295
```

- - - - - - - - - - -

COMPOUND SPIRAL cid pb pcs ntt nsc arc dc1 dc2 sign

cid	Curve identification (0-999).
pb	Predefined point anywhere on the back tangent.
pcs	Predefined point at cs, the point of change from curve to spiral.
ntt	New point number assigned to tt, the point of intersection of the two tangents.
nsc	New point number assigned to the sc, the point of change from spiral to curve.
arc	Arc length of compound spiral, measured from cs to sc.
dc1	Degree of curvature of circular curve 1 (the central angle, in decimal degrees, which subtends a 100-foot arc).
dc2	Degree of curvature of circular curve 2.
sign	1. for clockwise curve. -1. for counterclockwise curve.

Output: Curve number, coordinates of cs, tt, and sc, tangents lengths (cs to tt and tt to sc) and their azimuths.



Example:

```
STORE 23 600 180
     23
24 700 450
     24
COMPOUND SPIRAL 500 23 24 25 26 400. 80. 140. 1.
  COMPOUND SPIRAL  500

     24        700.0000        450.0000

     25        708.2590        472.2994

     26        675.7631        491.3061

  TANGENTS T1=        23.7798 T2=        37.6463

  BACK AZ=  69- 40- 36.7

  FORWARD AZ= 149- 40- 36.7

500 23 24 27 28 400. 80. 140. -1.
  COMPOUND SPIRAL  500

     24        700.0000        450.0000

     27        708.2590        472.2994

     28        745.2959        465.5532

  TANGENTS T1=        23.7798 T2=        37.6463

  BACK AZ=  69- 40- 36.7

  FORWARD AZ= 349- 40- 36.7
```

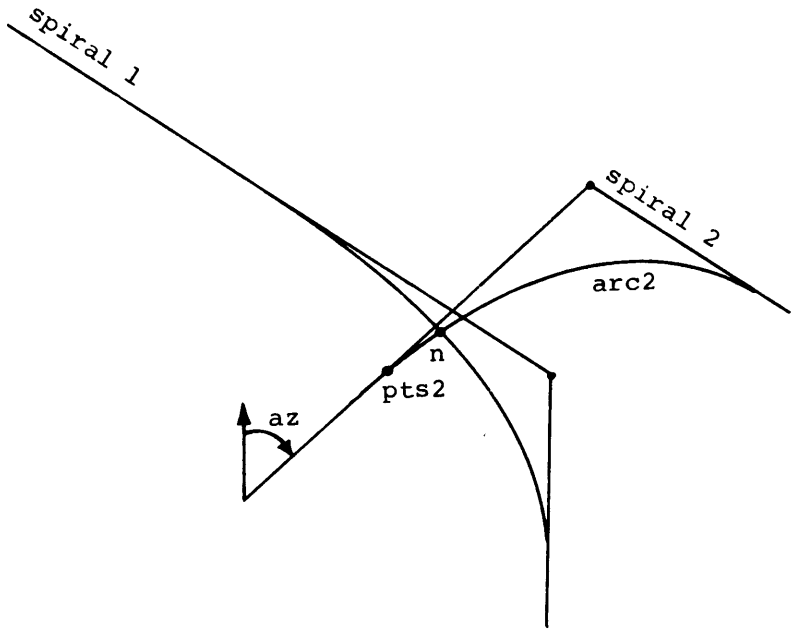
- - - - - - - - - - -

SPIRAL SPIRAL n pts2 az arc2 r2

Locate the intersection point n of the previously defined spiral, SPIRAL 1, and a second spiral, SPIRAL 2. SPIRAL 1 must be defined previously by a SIMPLE SPIRAL or SPIRAL LENGTH command.

Input:
- n — New point number assigned to the point of intersection.
- pts2 — Predefined transition point from tangent to SPIRAL 2.
- az — Azimuth of back tangent for SPIRAL 2.
- arc2 — Arc length of SPIRAL 2 (from ts2 to sc2).
- r2 — Radius of circular curve of SPIRAL 2. Radius is positive for SPIRAL 2 clockwise. Radius is negative for SPIRAL 2 counterclockwise.

Output: Point of intersection n and its coordinates.
Distance from tsl to the point of intersection n.
Distance from ts2 to the point of intersection n.



Example:

```
STORE 29 600 650
     29
30 550 500
     30
SIMPLE SPIRAL 4 29 30 31 32 200. 100. -1.
  SPIRAL    4 SIGN=-1. L=       200.0000 DC=100- 0- 0.0 DEFLN ANG=100- 0- 0.0
```

```
  LONG TAN L=        163.5266 AZ= 251- 33- 54.2

  SHORT TAN L =       94.8019 AZ= 151- 33- 54.2

     30        550.0000        500.0000

     31        498.2884        344.8651

     32        414.9236        390.0060

STORE 33 550 650
     33
34 600 480
     34
SPIRAL SPIRAL
          35 33 120. 200 -80
     35        585.4700        577.6292

  DIST FROM  30 TO INTERSECTION POINT IS       -85.7443

  DIST FROM  33 TO INTERSECTION POINT IS       -80.7453
```

- - - - - - - - - - -

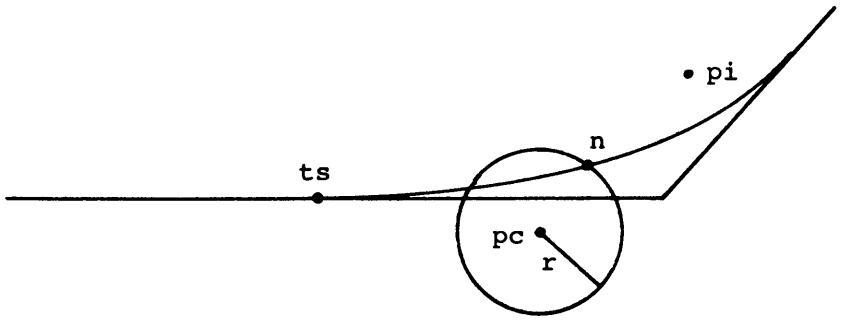
CURVE SPIRAL n r pc pi

Find the intersection point n of the previously defined spiral (defined by SIMPLE SPIRAL (in) or SPIRAL LENGTH) and the circular curve defined by center pc and the radius r. If more than one intersection is found, point n is the point closest to pi.

Input:

n	New point number assigned to the point of intersection of the spiral and the curve closest to p.
r	Radius of circular curve.
pc	Center of circular curve.
pi	Any point.

Output: Coordinates of point n and the distance from ts to n.



Example:

```
SPIRAL LENGTH 998 13 14 16 106. -1.
  SPIRAL  998 L=       223.8862 DC= 22- 6-12.5

  LONG TAN  L=       150.7409 AZ= 106-  0-  0.0

  SHORT TAN L=        75.9816 AZ=  81- 15- 24.1

     13        450.0000        180.0000

     16        408.4502        324.9014

     14        420.0000        400.0000

STORE 36 430 300  * PC
     36
37 410 360  * PI
     37
CURVE SPIRAL 38 80 36 37
     38        417.6168        379.0358

          ARC FROM   13 TO   38  H =       202.7819
```

- - - - - - - - - - -

FIT ALIGNMENT cid pb ptt dc arcl arc2 def sign

Calculate the alignment from the following:

cid — Curve identification number (0-999). See note below.

pb — Predefined point anywhere on back tangent.

ptt — Known point at the intersection point of the tangents, tt.

dc — Degree of curvature (defined as the angle, in decimal degrees, subtended by a 100-foot arc).

arcl — Arc length of first spiral (from ts to sc).

arc2 — Arc length of second spiral (from cs to st).

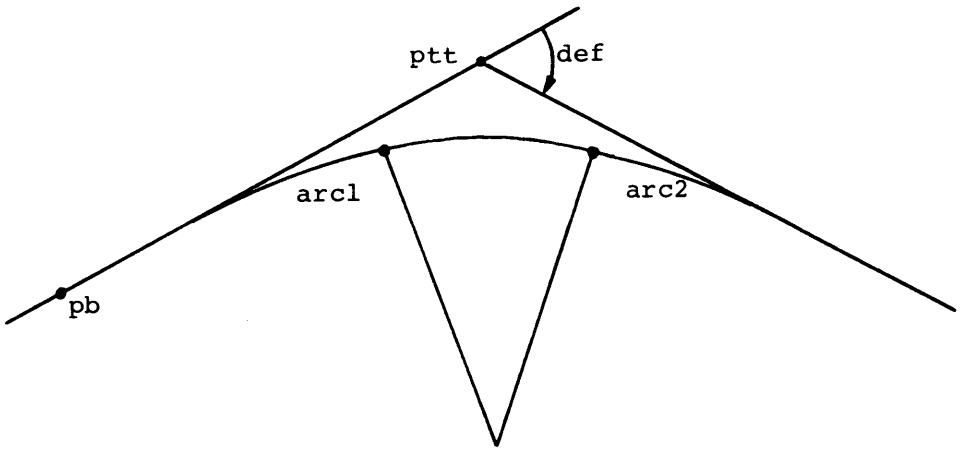
def — Total deflection angle for tangents.

sign — Clockwise or counterclockwise indicator.
1. for spiral clockwise.
-1. for spiral counterclockwise.

NOTE

Points from (cid) to (cid+8) are destroyed in the calculation of results for this command. Care should be taken not to destroy any known points in this range.

Output: Same as the output of SIMPLE SPIRAL command for the first spiral, SIMPLE CURVE command for the circular curve, and SIMPLE SPIRAL command for the second spiral as well as main tangent lengths (ts to tt and tt to st), deflection angle, curve central angle, the intersection point (cid+8) of the circular curve and the line joining tt to the circle center; the offset distance from tt to this point is also given.



Example:

```
STORE 39 800 200  * PB
     39
40 1000 400  * PI1
     40
FIT ALIGNMENT 500 39 40 60. 50. 50. 67. 1.
  SPIRAL 500 SIGN= 1. L=        50.0000 DC= 60- 0- 0.0 DEFLN ANG= 15- 0- 0.0

  LONG TAN L=         33.4538 AZ=  45-  0-  0.0

  SHORT TAN L =        16.7763 AZ=  60-  0-  0.0

    501        937.1604        337.1604

    502        960.8158        360.8158

    503        969.2039        375.3444

  SIMPLE CURVE  500

    503        969.2039        375.3444

    504        985.1796        403.0152

    505        981.2857        434.7286

                    31.9515 = TAN LENGTH        61.6667 = CURVE LENGTH

                                    BACK AZIMUTH =  60- 0- 0.0
                                FORWARD AZIMUTH =  97- 0- 0.0
```

```
SPIRAL 500 SIGN= 1. L=        50.0000 DC= 60- 0- 0.0 DEFLN ANG= 15- 0- 0.0

LONG TAN L=         33.4538 AZ= 112-  0-  0.0

SHORT TAN L =        16.7763 AZ=  97-  0-  0.0

  507        966.7092        482.3976

  506        979.2412        451.3798

  505        981.2857        434.7286

FROM PT. 501 TO PT.  40 DIST=        88.8686

FROM PT.  40 TO PT. 507 DIST=        88.8686

DEFLECTION ANGLE =  67-  0-  0.0
CURVE CENTRAL ANGLE =  37-  0-  0.0

RADIAL FROM  40 INTERSECTS ALIGNMENT AT 508
OFFSET=        20.3276
  508        980.0805        404.0527

500 AY 40 60. 50. 50. 0Z. -1.

SPIRAL 600 SIGN=-1. L=        50.0000 DC= 60- 0- 0.0 DEFLN ANG= 15- 0- 0.0

LONG TAN L=         33.4538 AZ=  45-  0-  0.0

SHORT TAN L =        16.7763 AZ=  30-  0-  0.0

  601        937.1604        337.1604

  602        960.8158        360.8158

  603        975.3444        369.2039

SIMPLE CURVE  600

  603        975.3444        369.2039

  604       1003.0152        385.1796

  605       1034.7286        381.2857

                  31.9515 = TAN LENGTH        61.6667 = CURVE LENGTH

                                    BACK AZIMUTH =  30- 0- 0.0
                                 FORWARD AZIMUTH = 353- 0- 0.0
```

```
SPIRAL 600 SIGN=-1. L=        50.0000 DC= 60- 0- 0.0 DEFLN ANG= 15- 0- 0.0

LONG TAN L=          33.4538 AZ= 338-  0-  0.0

SHORT TAN L =         16.7763 AZ= 353-  0-  0.0

  607       1082.3976        366.7092

  606       1051.3798        379.2412

  605       1034.7286        381.2857

FROM PT. 601 TO PT.  40 DIST=         88.8686

FROM PT.  40 TO PT. 607 DIST=         88.8686

DEFLECTION ANGLE =  67-  0-  0.0
CURVE CENTRAL ANGLE =  37-  0-  0.0

RADIAL FROM  40 INTERSECTS ALIGNMENT AT 608
OFFSET=        20.3276
  608       1004.0527        380.0805
```

3.9 TABULAR OUTPUT

These commands neither store nor modify any data contained in a COGO table. Their sole purpose is to interpret and output data already stored in the COGO table in a meaningful manner to users.

The output of a COGO job can be produced on a terminal or printer. The standard output format has answers interspersed with the input list. The printing of input or output can be suppressed.

- - - - - - - - - -

LIST COORDINATES desc

Output: The coordinates of the points specified in desc are listed. Undefined points are not listed.

Examples:

```
LIST COORDS 1-50          *Points 1 through 50
      (1)                 *Point 1
      1                   *Points of figure 1
      1 50                *Points 1 and 50
```

```
STORE 949 10414.1049 9090.8067
    949
960 10413.8278 9200.9691
    960
959 10473.8277 9201.0200
    959
950 10474.1049 9090.8309
    950
STORE FIGURE
       1 ( 949 960 959 950 949)
            1
LIST COORDINATES
         (949 960 959 950)
    949     10414.1049      9090.8067

    960     10413.8278      9200.9691

    959     10473.8277      9201.0200

    950     10474.1049      9090.8309

1
    949     10414.1049      9090.8067

    960     10413.8278      9200.9691

    959     10473.8277      9201.0200

    950     10474.1049      9090.8309

    949     10414.1049      9090.8067

960 950
    960     10413.8278      9200.9691

    950     10474.1049      9090.8309
```

- - - - - - - - - -

LIST FIGURES desc

Output: The figures specified in desc are listed. Undefined figures are not listed.

List the figures specified by desc, where desc can be a figure number or a list of figure and point numbers. Parentheses must enclose a list. If desc is a figure number and the list associated with that figure number includes point numbers, the LIST FIGURES command does not list the point numbers.

Examples:

```
LIST FIGS (1 3)          *List figures 1 and 3.
      (10 10)            *List figure 10 twice.
      (1-20 600-650)     *List figures 1 through 20 and
                         *600 through 650.

LIST FIGURES 77          *List the figures whose numbers
                         *appear in figure 77.

LIST FIGURE (77)         *List figure 77.
```

```
DELETE FIGURES 2-3
STORE FIGURE 2(950 959 958 951 950)
               2
3 (951 958 957 952 951)
               3
4(952 957 956 953 952)
               4
5 (953 956 630 C638R 980 969 C910L 1003 953)
               5
9876 (1 3-5 10)
           9876
10 (101 105 836 4 78 C12R 52-54 781 210 101)
             10
LIST FIGURES
         (1-5 9876)
               1 (949 960 959 950 949)

               2 (950 959 958 951 950)

               3 (951 958 957 952 951)

               4 (952 957 956 953 952)

               5 (953 956 630 C638R 980 969 C910L 1003 953)

            9876 (1 3-5 10)

1
2
3
4
5
9876
               1 (949 960 959 950 949)

               3 (951 958 957 952 951)

               4 (952 957 956 953 952)

               5 (953 956 630 C638R 980 969 C910L 1003 953)

              10 (101 105 836 4 78 C12R 52-54 781 210 101)
```

```
10
              4 (952 957 956 953 952)

9876
              1 (949 960 959 950 949)

              3 (951 958 957 952 951)

              4 (952 957 956 953 952)

              5 (953 956 630 C638R 980 969 C910L 1003 953)

             10 (101 105 836 4 78 C12R 52-54 781 210 101)
```

- - - - - - - - - - -

LIST POINT NUMBERS desc

The point numbers included in the list specified by desc that have been defined are listed in the order defined by desc.

Examples

LIST POINT NUMBERS 1-999	* All valid point numbers * between 1 and 999 are listed.
LST PT NUM 1	* All valid point numbers * described by figure 1 are * listed.
L P N (1-50 100-200 450-600)	* All valid point numbers * 1 through 50, 100 through * 200, and 450 through * 600 are listed.

- - - - - - - - - - -

DISTANCE desc

Compute the distances between the points of the description desc.

Output: Distance from first point to second point, second point to third point, etc.

Example:

```
DISTANCE
     (949 960 959 950 949)
                    FROM  949  TO  960      110.1627 FT.
                    FROM  960  TO  959       59.9999 FT.
                    FROM  959  TO  950      110.1894 FT.
                    FROM  950  TO  949       60.0000 FT.

                    FROM  949  TO  960      110.1627 FT.
                    FROM  960  TO  959       59.9999 FT.
                    FROM  959  TO  950      110.1894 FT.
                    FROM  950  TO  949       60.0000 FT.
```

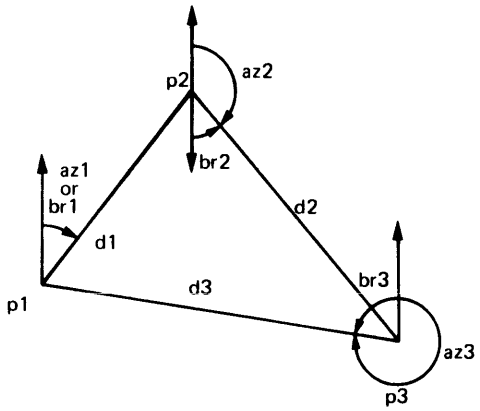
INVERSE AZIMUTHS desc

INVERSE BEARINGS desc

Compute the azimuth or bearing, respectively, of the line segments from the first point in the figure described by desc to the second, from the second point to the third, and so forth.

These commands are generalizations of the LOCATE AZIMUTH and LOCATE BEARING commands described in Section 3.5.

Output: Azimuth or bearing, respectively, and the length of line segments from first point to the second, from second point to the third, and so forth.



desc = (p1 p2 p3 p1)

Example:

```
INVERSE AZIMUTHS
        1
                FROM  949  TO  960      90- 8-38.8        110.1627
                FROM  960  TO  959       0- 2-55.0         59.9999
                FROM  959  TO  950     270- 8-38.9        110.1894
                FROM  950  TO  949     180- 1-23.2         60.0000
INVERSE BEARINGS
         (949 960 959 950 949)
                FROM  949  TO  960   S 89-51-21.2 E        110.1627
                FROM  960  TO  959   N  0- 2-55.0 E         59.9999
                FROM  959  TO  950   N 89-51-21.1 W        110.1894
                FROM  950  TO  949   S  0- 1-23.2 W         60.0000
```

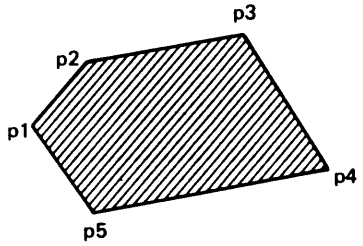
- - - - - - - - - -

AREA desc

AREA AZIMUTHS desc

AREA BEARINGS desc

Compute the area of the closed polygon defined by the list of points in the description desc. To define a closed polygon, the last point number in the description desc must be the same as the first. Starting at one point, the corners of the polygon should be entered consecutively along its perimeter in one direction until the starting point is reached.



desc = (p1 p2 p3 p4 p5 p1)

Output: Area of the figure in square feet and acres.

The AREA AZIMUTHS and AREA BEARINGS commands also produce a table of coordinates of the corners, the azimuth or bearing, respectively, and length of each side.

Example:

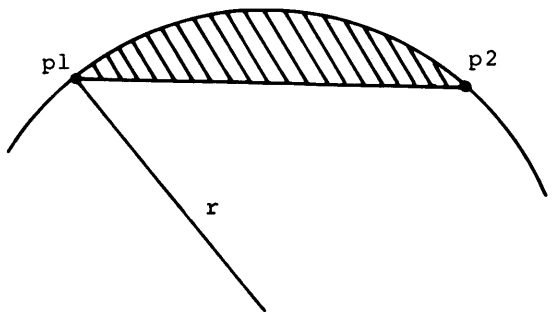
```
AREA 1

            AREA=     6610.550 SQ.FT.=   0.15175735 ACRES

AREA AZIMUTHS
          1
    949     10414.1049      9090.8067
                                          90- 8-38.8        110.1627
    960     10413.8278      9200.9691
                                           0- 2-55.0         59.9999
    959     10473.8277      9201.0200
                                         270- 8-38.9        110.1894
    950     10474.1049      9090.8309
                                         180- 1-23.2         60.0000
    949     10414.1049      9090.8067

            AREA=     6610.550 SQ.FT.=   0.15175735 ACRES

AREA BEARINGS
          (949 960 959 950 949)
    949     10414.1049      9090.8067
                                       S 89-51-21.2 E       110.1627
    960     10413.8278      9200.9691
                                       N  0- 2-55.0 E        59.9999
    959     10473.8277      9201.0200
                                       N 89-51-21.1 W       110.1894
    950     10474.1049      9090.8309
                                       S  0- 1-23.2 W        60.0000
    949     10414.1049      9090.8067

            AREA=     6610.550 SQ.FT.=   0.15175735 ACRES
```

- - - - - - - - - - -

SEGMENT p1 p2 r

Compute the area of a circular segment whose boundaries are the arc between p1 and p2 with radius r and the chord between p1 and p2.

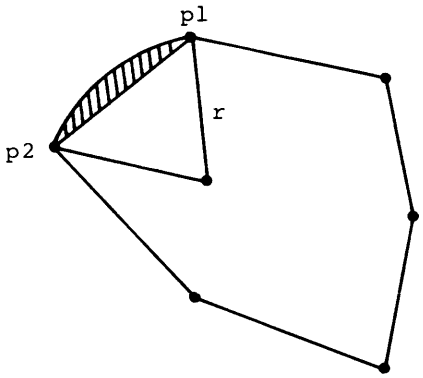
Output: The lengths of the chord and arc, and the area of the segment in square feet and acres.



- - - - - - - - - - -

SEGMENT PLUS p1 p2 r

SEGMENT MINUS p1 p2 r



Find the area of the segment as in the SEGMENT command, then add (subtract) that area to (from) the cumulative area resulting from all SEGMENT PLUS (SEGMENT MINUS) commands following the most recent AREA-type command (and including the result of the AREA command itself). These commands allow segments to be added to or subtracted from polygons for parcels bounded by curves.

Output: Chord length, arc length, segment area, and cumulative area.

Example:

```
STORE 252 3426.167 6728.368
    252
257 3461.543 6743.700
    257
SEGMENT 252 257 80.
     SEGMENT AREA=        60.776 SQ.FT.=    0.00139522 ACRES R=    80.0000

                     CHORD=    38.5556 DELTA= 27-53-16.3    L=    38.9389

SEGMENT PLUS 252 257 80
     SEGMENT AREA=        60.776 SQ.FT.=    0.00139522 ACRES R=    80.0000

                     CHORD=    38.5556 DELTA= 27-53-16.3    L=    38.9389

             AREA=      6671.326 SQ.FT.=    0.15315257 ACRES

SEGMENT 252 257 80.
     SEGMENT AREA=        60.776 SQ.FT.=    0.00139522 ACRES R=    80.0000

                     CHORD=    38.5556 DELTA= 27-53-16.3    L=    38.9389

SEGMENT MINUS 252 257 80
     SEGMENT AREA=        60.776 SQ.FT.=    0.00139522 ACRES R=    80.0000

                     CHORD=    38.5556 DELTA= 27-53-16.3    L=    38.9389

             AREA=      6610.550 SQ.FT.=    0.15175735 ACRES
```

- - - - - - - - - - -

STAKING NOTES inst ibs rad [desc]

The COGO file is searched for all point numbers that are within a radius of rad to the instrument point number inst. Output consists of backsite bearing and distance from inst to ibs and the bearing, distance and right azimuth from backsite point ibs to all defined points within the defined radius. An optional figure desc may be specified in which case only the point numbers within that description are printed.

Example:

```
STORE
    1 345.38 223.47
      1
2 400 400
      2
3 700 100
      3
4 300 500
      4
5 300 100
      5
DIVIDE ARC
        3 4 5 20 10
```

10	698.7669	131.3836
11	695.0753	162.5738
12	688.9480	193.3781
13	680.4226	223.6068
14	669.5518	253.0734
15	656.4026	281.5962
16	641.0561	308.9994
17	623.6068	335.1141
18	604.1624	359.7792
19	582.8427	382.8427
20	559.7792	404.1624
21	535.1141	423.6068
22	508.9994	441.0561
23	481.5962	456.4026
24	453.0734	469.5518
25	423.6068	480.4226
26	393.3781	488.9480
27	362.5738	495.0753
28	331.3836	498.7669

```
STAKING NOTES
        2 1 200
    TRAN TO BS
          FS     BEAR.          DIST.     AZI.RT.
   2       1  S72-48-26.8W   184.79
           4  S45- 0- 0.0E   141.42   242-11-33.2
          19  N 5-21-38.6W   183.65   101-49-54.5
          20  N 1-29-32.2E   159.83   108-41- 5.3
          21  N 9-54-37.9E   137.16   117- 6-11.1
          22  N20-38-22.4E   116.48   127-49-55.5
          23  N34-39-13.7E    99.19   141-50-46.9
          24  N52-39-12.9E    87.49   159-50-46.1
          25  N73-38-28.4E    83.82   180-50- 1.6
          26  S85-44-32.6E    89.19   201-27- 0.6
          27  S68-30-46.9E   102.18   218-40-46.3
          28  S55-12-40.1E   120.26   231-58-53.1
```

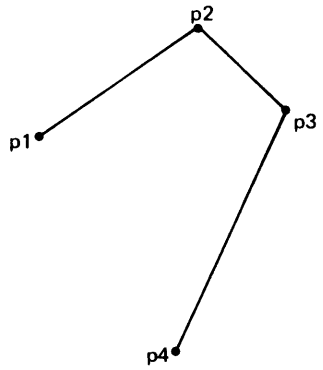
- - - - - - - - - - -

TRAVERSE AZIMUTHS desc

TRAVERSE BEARINGS desc

TRAVERSE ANGLES desc

TRAVERSE DEFLECTIONS desc



desc = (p1 p2 p3 p4)

Compute the azimuths, bearings, angles, or deflections, respectively, and the lengths of each line segment defined by consecutive points in the descriptions desc.

These commands are similar to the AREA AZIMUTHS and AREA BEARINGS commands. They are different from these two AREA commands in that they do not compute an area and do not require the first and last points in the description to be the same.

Output: Coordinates of each point in the description desc.

The length and azimuth, bearing, angle, or deflection of each line segment defined by consecutive points in desc.

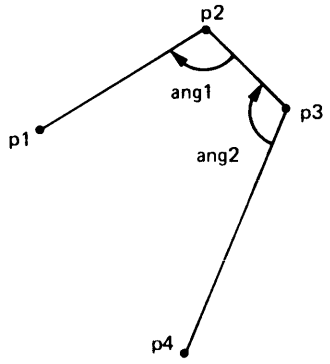
Examples:

```
TRAVERSE AZIMUTHS
         1
    949      10414.1049       9090.8067
                                                90- 8-38.8          110.1627
    960      10413.8278       9200.9691
                                                 0- 2-55.0           59.9999
    959      10473.8277       9201.0200
                                               270- 8-38.9          110.1894
    950      10474.1049       9090.8309
                                               180- 1-23.2           60.0000
    949      10414.1049       9090.8067

TRAVERSE BEARINGS
          1
    949      10414.1049       9090.8067
                                            S 89-51-21.2 E          110.1627
    960      10413.8278       9200.9691
                                            N  0- 2-55.0 E           59.9999
    959      10473.8277       9201.0200
                                            N 89-51-21.1 W          110.1894
    950      10474.1049       9090.8309
                                            S  0- 1-23.2 W           60.0000
    949      10414.1049       9090.8067

TRAVERSE ANGLES
         949 960 959 950 949
    949      10414.1049       9090.8067
                                                                    110.1627
    960      10413.8278       9200.9691          89-54-16.1
                                                                     59.9999
    959      10473.8277       9201.0200          90- 5-43.9
                                                                    110.1894
    950      10474.1049       9090.8309          89-52-44.3
                                                                     60.0000
    949      10414.1049       9090.8067

TRAVERSE DEFLECTIONS
         ( 949 960 959 950 949)
    949      10414.1049       9090.8067
                                                                    110.1627
    960      10413.8278       9200.9691          90- 5-43.9 L
                                                                     59.9999
    959      10473.8277       9201.0200          89-54-16.1 L
                                                                    110.1894
    950      10474.1049       9090.8309          90- 7-15.7 L
                                                                     60.0000
    949      10414.1049       9090.8067
```

- - - - - - - - - - -

ANGLES desc

Compute the clockwise angles formed by the line segments connecting consecutive points in the description desc and the lengths of each of the line segments.



desc = (p1 p2 p3 p4)

Output: Lengths of line segments from the first point to the second point and from the second point to the third point and the angle between those two line segments, lengths of line segments from the second point to the third point and from third point to the fourth point and the angle between those two line segments, and so forth.

Example:

```
ANGLES 1
  ANGLE  949  960  959        110.1628          89-54-16.1          60.0000

  ANGLE  960  959  950         60.0000          90- 5-43.9         110.1895

  ANGLE  959  950  949        110.1895          89-52-44.3          60.0001

949 960 959
  ANGLE  949  960  959        110.1628          89-54-16.1          60.0000
```

CHAPTER 4

SAMPLE JOBS

The purpose of this chapter is to better acquaint users with COGO-10/20.

4.1 SAMPLE JOB 1: LAYING OUT A ROADWAY

The problem, as shown in Figure 4-1, is to lay out a roadway through the center of a given lot and to determine the area of the remaining usable part of the lot.

To prevent confusion between defined and undefined point numbers, all points in the coordinate table are set initially to (0,0).

```
DELETE COORDINATES
          1 - 999
```

A starting point is defined by the STORE command.

```
STORE 1 500. 500.
      1
```

To find the coordinates of point 2, the length and azimuth of the line connecting points 1 and 2 are specified. This line is referred to as line 1-2.

```
LOCATE AZIMUTH 1 2 63. 585.
      2       765.5844      1021.2388
```

Now proceed clockwise around the perimeter of the lot, defining each corner by its distance and deflection angles from the previous border segment. Notice that the longest side of the lot has an azimuth of 63°0'0", and that all angles around the border are right angles. Thus, the deflection angle in each command is either 90° for an outside corner or -90° for an inside corner. Because the same command is used consecutively six times, the command name need appear only once.

```
LOCATE DEFLECTION
 1 2 3    -90. 400.
       3       1121.9871        839.6426

2 3 4 90. 1347.
       4       1733.5123       2039.8284

3 4 5 90. 400.
       5       1377.1096       2221.4246

4 5 6 90. 547.
       6       1128.7768       1734.0440

5 6 7 -90. 400.
       7        772.3742       1915.6402


       8        143.5974        681.5962

INVERSE AZIMUTHS
                  1 8
                     FROM    1  TO    8       153- 0- 0.0         400.0000
```

A series of point numbers to define a perimeter can be grouped as a figure with the STORE FIGURE command. This list of points can then be entered into a command by using the figure number. The points in figure 1 are used to calculate the area in the AREA 1 command.

```
STORE FIGURE 1 ( 1 - 8 1 )
                 1
AREA  1

             AREA=  1092800.002 SQ.FT.=  25.08723603 ACRES
```

To find the point where the center of the roadway crosses line 1-8 (point 10), the line is divided into two equal segments.

```
DIVIDE LINE 8 1 2 10
     10        321.7987        590.7981
```

To find the point where the right-hand side of the roadway intersects line 4-5 at the opposite side of the lot, a procedure is followed similar to that used on line 1-8. The command finds the midpoint (point 13) of line 4-5.

```
DIVIDE LINE 4 5 2 13
     13       1555.3110       2130.6265
```

Next, set up a traverse across the lot approximating the center line of the roadway. The traverse starts on a course parallel to the longest side of the lot, makes a 45° turn to the left, a 45° turn to the right, and ends up once again parallel to the longest side of the lot. The 45° portion of the roadway passes through a point (point 15) midway between points 2 and 6.

```
DIVIDE LINE 2 6 2 15
     15        947.1806       1377.6414
```

By using the azimuth intersect command twice, the points 14 and 16 can be located at the intersections of the first and third courses with the 45° leg. Since the azimuth of the first and third courses is parallel to the lot lines, the azimuth of the 45° leg through point 15 is 63°-45°=18°.

```
AZIMUTH INTERSECT
              14 10 63 0 0 15 18 0 0
     14        678.1812       1290.2382

              16 13 63 0 0 15 18 0 0
     16       1216.1800       1465.0446
```

A 50 ft. transition tangent from points 17 to 18 is now located by going 25 ft. on either side of point 15.

```
LOCATE LINE 15 14 17 25
     17        923.4042       1369.9160

           15 16 18 25
     18        970.9571       1385.3669
```

The distance from point 14 to point 17 along the tangent to curve 1 is used to locate the beginning of curve 1 (point 19) on the line from point 14 to point 10. In the following command, the previously defined distance from point 14 to point 17 is substituted by COGO. Curve 2 from point 18 to point 20 is handled similarly.

```
LOCATE LINE 14 10 19 D 14 17
          19        561.1231       1060.4987

                16 13 20 D 16 18
          20       1333.2382       1694.7842
```

Knowing that the tangents to curve 1 are lines from point 19 to point 14 and from point 14 to point 17 and knowing that points 19 and 17 are points of tangency, the radius point (point 50) for a circular curve through points 19 and 17 can be found using the FIT CURVE command. The radius point (point 51) for curve 2 can be found in a similar manner using points 18, 16, and 20.

```
FIT CURVE 19 14 17 19 50 17
     19        561.1231       1060.4987

     50       1115.7634        777.8953

     17        923.4042       1369.9160

          18 16 20 18 51 20
     18        970.9571       1385.3669

     51        778.5979       1977.3875

     20       1333.2382       1694.7842
```

The center line is now fully described and is stored as figure 2.

```
STORE FIGURE 2 (10 19 C 50 L 17 18 C 51 R 20 13 )
               2
```

The PARALLEL FIGURE command now locates the points along either side of the center line stored in figure 2. The road is to be 40 ft. wide. The first PARALLEL FIGURE command locates all the points on the left side (-20) of the right-of-way and the second command locates the points on the right side (20) of the right-of-way.

```
PARALLEL FIGURE 2 -20 80
     80        339.6188        581.7183

     81        578.9432       1051.4189

     82        929.5846       1350.8949

     83        977.1374       1366.3457

     84       1351.0583       1685.7044

     85       1573.1311       2121.5467

PARALLEL FIGURE 2  20 90
     90        303.9786        599.8779

     91        543.3030       1069.5785

     92        917.2239       1388.9371

     93        964.7767       1404.3880

     94       1315.4181       1703.8640

     95       1537.4908       2139.7063
```

The left and right side lines are stored as figures 3 and 4, respectively. Note that the radius points associated with figures 3 and 4 are the same as the radius points located by the FIT CURVE command for the center line.

```
STORE FIGURE 3 ( 80 81 C 50 L 82 83 C 51 R 84 85 )
               3
4 (90 91 C 50 L 92 93 C 51 R 94 95 )
               4
```

The land to either side of the right-of-way can be defined by the AREA BEARINGS command. A complete description, including radius, deltas, and curve length, is listed along with the bearings and distance for each course. The AREA AZIMUTH command would give similar information with azimuths replacing bearings.

```
AREA BEARINGS ( 90 91 C 50 L 92 93 C 51 R 94 95 5 - 8 90 )
     90        303.9786        599.8779
                                          N 63- 0- 0.0 E       527.1573
     91        543.3030       1069.5785
                                          N 27- 0- 0.0 W       642.4874
CC   50       1115.7634        777.8953 DELTA= 45- 0- 0.0 L  L=  504.6085
                                          S 72- 0- 0.0 E       642.4874
     92        917.2239       1388.9371
                                          N 18- 0- 0.0 E        50.0000
     93        964.7767       1404.3880
                                          S 72- 0- 0.0 E       602.4874
CC   51        778.5979       1977.3875 DELTA= 45- 0- 0.0 R  L=  473.1925
                                          N 27- 0- 0.0 W       602.4874
     94       1315.4181       1703.8640
                                          N 63- 0- 0.0 E       489.1573
     95       1537.4908       2139.7063
                                          S 27- 0- 0.0 E       180.0000
      5       1377.1096       2221.4246
                                          S 63- 0- 0.0 W       547.0000
      6       1128.7768       1734.0440
                                          S 27- 0- 0.0 E       400.0000
      7        772.3742       1915.6402
                                          S 63- 0- 0.0 W      1385.0000
      8        143.5974        681.5962
                                          N 27- 0- 0.0 W       180.0000
     90        303.9786        599.8779
               AREA=    505517.692 SQ.FT.=  11.60508935 ACRES

               ( 80 81 C 50 L 82 83 C 51 R 84 85 4 - 1 80 )

     80        339.6188        581.7183
                                          N 63- 0- 0.0 E       527.1573
     81        578.9432       1051.4189
                                          N 27- 0- 0.0 W       602.4874
CC   50       1115.7634        777.8953 DELTA= 45- 0- 0.0 L  L=  473.1925
                                          S 72- 0- 0.0 E       602.4874
     82        929.5846       1350.8949
                                          N 18- 0- 0.0 E        50.0000
     83        977.1374       1366.3457
                                          S 72- 0- 0.0 E       642.4874
CC   51        778.5979       1977.3875 DELTA= 45- 0- 0.0 R  L=  504.6085
                                          N 27- 0- 0.0 W       642.4874
     84       1351.0583       1685.7044
                                          N 63- 0- 0.0 E       489.1573
     85       1573.1311       2121.5467
                                          N 27- 0- 0.0 W       180.0000
      4       1733.5123       2039.8284
                                          S 63- 0- 0.0 W      1347.0000
      3       1121.9871        839.6426
                                          S 27- 0- 0.0 E       400.0000
      2        765.5844       1021.2388
                                          S 63- 0- 0.0 W       585.0000
      1        500.0000        500.0000
                                          S 27- 0- 0.0 E       180.0000
     80        339.6188        581.7183
               AREA=    505517.692 SQ.FT.=  11.60508934 ACRES

END OF RUN
```

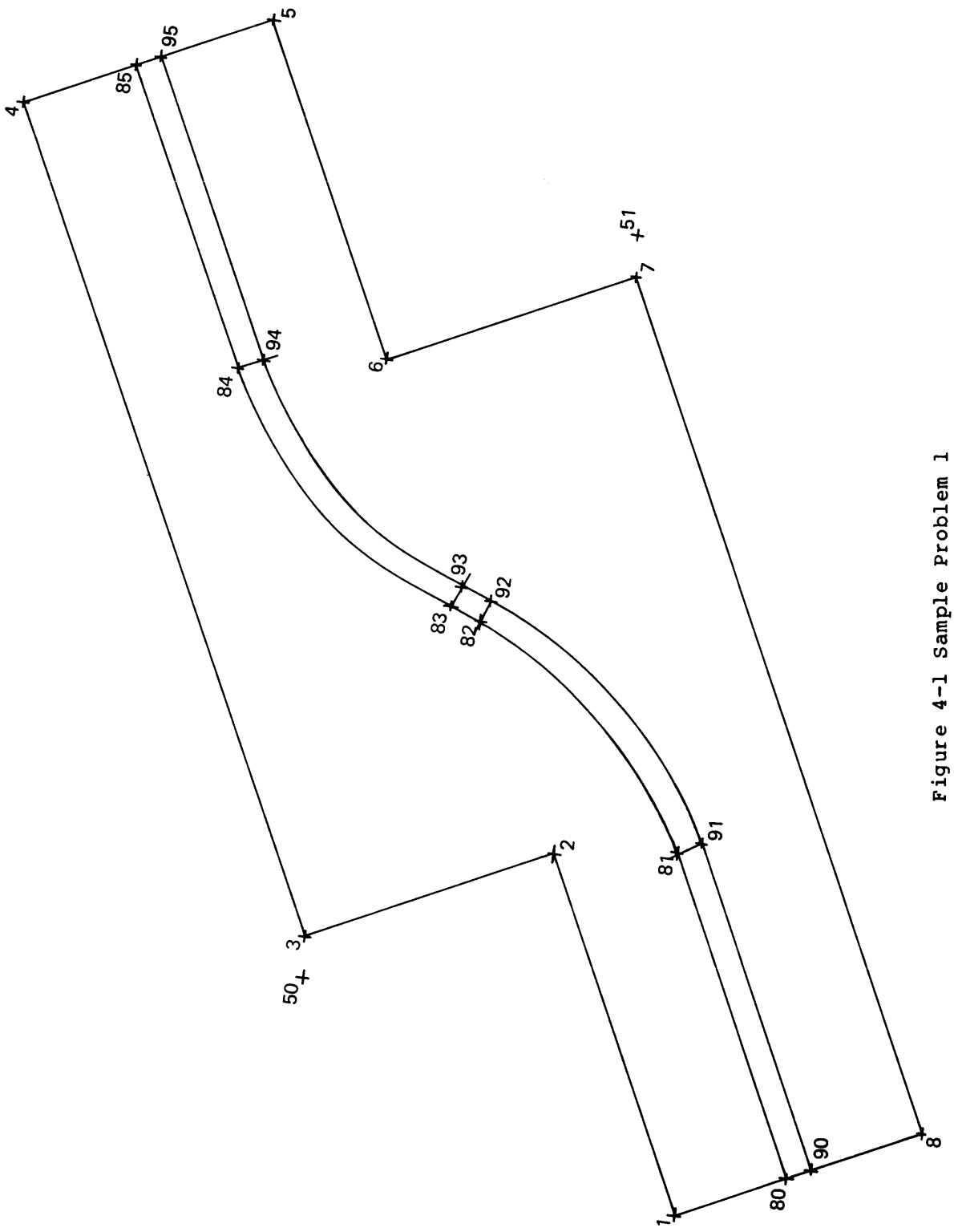


Figure 4-1 Sample Problem 1

4.2 SAMPLE JOB 2: PARCEL TAKING FOR STREET RIGHT-OF-WAY

The problem, as shown in Figure 4-2, involves conversion of a dead end street into a through street with accompanying straightening and widening of the right-of-way to 200 feet. This sample shows computation of the parcel taking from the L. B. Jones property located at the cul-de-sac. This sample computes the property closure from the deed, converts the property into the coordinate system of the road, computes the taking line, and prints the taking description.

The coordinates of two points on the new center line are stored as points 11 and 12.

```
STORE  11       4980.4290       8190.8836 *  POINT ON C/L
     11
STORE  12       5402.8286       8261.3101 *  POINT ON C/L
     12
```

Since no command name is specified on these lines, the command names are assumed to be the same as the previous command (STORE). Coordinates of points 54 and 52, two lot corners located in the field, are stored.

```
   54 5180.557 8293.530 * LOT CORNER
54
    52 5115.365 8526.918* LOT CORNER
52
```

An assumed starting point for running out the property description is stored as point 1.

```
STORE
   1 10000 10000 * ARBITRARY STARTING POINT
     1
```

The LOCATE BEARING command from point 1 locates point 2 at bearing S 29-18-30 E and distance 282.33 feet. The other LOCATE BEARING commands extend the remaining boundary courses and are formatted to show some of the flexibility available.

```
LOCATE BEARING
          1 2 S29-18-30E 282.33     * DEED DESCRIPTION
      2       9753.8088      10138.2032

          2 3 S41 12 30W 135
      3       9652.2457      10049.2653

          3 4 N38 7 02 W 228.25
      4       9831.8213       9908.3729

          4 5 A 3 4 65
      5       9882.9600       9868.2502

          5 6 N67 1E D 4 5
      6       9908.3401       9928.0904

          6 7 A 5 6 50
      7       9927.8633       9974.1213
```

```
    7 8 N29-32-10 W D6 7
8      9971.3655      9949.4727

    8 9 N60 27 50E 58.14
9      10000.0269     10000.0571
```

The delimeter A instructs COGO to substitute the azimuth of line 3-4 for A 3 4 before executing the command.

The INVERSE BEARINGS command checks the error of closure. The figure, (9 1), is entered here without parentheses. This is permitted only when the figure contains more than one number, fits entirely on one card, and is the only item allowed for the particular command.

```
INVERSE BEARINGS
            — 9 1  * ERROR OF CLOSURE
                FROM    9  TO    1      S 64-45-29.5 W          0.0632
```

The numbers of the points that make up the boundary are stored as figure 1. Notice how the curves are specified in the figure. This figure could also have been written as (1 2 3 4 C5L 6 C7R 8 1) or (1-C5L-C7R 8 1).

```
STORE FIGURE   1 ( 1-4 C5L 6 C 7 R 8 1 ) *ORIGINAL PARCEL
               1
```

This command converts the coordinates of the parcel into the highway coordinate system. The first 1 specifies that figure 1 contains the list of points to be converted. The second 1 specifies that the converted coordinates should be stored as the points listed in figure 1. The original values of these points will be lost by this conversion. If it were desired that the original values remain intact, the second 1 could be replaced by a figure (or reference to a stored figure) that specified different point numbers. Then, after the conversion, both the original and the converted coordinates would be available for computation.

```
CONVERT MERIDIAN  1 1   4 54  ( A54 52  -  A 4 2 )
                                SHIFT = 199- 8-46.2          4923.6143
                       CONVERSION ANGLE = 356-51-27.2
                           SCALE FACTOR = 0.100000000E+01
```

The third and fourth data items, 4 54, specify the shift of the conversion. Four is the number of a point in the system to be converted and 54 is the same point in the desired system.

The remaining data item (A 54 52 - A 4 2) specifies the conversion angle as the difference between two azimuths. A 4 2 is the azimuth between two points according to the deed. A 54 52 is the azimuth between the same points as located in the field. The difference between these is the desired conversion angle.

The distance from 2 to 52 is the difference between the property corner as located by the converted deed and in the field. Note that here the figure, (2 52), is entered without parentheses as in line 15.

```
DISTANCE 2 52 * MISFIT AT REAR CORNER
                FROM    2  TO   52          0.3875 FT.
```

The area of the parcel is computed and printed by this command. Note the reference to stored figure 1.

```
AREA              1   *   L. B. JONES PROPERTY  (ORIGINAL)

                AREA=     41031.881 SQ.FT.=    0.94196236 ACRES
```

This line contains only a comment. The double asterisk (**) causes a new page to be started in any output sent to the line printer.

```
**                          LOCATION OF TAKING LINE
```

This POINTS INTERSECT command defines point 22 as the intersection of the line through points 11 and 12 and a line through points 1 and 8 after first offsetting the line through 11 and 12 to the right 100 feet.

```
POINTS INTERSECT
           22  11 12  1 8  100.
     22       5338.1795       8351.9116
```

Point 23, the other intersection of the right-of-way with the property, is computed by this POINTS INTERSECT.

```
POINTS INTERSECT 23 11 12 3 4 100.
     23       5149.8088       8320.5046
```

This AREA BEARINGS command will print the description and area of the taking. Note that the output (in Figure 3) includes coordinates, bearings, distances, and curve data. The figure specified in the input to this command, (4 C5L - C7R 8 22 23 4), could have been written without parentheses as explained in the paragraph describing the INVERSE BEARINGS command in this section.

```
AREA BEARINGS     ( 4 C5L-C7R 8 22 23 4) * TAKING TK-26
      4       5180.5570       8293.5300
                                          N 41-15-34.8 W        65.0000
CC    5       5229.4194       8250.6643 DELTA= 74-51-58.0 L  L=   84.9330
                                          N 63-52-27.2 E        65.0000
      6       5258.0417       8309.0232
                                          N 63-52-27.2 E        50.0000
CC    7       5280.0588       8353.9147 DELTA= 83-26-50.0 R  L=   72.8215
                                          N 32-40-42.8 W        50.0000
      8       5322.1445       8326.9184
                                          N 57-19- 0.4 E        29.6948
     22       5338.1795       8351.9116
                                          S  9-27-57.0 W       190.9710
     23       5149.8088       8320.5046
                                          N 41-15-34.8 W        40.9033
      4       5180.5570       8293.5300

                AREA=      4579.478 SQ.FT.=    0.10513034 ACRES

END OF JOB
```

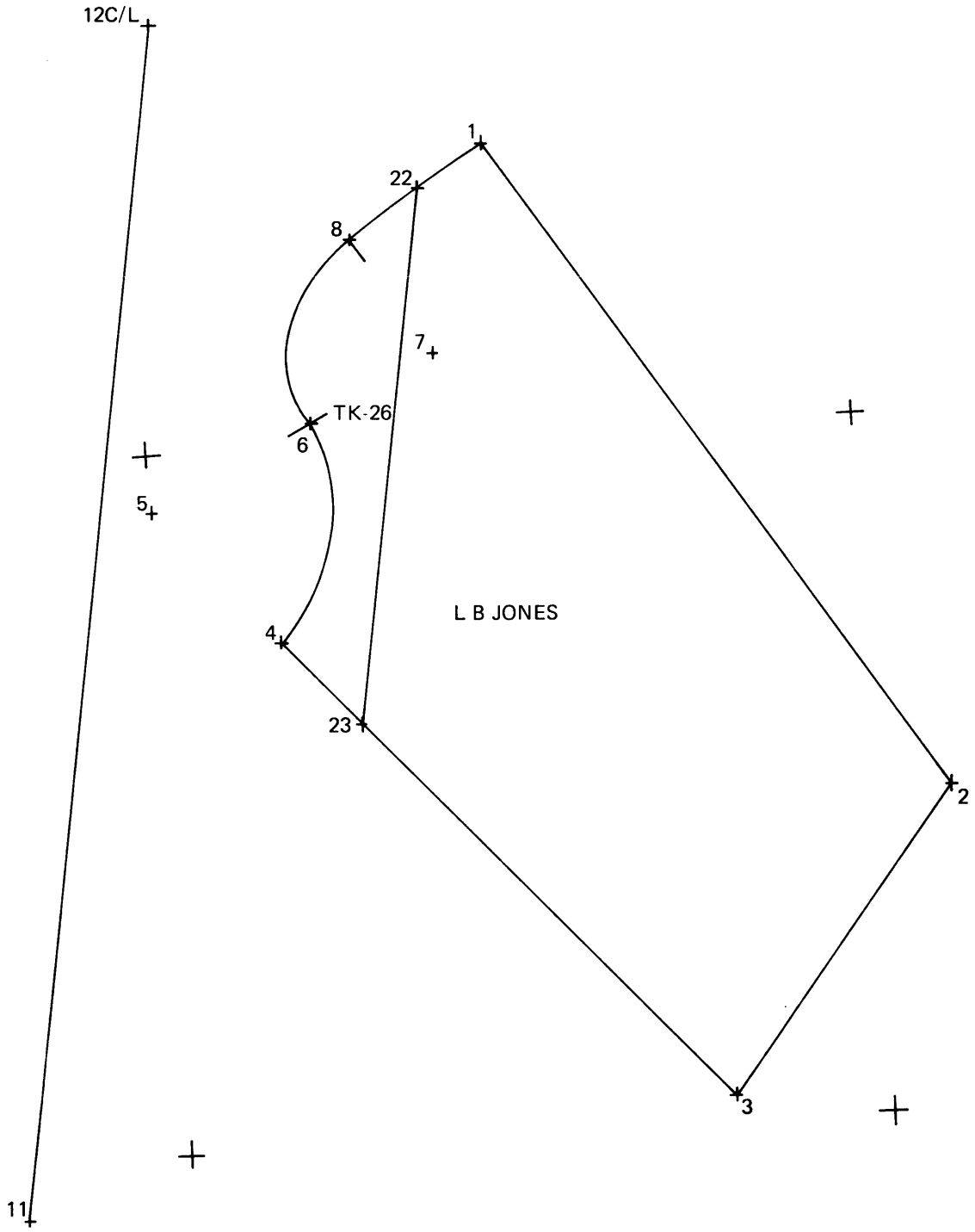


Figure 4-2 Sample Problem 2

4.3 SAMPLE JOB 3: SUBDIVISION COMPUTATION

```
       * EXAMPLE OF SUBDIVISION COMPUTATION AND PLOTTING
STORE 1 10000. 10000.
      1
LOCATE BEARING
        1 10  S 11 33.0 W  67.0
     10       9934.3567       9986.5851

LOCATE ANGLE
        1 10 16  -90.  285.5
     16       9991.5204       9706.8663

      10 16 17 42. 285.5                  * POINT 17 PT AND PC
     17       9761.8712       9876.4879

      10 16 18 42.  (285.5 + 165)         * POINT 18 CC
     18       9629.1492       9974.5179

      16 18 21 -53 31.0  165              * POINT 21 PT
     21       9629.2452       9809.5179

STORE FIGURE
      1 ( 1 10 C16R 17 C18L 21)           * FIG 1 IR R/W OF ROAD
             1
LOCATE BEARING
        1 2 S 78 27 E 122.63              * POINT 2
      2       9975.4467      10120.1468

        1 3 S 78 27 E (122.63 + 109.4)    * POINT 3
      3       9953.5423      10227.3315

        2 11 S 11 35. W  158              * POINT 11 ON SAC AT PC
     11       9820.6645      10088.4215

FIGURE LINE INTERSECT
          24 1 11 N 62 27. W 10           * LOC POINT 24 ON CURVE
       INTERSECTION IS AT SEGMENT   10   17
     24       9882.1387       9970.5819

*  DEFINE CUL-DESAC AS FIGURE 2
LOCATE ANGLE
        24 11 27 90.  50.                 * CURVE CC    POINT 27
     27       9864.9949      10111.5476

        11 27 25  -48 11.5 50             * PT OF CURVE  POINT 25
     25       9818.2047      10129.1737

        11 27 26  -48 11.5 (50 + 40)      * CC OF SAC   POINT 26
     26       9780.7726      10143.2746

        27 26 14  (180. + 48 11.5 ) 40    * PT OF SAC  POINT 14
     14       9745.3083      10124.7737

FIGURE LINE INTERSECT
          23 1 14 N 62 27. W  10          * LOC POINT 23 ON CURVE
       INTERSECTION IS AT SEGMENT   10   17
     23       9837.8106       9947.4564

STORE FIGURE
      2 (24 11 C27L 25 C26R 14 23)        * STORE FIGURE 2

      2
```

```
*  COMPUTE PERIMETER OF LOTS
LOCATE BEARING
        3 4 S 72 27 45 E 90.36          * POINT 4
      4       9926.3141      10313.4915

        4 5 S 9 11. E  252.6            * POINT 5
      5       9676.9517      10353.8050

        5 6 S 0 2. W 125.26             * POINT 6
      6       9551.6917      10353.7321

        6 7 N 89 58. W 227.32           * POINT 7
      7       9551.8240      10126.4122

       7 8 N 63 36.5 W  173.19          * POINT 8
      8       9628.8078       9971.2726

FIGURE LINE INTERSECT
           9 1 8 N 66 8. W 17           * POINT 9
         INTERSECTION IS AT SEGMENT   17   21
      9       9694.4203      9822.9768

*  LOCATE INTERSECTIONS OF LOT LINES WITH CUL-DE-SAC
     12 2 4 S 49 13 15 W 25             * POINT 12
         INTERSECTION IS AT SEGMENT   25   14
     12      9806.1581      10174.1871

     13 2 5 N 63 58 30 W 12             * POINT 13
         INTERSECTION IS AT SEGMENT   25   14
     13      9762.3994      10178.8052

     30 2 7 N 1 22. W  13               * POINT 30
         INTERSECTION IS AT SEGMENT   14   23
     30      9746.8812      10121.7586

     15 2 8 N 28 23. E 30               * POINT 15
         INTERSECTION IS AT SEGMENT   14   23
     15      9782.1603      10054.1324

*  DEFINE AND STORE FIGURES
STORE FIGURE
       17 (10 1 2 11 24 C16L 10)
          17
       18 (11 2-4 12 C26L 25 C27R 11)
          18
       19 (12 4 5 13 C26L 12)
          19
       20 (13 5-7 30 14 C26L 13)
          20
       27 (15 30 7 8 15)
          27
       28 (23 15 8 9 C18R 17 C16L 23)
          28
AREA BEARINGS
      17      * LOT 17
     10       9934.3567      9986.5851

                                        N 11-33- 0.0 E       67.0000
      1      10000.0000     10000.0000
                                        S 78-27- 0.0 E      122.6300
      2       9975.4467     10120.1468
                                        S 11-35- 0.0 W      158.0000
     11       9820.6645     10088.4215
                                        N 62-27- 0.0 W      132.9107
     24       9882.1387      9970.5819
```

```
                                                N 67-28-21.8 W       285.5000
CC    16       9991.5204       9706.8663 DELTA= 10-58-38.2 L  L=    54.6989
                                                S 78-27- 0.0 E       285.5000
      10       9934.3567       9986.5851

               AREA=     17313.679 SQ.FT.=    0.39746738 ACRES

      18        * LOT 18
      11       9820.6645       10088.4215
                                                N 11-35- 0.0 E       158.0000
       2       9975.4467       10120.1468
                                                S 78-27- 0.0 E       109.4000
       3       9953.5423       10227.3315
                                                S 72-27-45.0 E        90.3600
       4       9926.3141       10313.4915
                                                S 49-13-15.0 W       183.9652
      12       9806.1581       10174.1871
                                                S 50-36-25.1 W        40.0000
CC    26       9780.7726       10143.2746 DELTA= 71-14-55.1 L  L=    49.7410
                                                N 20-38-30.0 W        40.0000
      25       9818.2047       10129.1737
                                                N 20-38-30.0 W        50.0000
CC    27       9864.9949       10111.5476 DELTA= 48-11-30.0 R  L=    42.0552
                                                S 27-33- 0.0 W        50.0000
      11       9820.6645       10088.4215

               AREA=     22113.413 SQ.FT.=    0.50765411 ACRES

      19        * LOT 19
      12       9806.1581       10174.1871
                                                N 49-13-15.0 E       183.9652
       4       9926.3141       10313.4915
                                                S  9-11- 0.0 E       252.6000
       5       9676.9517       10353.8050
                                                N 63-58-30.0 W       194.7465
      13       9762.3994       10178.8052
                                                N 62-39-21.8 W        40.0000
CC    26       9780.7726       10143.2746 DELTA= 66-44-13.1 L  L=    46.5912
                                                N 50-36-25.1 E        40.0000
      12       9806.1581       10174.1871

               AREA=     23225.311 SQ.FT.=    0.53317977 ACRES

      20        * LOT 20
      13       9762.3994       10178.8052
                                                S 63-58-30.0 E       194.7465
       5       9676.9517       10353.8050
                                                S  0- 2- 0.0 W       125.2600
       6       9551.6917       10353.7321
                                                N 89-58- 0.0 W       227.3200
       7       9551.8240       10126.4122
                                                N  1-22- 0.0 W       195.1127
      30       9746.8812       10121.7586
                                                S 62-27- 0.0 E         3.4007
      14       9745.3083       10124.7737
                                                N 27-33- 0.0 E        40.0000
```

```
CC    26        9780.7726      10143.2746 DELTA= 90-12-21.8 L  L=    62.9757
                                                S 62-39-21.8 E        40.0000
      13        9762.3994      10178.8052
                AREA=     39972.809 SQ.FT.=    0.91764942 ACRES

       27       * LOT 27
      15        9782.1603      10054.1324
                                                S 62-27- 0.0 E        76.2753
      30        9746.8812      10121.7586
                                                S  1-22- 0.0 E       195.1127
       7        9551.8240      10126.4122
                                                N 63-36-30.0 W       173.1900
       8        9628.8078       9971.2726
                                                N 28-23- 0.0 E       174.3064
      15        9782.1603      10054.1324
                AREA=     21598.349 SQ.FT.=    0.49582986 ACRES

       28       * LOT 28
      23        9837.8106       9947.4564
                                                S 62-27- 0.0 E       120.3193
      15        9782.1603      10054.1324
                                                S 28-23- 0.0 W       174.3064
       8        9628.8078       9971.2726
                                                N 66- 8- 0.0 W       162.1624
       9        9694.4203       9822.9768
                                                S 66-41-51.7 E       165.0000
CC    18        9629.1492       9974.5179 DELTA= 30-14-51.7 R  L=    87.1072
                                                N 36-27- 0.0 W       165.0000
      17        9761.8712       9876.4879
                                                N 36-27- 0.0 W       285.5000
CC    16        9991.5204       9706.8663 DELTA= 20-58-33.7 L  L=   104.5218
                                                S 57-25-33.7 E       285.5000
      23        9837.8106       9947.4564
                AREA=     25560.656 SQ.FT.=    0.58679192 ACRES

END OF RUN
```

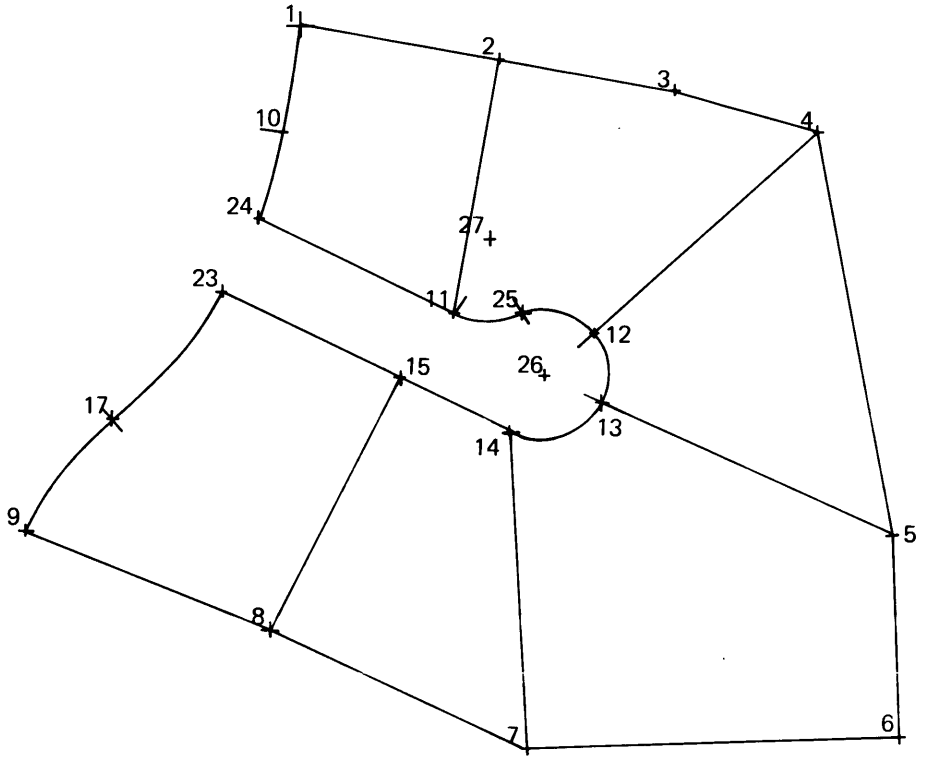


Figure 4-3 Sample Problem 3

APPENDIX A

SUMMARY OF COGO COMMANDS

This appendix lists the command names alphabetically with their abbreviations, associated arguments, and the page number where the command is explained in detail.

Command Name	Valid Abbreviations	Arguments	Page Number
ADJUST ANGULAR ERROR	A A E, A AN ERR, AD AN ER	desc nclos cloaz aerr [grid descrz]	3-29
ADJUST AREA	A AR, AJ A, AD AR	descl ar p1 p2 p3 p4 [desc2]	3-23
ADJUST TRAVERSE COMPASS	A T CP, A T COMP, AD TR CP	desc nclos accur [nstr brglst aerr]	3-27
ADJUST TRAVERSE CRANDALL	A T CR, AD TR CR, A TR CR	desc nclos accur [nstr brglst aerr]	3-27
ADJUST TRAVERSE TRANSIT	A T T, AD TR TR, AD TR TRAN	desc nclos accur [nstr brglst aerr]	3-27
ALIGNMENT	AGN, ANN, ALG, ALN	cid pb ptt pa ntc nc nct r t sb x	3-49
ALIGNMENT OFFSET	AL O, AL OFF	desc p1 psl n ps2	3-47
ANGLES	ANG, AGL	desc	3-78
ANGLES INTERSECT	AN I, AN IN, AN INT	n j angl k ang2 [off1 off2]	3-37
ARC ARC INTERSECT	A A I, A A INT	n pc1 r1 pc2 r2 pi	3-39
ARC LINE AZIMUTH	A L A, A L AZ	n pc r p az pi [off]	3-38
ARC LINE BEARING	A L B, AR LN BR	n pc r p br pi [off]	3-38
ARC LINE POINTS	A L P, AR LN PT	n pc r p1 p2 pi [off]	3-39
AREA	AR, AA	desc	3-72
AREA AZIMUTH	AR A, A AZ, AR AZ	desc	3-72
AREA BEARINGS	AR B, AR BR, A B	desc	3-72
AZIMUTH INTERSECT	AZ I, AZ I, A INT	n p1 az1 p2 az2 [off1 off2]	3-35
BEARING INTERSECT	BR I, B I, B INT	n p1 br1 p2 br2 [off off2]	3-35

Command Name	Valid Abbreviations	Arguments	Page Number
COMPOUND SPIRAL	CO S, CO SP, COM S	cid pb pcs ntt nsc arc dcl dc2 sign	3-61
CONVERT MERIDIAN	C M, CO M, C MR	descl desc2 pl p2 ang [scale]	3-21
COORDINATE OFFSET	C O, C OF, CO OF	n s d	3-53
COORDINATE POA	C POA, CO POA, C PA	n s	3-53
COORDINATE POSP	C POS, C PSP. CO PP	n arc	3-60
CURVE SPIRAL	CV S, CV SP, CV SPL	n r pc pi	3-64
DEFINE CURVE	D CV, DEF C, DF C	cid ntc stc ptt pct sct pc sign	3-52
DELETE COORDINATES	D CO, DL C, DL CO	desc	3-6
DELETE FIGURES	DL F, DL FG	desc	3-7
DESCRIBE ALIGNMENT AZ	D A A, DESC AL AZ	desc pl psl [pol ps2 ps3]	3-47
DESCRIBE ALIGNMENT BR	D A B, DESC AL BR	desc pl psl [pol ps2 ps3]	3-47
DISTANCE	DIS, DT, DST, DIST	desc	3-71
DIVIDE ARC	D A, D AR, DV A	pl p2 pc div [nl]	3-14
DIVIDE FIGURE	DIV F, DIV FIG	desc div nl	3-15
DIVIDE LINE	D L, DV L, D LN	pl p2 div [nl]	3-14
END OF JOB	E/O/J, E O J		3-3
END OF RUN	E/O/R, E O R		3-3

Command Name	Valid Abbreviations	Arguments	Page Number
EXTEND ARC	E A, EX A, E AR	p pc n arc	3-32
FIGURE ARC INTERSECT	F A I, F AR INT	n fgn pc r pi [offfg]	3-41
FIGURE FIGURE INTERSECT	F F I, F FG IN	n fg1 fg2 pi [offfg1 offfg2]	3-42
FIGURE LINE INTERSECT	F L I, FIG LN INT	n fgn p az pi [offfg off]	3-40
FIT ALIGNMENT	F A, F AL, FT A	cid pb ptt dc arc1 arc2 def sign	3-65
FIT CURVE	F C, FIT CRV	p1 p2 p3 n1 nc n2 [r]	3-13
FORESECTION	F, FS, FORE, FRS	n j ang1 k ang2 [off1 off2]	3-37
INVERSE AZIMUTHS	I A, I AZ, IN A	desc	3-72
INVERSE BEARINGS	I B, I BR, IN B	desc	3-72
LINE SPIRAL	L S, LN S, L SP	n p1 p2 pi	3-60
LIST COORDINATES	L C, L CO, LST C	desc	3-69
LIST FIGURES	L F, L FG, LST F	desc	3-70
LIST POINT NUMBERS	L P N, LST P N, L P NUM	desc	3-71
LOCATE ANGLE	L AG, L ANG, LC AN	p1 p2 n ang d	3-31
LOCATE AZIMUTH	L AZ, LOC AZ, LC AZ	p n az d	3-31
LOCATE BEARING	L B, L BR, LOC B	p n br d	3-31
LOCATE DEFLECTION	L D, L DFL, L DEF	p1 p2 n def d	3-32

Command Name	Valid Abbreviations	Arguments	Page Number
LOCATE FROM ALIGNMENT	L F A, L F AL	desc p psl n ps2 [off]	3-47
LOCATE LINE	L L, L LN, LOC L	p1 p2 n d	3-32
OFFSET ALIGNMENT	O A, OF A, O AL	n p	3-54
PARALLEL FIGURE	PR FG, PAR F, PL F	fgn offfg nl	3-16
PARALLEL LINE	PR L, PAR L	p1 p2 off nl n2	3-15
POINTS AZIMUTH INTERSECT	P A I, P AZ I	n p1 p2 p3 az3 [off1 off2]	3-36
POINTS BEARING INTERSECT	P B I, P BR I	n p1 p2 p3 br3 [off1 off2]	3-36
POINTS INTERSECT	P I, PT I, P INT	n p1 p2 p3 p4 [off1 off2]	3-35
POINTS ON ALIGNMENT	P O A, PT O A	desc p psl d nl [off ps2 ps3]	3-46
PUNCH COORDINATES	PN C, PU C, P COR	desc	3-5
PUNCH FIGURES	PU F, PN F, PU FG	desc	3-7
REDEFINE	R, RDF, RDFN	p n	3-5
SET ERROR LIMIT	S E L, S ERR LIM, ST ER LMT	num	3-3
SET OUTPUT DSK:	S O D, S OUT DSK, O DSK	[/name.ext]	3-4
SET OUTPUT LPT:	S O L, S OUT TTY, S O TTY		3-3
SET OUTPUT TTY:	S O T, S OUT TTY, S O TTY		3-3
SEGMENT	SG, SGM, SMT	p1 p2 r	3-73

Command Name	Valid Abbreviations	Arguments	Page Number
SEGMENT MINUS	S M, S MIN, SG M	p1 p2 r	3-74
SEGMENT PLUS	SE P, SG PL, S PLS	p1 p2 r	3-74
SIMPLE CURVE	S C, S CV, SPL C	cid pb ptc ntt nct dc cang sign	3-13
SIMPLE SPIRAL	SM S, SMP SP	cid pb pts ntt nsc +ls dc sign cid pb nts ntt psc -ls dc sign	3-55
SPIRAL LENGTH	S L, SP L	cid pts psc ntt az sign	3-58
SPIRAL OFFSET	S O, SP O, SP OFF	n p	3-59
SPIRAL SPIRAL	SPR S, SRL SP	n pts2 az arc2 r2	3-63
STAKING NOTES	S N, ST N, ST NT	inst ibs rad [desc]	3-75
START OF JOB	S/O/J, S O J	[/name]	3-3
STATION FROM COORDINATES	S F C, S F CO	p	3-54
STATIONS AND OFFSETS	S A O, S A OFF	descl p ps desc2	3-45
STORE	STR	n nor eas	3-5
STORE FIGURE	S F, S FG, ST F	fgn desc	3-6
STREETS INTERSECT	S I, S INT, STR I	r fgl wl fg2 w2 pi ncls nll nlr nrl nrr	3-17

Command Name	Valid Abbreviations	Arguments	Page Number
TANGENT	T, TNG, TT	n1 pc1 r1 n2 pc2 r2 [sign ext]	3-9
TANGENT OFFSET	T O, T OF, T OFF	n p1 p2 p3	3-12
TRAVERSE ANGLES	T AN, TR ANG	desc	3-76
TRAVERSE AZIMUTHS	T AZ, TR AZ	desc	3-76
TRAVERSE BEARINGS	T B, T BR, TRV B	desc	3-76
TRAVERSE DEFLECTIONS	T D, T DEF, T DFL	desc	3-76

APPENDIX B

ERROR MESSAGES

Error Message	Command	Cause of Error
ARC GREATER THAN SPIRAL LENGTH	COORDINATE POSP	The distance along the spiral as specified in the input is greater than the length of the spiral that is presently stored.
AREA NOT FOUND AFTER 20 TRIES	ADJUST AREA	COGO has not been able to find a position for the sides being adjusted that gives the desired area.
BAD DATA	(ANY COMMAND)	The data supplied for this command is not in accordance with the data required for this particular command. The point in the input where the error was detected is underlined. If no underline appears, the error was caused by not enough data. If the cause of the error is not apparent, compare the input against the data codes in Appendix A, Summary of COGO Commands.
DISTANCE IS ZERO	LOCATE AZIMUTH LOCATE BEARING LOCATE ANGLE LOCATE DEFLECTION LOCATE LINE	The distance specified as input to this command is zero. This is a warning message printed to advise the user of an unlikely situation.
FIGURE X CHANGED	STORE FIGURE	This message serves as a warning to the user that he is changing a figure. The old value is printed so that it can be restored if necessary.
FIRST POINT DIFFERENT FROM LAST	AREA AREA AZIMUTHS AREA BEARINGS ADJUST AREA	The user has requested that COGO compute the area on an open traverse.

Error Message	Command	Cause of Error
INVALID COMMAND	(NONE)	The first nonblank column of this record contains an alphabetic character indicating that it is the first letter of a command, but the command as entered cannot be recognized.
INVALID FIGURE X	(MANY)	The number specified as a figure number is not in the range from 1 to 9999.
INVALID NUMBER OF PARTS X	DIVIDE LINE DIVIDE ARC DIVIDE FIGURE	The number of parts specified is not positive.
INVALID POINT X	(MANY)	An attempt has been made to store a point with a point number that is zero or negative or greater than the length of the coordinate area or greater than 9999.
INVALID POINT RANGE X TO Y	DIVIDE LINE DIVIDE ARC DIVIDE FIGURE	The points to be defined as specified by the number of parts and the starting point number include at least one invalid point.
INVALID STATION INTERVAL X	POINTS ON ALIGNMENT	The station interval specified is not positive.
LARGER RADIUS MUST BE FIRST	TANGENT	The first radius entered is smaller than the second, and sign and cross have been specified.
NO ANGLE	DIVIDE ARC	One of the sides of the angle to be divided has zero length and, therefore, has no direction.
NO CURVE DEFINED	COORDINATE POA COORDINATE OFFSET STATION FROM COORDINATE OFFSET ALIGNMENT	These commands require that a curve be previously stored by an alignment or defined curve command.
NO INTERSECTION	(INTERSECTION COMMANDS)	The specified elements do not intersect, or the intersection cannot be computed because of some condition noted in an earlier error message.
NO LINE	PARALLEL LINE	The points specified to define the line have the same coordinates and, therefore, do not define a line.

Error Message	Command	Cause of Error
NO OFFSET POSSIBLE	TANGENT OFFSET SPIRAL OFFSET	No offset can be computed because of a previously noted error condition; or in the case of tangent offset, the two points defining the line are actually the same point, or in the case of spiral offset, the offset does not fall on the spiral; or it has not been found after 1000 tries.
NO PREVIOUS CURVE TO DEFINE STATIONING	ALIGNMENT	The station has been specified as -1 meaning that stationing is to be carried forward from the previous curve, but no previous curve has been stored by an alignment or define curve command.
NO SPIRAL DEFINED	SPIRAL OFFSET COORDINATE POSP LINE SPIRAL SPIRAL SPIRAL CURVE SPIRAL	These commands require a spiral to be stored by a simple spiral, spiral length, or compound spiral command. This has not been done.
NO TANGENT POSSIBLE	TANGENT	No tangent is being computed because of the condition noted in a previous message or because one circle lies entirely within the other or because a cross tangent between intersecting circles has been requested.
POINT X CHANGED	(MANY)	The coordinates being stored for point X are replacing previously stored coordinates for X.
POINT X OR Y OUT OF SEQUENCE	ADJUST AREA	Points X and Y were specified as the starting and ending points of the sides to be adjusted. However, one of these points is not in the description. Or one of them is the first or last point in the description, or Y precedes X in the description.
PREVIOUS JOB TERMINATED		A START OF JOB was read during the current job. The current job is terminated so that the new job can be started. The table in use by the terminated job is updated as if an END OF JOB had been read.

ERROR MESSAGES

Error Message	Command	Cause of Error
RADIUS LESS THAN HALF CHORD	SEGMENT SEGMENT PLUS SEGMENT MINUS	The distance between the two points specified is greater than twice the radius specified. Therefore, the points cannot be on the curve.
SECOND FIGURE SMALLER THAN FIRST	CONVERT MERIDIAN	The description of the points to be defined contains fewer points than the number of points to be converted.
SIDE AT X HAS CHANGED DIRECTION	ADJUST AREA	The adjustment required is so great that the sides of the figure no longer intersect unless extended backwards.
TANGENT LENGTH IS ZERO	FIT CURVE	The two points specified to define one of the tangents are identical and therefore do not define a line.
UNDEFINED DIRECTION	ADJUST AREA LOCATE ANGLE LOCATE DEFLECTION LOCATE LINE	Two points that define a line are identical.
UNDEFINED FIGURE X	(MANY)	This command has requested use of figure X. This figure is not currently defined.
UNDEFINED POINT X	(MANY)	This command has requested use of point X. This point is not currently defined.
UNDEFINED RADIUS	FIT CURVE	The radius was not specified in the input nor is it defined by coincidence of nb and npc or npt and na.

APPENDIX C

OPERATING PROCEDURES

C.1 LOGGING ON THE SYSTEM

Login is the process that identifies the user to the computer system.

C.1.1 TOPS-10 Login

Before login, obtain a project-programmer number and a password.

To begin interaction with the terminal, press the key labeled CTRL and at the same time type a C. This is called typing a control C. After typing a control C, TOPS-10 prints the period as a prompt character.

After the operating system prompt, type LOGIN, type a space, type the appropriate project-programmer number, and press RETURN.

```
.LOGIN 200,200
```

The project-programmer number (PPN) is a 1- to 6-digit project identification number followed by a comma and a 1- to 6- digit programmer identification number.

After the system responds with the password prompt, type the password.

```
PASSWORD:
```

The entire LOGIN procedure is shown as follows:

```
.LOGIN 200,200
PASSWORD:
```

C.1.2 TOPS-20 Login

Before login, obtain a user name, a password, and an account string.

To begin interaction with the terminal, press the key labeled CTRL and at the same time type a C. This is called typing a control C. After typing a control C, TOPS-20 prints a system message and the at sign as a prompt character.

After the operating system prompt, type LOGIN, and press the ESC key to call the guide word for the next argument.

```
@LOGIN (USER)
```

After the guide word (USER), type the user name, and press the ESC key.

```
@LOGIN (USER) COGO (PASSWORD)
```

After (PASSWORD), type the password (the characters of the password are not printed), and press the ESC key.

```
@LOGIN (USER) COGO (PASSWORD) (ACCOUNT)
```

After (ACCOUNT), type the account string.

```
@LOGIN (USER) COGO (PASSWORD) (ACCOUNT) PROJ.TASK
```

To complete the LOGIN procedure, press RETURN.

C.2 SPECIFYING A FILE

All programs, data, and text must be written into files. Each file is labeled and stored. The label on a file is called a file specification.

C.2.1 TOPS-10 File Specification

The TOPS-10 file specification format is as follows:

```
dev:filename.ext[proj.prog]
```

where

dev:	The 3- to 6-character name of the physical device containing the file.
filename	The filename specifies the name of the file.
.ext	The file extension specifies the general purpose of the file.
[proj,prog]	Two octal numbers separated by commas and enclosed by square brackets. The project-programmer number identifies the user and the user's file storage area on the file structure.

C.2.2 TOPS-20 File Specification

The TOPS-20 file specification format is as follows:

```
dev:<dir>filename.typ.gen
```

where

dev:	The physical or logical structure containing the file.
<dir>	The directory name identifies the user's file storage area on the file structure.
filename	The filename specifies the name of the file.

.typ	The file type specifies the general purpose of the file.
.gen	The generation number indicates the number of times a file is updated.

C.3 CREATING AN INPUT FILE

Two methods to create a COGO input file are as follows:

- Keypunch data on cards.
- Use a text editor.

C.3.1 Using Cards

When using cards to create the COGO input file, keypunch the commands and data as specified in Chapter 3. After the deck is keypunched, run COGO using the cards as input. Specify the card reader as the input device when entering the filename to COGO. The same procedure can be used for any subsequent runs.

Some cards can be used from run to run.

If dealing with cards becomes cumbersome, transfer the information from the card reader to another input device. Section C.8 describes how to transfer or copy files.

C.3.2 Using The Text Editor

Using a text editor requires the user to be familiar with editing commands.

At any time after login, the text editor is called by typing SOS for TOPS-10 or CREATE for TOPS-20 in response to the operating system prompt.

Enter the commands and data as described in Chapter 3. Because COGO accepts data in a free format (not position or column justified), the data for a command can be continued on as many lines as required. COGO continues to read data after the command until all input requirements are satisfied. Until a new command is entered, COGO assumes that the same commands repeats and will continue to read data. After each line of input, press the RETURN key.

C.4 LISTING A FILE

To list a file on the user terminal, give the following command after the operating system prompt:

```
TYPE filespec
```

To list a file on the line printer, give the following command after the operating system prompt:

```
PRINT filespec
```

For more information on the TYPE and PRINT commands, see the DECSYSTEM-20 User's Guide and the DECsystem-10 Operating System Commands Manual.

C.5 RUNNING COGO

To run COGO, type the following command after the operating system prompt:

```
RUN COGO
```

COGO responds with a program identification and the following prompt:

```
SPECIFY INPUT DEVICE/FILENAME>
```

The user response to the prompt is either an input device name for conversational node input or a file specification for the disk file containing the COGO input data. Possible responses are as follows:

TTY:	Input data is accepted in a conversational mode from the user's terminal. The COGO prompt for input data is the character >.
xxxxxx	Input data is read from the file xxxxxx.DAT, where xxxxxx is a filename with a maximum of six characters.
xxxxxx.	Input data is read from the file xxxxxx., where xxxxxx is a filename with a maximum of six characters. No file extension is used.
xxxxxx.ext	Input data is read from the file xxxxxx.ext, where xxxxxx is a filename with a maximum of six characters and ext is a file extension with a maximum of three characters.

After answering the previous prompt and pressing RETURN, COGO responds with the following prompt:

```
SPECIFY OUTPUT DEVICE/FILENAME>
```

The user response to the prompt is either an output device name or a file specification for COGO output. Possible responses are as follows:

TTY:	COGO output is printed directly on the user's terminal.
LPT:	COGO output is printed directly on the system line printer.
yyyyyy	COGO output is written to the file yyyyyy.DAT, where yyyyyy is a filename with a maximum of six characters.
yyyyyy.	COGO output is written to the file yyyyyy., where yyyyyy is a filename with a maximum of six characters. No file extension is used.

yyyyyy.ext	COGO output is written to the file yyyyyy.ext, where yyyyyy is a filename with a maximum of six characters and ext is a file extension with a maximum of three characters.
RET	If neither a device nor a filename is specified, COGO output is written to the file FOR03.DAT.

C.6 DIRECTORY LISTINGS

Each user has a directory file that contains a list of the files stored in the user's directory. To obtain a directory listing, give the following command after the operating system prompt:

```
DIRECTORY
```

C.7 DELETING FILES

To delete a file, give the following command after the operating system prompt:

```
DELETE filespec
```

C.8 COPYING FILES

To copy a file, give the following command after the operating system prompt:

```
COPY filespec-1 filespec-2
```

This procedure can be used to create a back-up copy of current input files.

To use DECtape or magtape, see the DECsystem-10 Operating System Commands or the DECSYSTEM-20 User's Guide.

C.9 ABORTING COGO

To terminate a run before its logical conclusion, press CTRL/C twice.

To leave COGO after completing a run, type END OF RUN in response to the COGO prompt.

C.10 LOGGING OFF THE SYSTEM

To logout, give the following command after the operating system prompt:

--- TOPS-10 ---

```
KJOB
```

--- TOPS-20 ---

```
LOGOUT
```

Glossary

Ahead Tangent

The tangent exiting from a curve.

Alignment

A series of tangents and circular curves that describe the centerline of a highway or easement.

Angle (ang)

Circular measurement taken from the intersection of two lines; given in degrees, minutes, and seconds where degrees and minutes are integers and seconds may contain a decimal part and are measured in a clockwise direction.

Arc

Arc length for curves in a horizontal plane. In circular curves, arc is measured from tc to ct; in spirals from ts to sc. (See point names.)

Back Tangent

The tangent entering a curve.

Bearing (br)

An angle less than 90 degrees between a line and a north-south line and measured either clockwise or counterclockwise from either north or south. Denoted by a quadrant number and angle or two direction notations.

CC

Center of curvature.

cid

The curve identification number of a circular curve or a spiral. It must be an integer in the range 0 to 999.

cl

Curve length.

Closure Distance

The length of the closure line.

Closure Line

The line between the actual end point of a traverse and the desired end point.

Command Statements

Predefined codes used with data files to define the actual geometry problem.

Command String

A line of input containing a command name or code and a list of data fields.

Compound Curve

A curve with two or more different degrees of curvature, both in the same direction.

Coordinate Table

A set of up to 9999 points upon which COGO subprograms operate.

d

A linear distance.

Data Field

A portion of an input string that has a specific meaning to the program.

dc

The degree of curvature of a circular curve. The dc is defined as the central angle which subtends a 100-foot arc.

Default

A predetermined value that is assumed if a user specified value is not entered.

Deflection Angle (def)

Clockwise measurement between the extension of a line and another line.

Degree of Curvature

Central angle of a circle which subtends a 100-foot arc and equal to $18000/\pi r$ degrees where r is the radius of the circle.

div

Number of divisions.

eas

The distance of a point east of the origin.

Error Message

A message telling the user that an error has occurred. It usually contains information to help the user fix the error.

ext

In the TANGENT command, ext=1 indicates an external tangent; ext=-1 indicates an internal, or cross tangent. In a file specification, ext is the extension.

Field

See Data field.

Format

An organization of information.

g

Grade in percent.

Independent Curve

A curve that has no relationship to previous curves.

I/O

An abbreviation for Input/Output

Input

Information read into the computer.

Input File

A device or a storage area on disk from which data or commands are read to run a program.

Integer

Whole number containing neither a fractional part nor a decimal point.

Interactive Mode

A method of using COGO in which both input and output are performed at a keyboard terminal.

ls

The length of a spiral curve. The distance is defined as the length along the spiral from ts to sc, or cs to sc in the case of a compound spiral.

Method of Least Squares

A method which matches two functions, f(x) and g(x), by satisfying the equation;

$$[f(x)-g(x)]2dx=\text{minimum}$$

Output

Information transferred from the computer to the user via some peripheral device.

Overflow

A condition that occurs when the computer attempts to put information in a space too small for it.

Point Names

The following conventions have been followed in the naming of points in alignment and spiral commands.

A point name consists of a prefix and/or a suffix. The prefix determines whether the point has been defined, and the suffix determines where the point occurs in the curve.

Prefixes:

p denotes a predefined point with coordinates stored in the coordinate table.

n Denotes a new point whose coordinates are to be found.

Suffixes:

s denotes the station of a predefined point.

a denotes a point on the ahead tangent.

b denotes a point on the back tangent.

c denotes the center of a circle or circular arc.

tc denotes the point of transition from tangent to curve.

ct denotes the point of transition from curve to tangent.

ts denotes the point of transition from tangent to spiral.

st denotes the point of transition from spiral to tangent.

sc denotes the point of transition from spiral to circular curve.

cs denotes the point of transition from circular curve to spiral.

tt denotes the point of intersection of two tangents.

Polygon

A multiple-sided figure made by the intersection of lines in the same plane.

Quadrant

Indicates which 90 sector a bearing falls in. NE=1, SE=2, SW=3, and NW=4.

r

Radius of circle.

Reverse Curve

A curve which changes direction from clockwise to counterclockwise, or vice-versa.

Segment

A figure made by the intersection of a circular arc and a chord.

sign

Indicates whether a curve turns right or left. When traveling from the back tangent to the ahead tangent, sign=1. indicates a right turn, and sign=-1. a left turn.

Special Operators

A comparative value that can be substituted in a command for actual distance, angle, azimuth, and bearing measurements.

Spiral In

Term describing a transition spiral with decreasing radius; that is, going from a line to a circle.

Spiral Out

Term describing a transition spiral with increasing radius; that is, going from a circle to a line.

Station

The length of a road along a path from a starting point to a point on an alignment, measured, in the horizontal plane.

t

Tangent length, measured from the point of tangency to the intersection of the tangents.

x

In an alignment, x measures the distance from pb to tc.

INDEX

COGO-10/20
User's Manual
AA-5510A-TK

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

_____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Please cut along this line.

---------- Fold Here ----------

---------- Do Not Tear - Fold Here and Staple ----------

FIRST CLASS
PERMIT NO. 152
MARLBOROUGH, MA
01752

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
200 Forest Street MR1-2/E37
Marlborough, Massachusetts 01752