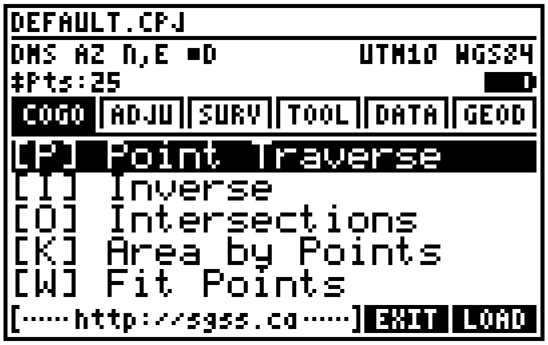
5 COGO Menu



5.1 Point Traverse

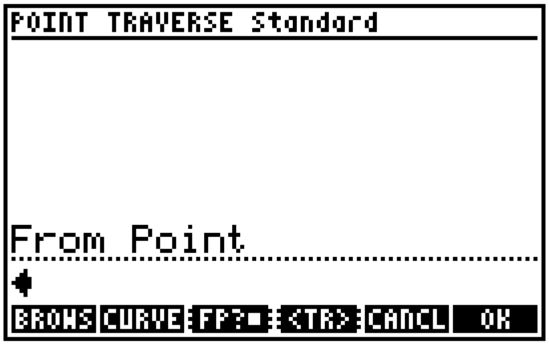
Point Traverse is the main COGO application and is available in two modes, the *Standard* mode and the *Sideshot* mode. The two modes operate completely different. The *Standard* mode is more powerful while the *Sideshot* mode is in some ways simpler to learn/use.

Standard Mode

Point Traverse *Standard* is a complete COGO solution with **Inverse** and **Intersections** built into the command line interface. Input screens within this application often take multiple types of input to allow a wide variety of possible uses.

‘From Point’ Screen

The *From Point* input screen prompts the user to input a point number to use as a starting point for further calculations. Input types accepted:



1. An existing point number – The program ensures the point exists, and then displays the next screen.
2. A non-existing point number – The user may enter a point number that has not yet been stored in the database. An input form will open to allow the user to enter coordinates for the new point.
3. Two point numbers in the format “From..To” – Calculate a point inverse between two points in the job. The *From Point* input screen is re-displayed after this input type is processed. For example, input *1..2* to calculate the inverse from Point 1 to Point 2.
4. Three point numbers in the format “Start..End..Offset” – Calculate a point to line inverse by entering the baseline start and end points and the offset point. The *From Point* input screen is

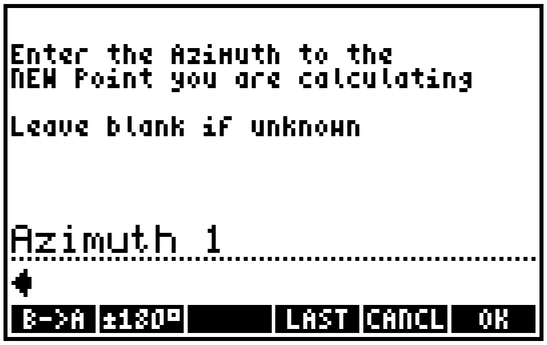
re-displayed after this input type is processed. For example, input *1..2..3* to calculate the offset of Point 3 from the line defined by points 1 and 2.

5. Three point numbers in the format "BC+CC+EC" – Calculate a curve inverse, direction 'right', by entering the beginning of curve, curve center and end of curve points separated by the '+' character. The *From Point* input screen is re-displayed after this input type is processed. For example, input *1+2+3* to inverse a curve connecting Point 1 and Point 3 in a clockwise direction with curve center (radius point) at Point 2.
6. Three point numbers in the format "BC-CC-EC" – Calculate a curve inverse, direction 'left', by entering the beginning of curve, curve center and end of curve points separated by the '-' character. The *From Point* input screen is re-displayed after this input type is processed. For example, input *1-2-3* to inverse a curve connecting Point 1 and Point 3 in a counter clockwise direction with curve center (radius point) at Point 2.
7. F1 **BROWS** – Opens the Point Browser to review points in the job and pick one from a list.
8. F2 **CURVE** – Starts the Curve Traverse program.
9. F3 **FP?■** – Toggles the *From Point* number suggestion setting. When the square is displayed after the "FP?", the setting is enabled and the program will suggest a point number automatically, otherwise no suggestions are made.
10. F4 **<TR>** or **<SS>** – Works in tandem with the **FP?** softkey to suggest point numbers by TRAVERSE or SIDESHOT methods. TRAVERSE always suggests the last saved point, while SIDESHOT keeps the point unchanged until changed by the user.
11. F5 **CANCL** – Exits the **Point Traverse** *Standard* program, the same as pressing the ON key.
12. F6 **OK** – Accepts the input provided, the same as pressing the ENTER key.
13. All other input is ignored or results in an error message.

'Azimuth 1' or 'Bearing 1' Screen

The direction reference setting determines whether *Azimuth* or *Bearing* is displayed. This screen prompts the user to input the azimuth or bearing to the new point from the FROM POINT. Input types accepted:

1. Azimuth or Bearing – The real number entered is interpreted based on the current angle unit and direction reference user settings and the next screen is displayed.



Enter the Azimuth to the NEW Point you are calculating

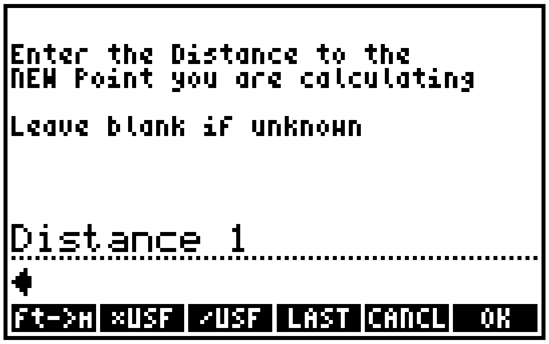
Leave blank if unknown

Azimuth 1

2. Any of the standard directions input options.
3. Any of the above input options followed by "++Offset" or "--Offset" – Define a parallel offset from the From Point. The '++' denotes an offset 'right' while '--' denotes an offset 'left'. For example, input *15.3025++5.5* to enter a parallel offset of 5.5 units to the 'right' of the From Point and perpendicular to the entered azimuth 15°30'25".
4. Leave blank, no input – Signals that the azimuth or bearing to the new point is unknown, which leaves the possibility of a Distance-Azimuth, a Distance-Distance, or a Distance-Angle intersection.
5. F1 **B->A** or **A->B** – Converts the input between bearings and azimuths. The appearance and action of this softkey varies depending on your direction reference setting.
6. F2 **±180°** – Flips the direction of the number in the command line by adding/subtracting 180 degrees (or 200 gons). THIS OPTION IS ONLY AVAILABLE WHEN AZIMUTHS ARE SET AS THE DIRECTION REFERENCE.
7. F4 **LAST** – Inserts the most recent azimuth or bearing input into the command line.
8. F5 **CANCL** – Cancels the traverse and returns the program to the From Point screen.
9. F6 **OK** – Accepts the input as provided.

'Distance 1' Screen

This screen prompts the user to input the distance to the new point from the From Point. Input types accepted:



Enter the Distance to the
NEW Point you are calculating
Leave blank if unknown

Distance 1

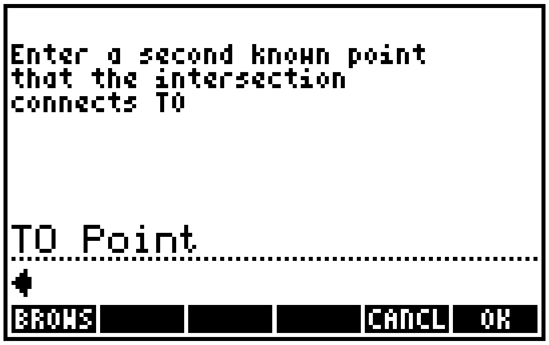
1. A distance – The number entered is used as the distance and the next screen is displayed.
2. Any of the standard distances input options.
3. Leave blank, no input – Signals that the distance to the new point is unknown, which leaves the possibility of an Azimuth-Azimuth or an Azimuth-Distance intersection, provided that the *Azimuth 1* input was given.
4. F1 **F->M** or **M->F** – Converts the input between metric and imperial units. The appearance and action of this softkey varies depending on your primary distance unit setting.
5. F2 **xUSF** – Multiplies the input by the user defined scale factor. THIS CAN BE USED FOR SCALING VALUES GRID<>GROUND AS YOU GO, OR COULD BE AN ALTERNATIVE METHOD TO CONVERT BETWEEN UNITS.
6. F3 **/USF** – Divides the input by the user defined scale factor.
7. F4 **LAST** – Inserts the most recent distance input into the command line.

8. F5 **CANCL** – Cancels the traverse and returns the program to the FROM POINT screen.

9. F6 **OK** – Accepts the input as provided.

‘TO Point’ Screen

This screen accepts the point number of a second known point that an intersection connected **to**. This screen appears when either the *Azimuth 1* or *Distance 1* inputs are unknown and left blank.

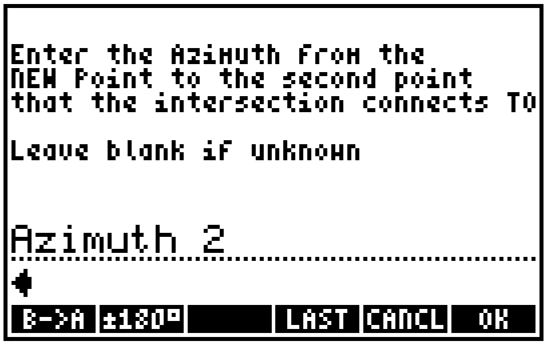


Enter a second known point
that the intersection
connects TO

TO Point

‘Azimuth 2’ or ‘Bearing 2’ Screen

This screen has two possible variations depending on whether *Azimuth 1* or *Distance 1* is known. In both cases the screen prompts the user to enter the azimuth or bearing from the new point that is being calculated TO the second known point. This screen accepts the same input types as the Azimuth 1 screen.

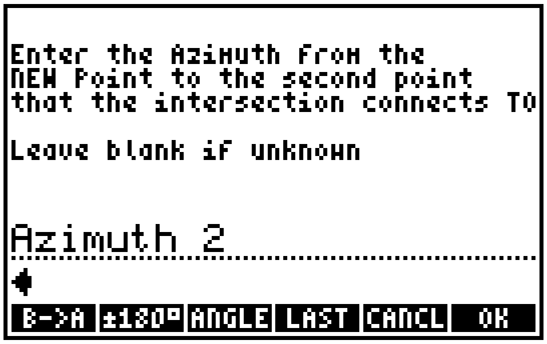


Enter the Azimuth from the
NEW Point to the second point
that the intersection connects TO

Leave blank if unknown

Azimuth 2

With a known *Distance 1* input there exists a possibility to perform a Distance-Angle intersection. This option is available through the F3 **ANGLE** softkey. The included angle TO the second known point at a certain distance from the FROM POINT is required for this type of intersection.



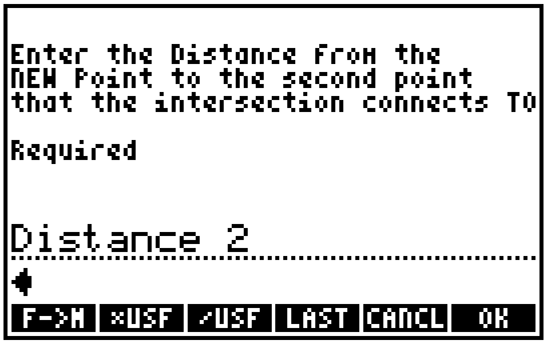
Enter the Azimuth from the
NEW Point to the second point
that the intersection connects TO

Leave blank if unknown

Azimuth 2

‘Distance 2’ Screen

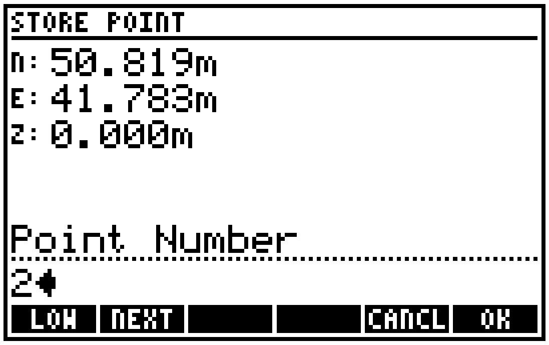
This screen prompts the user to enter the distance from the new point that is being calculated TO the second known point. This screen accepts the same input types as the Distance 1 screen.



```
Enter the Distance from the
NEW Point to the second point
that the intersection connects TO

Required

Distance 2

F->M  xUSF  /USF  LAST  CANCL  OK
```

STORE POINT Screen

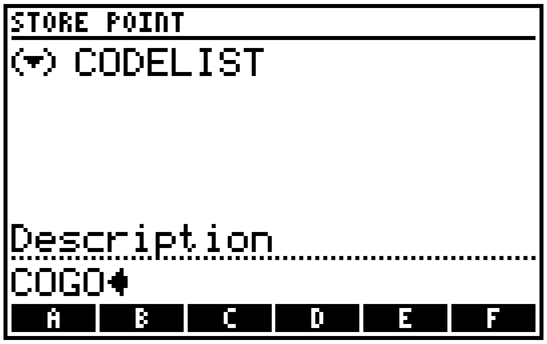
The STORE POINT screen displays the coordinates of the solved point and prompts the user to enter a point number. The menu features a few point searching capabilities:



```
STORE POINT
N: 50.819m
E: 41.783m
Z: 0.000m

Point Number
2
LOW  NEXT              CANCL  OK
```

1. F1 **LOW** – Inserts the lowest unused point number into the command line.
2. F2 **NEXT** – Inserts the next lowest unused point number starting from the currently entered value.

A screen prompting to enter a point description will follow if the description prompts toggle is set. Alpha mode is automatically set when this screen becomes active and the menu labels display the Alpha characters. The ▼ cursor key opens the codelist to select a point description. When the codelist translation toggle is set, the user can enter any defined code in the codelist and the program will automatically look up the description and store the code’s description.



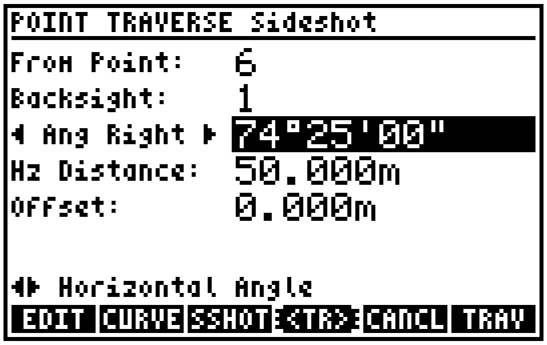
```
STORE POINT
(▼) CODELIST

Description
COGO
A  B  C  D  E  F
```

Sideshot Mode

Point Traverse *Sideshot* is a COGO application that accepts all input within a single input form with a few variations of possible input types.

‘From Point’ Field

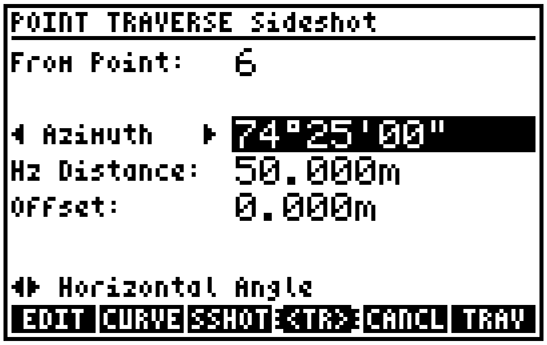
This field requires an existing point number to use as the starting point, or station. Entering a point number that does not exist in the job will open an input form to allow the user to enter coordinates for the new point.



```
POINT TRAVERSE Sideshot
From Point:    6
Backsight:     1
◀ Ang Right ▶ 74°25'00"
Hz Distance:   50.000m
Offset:        0.000m

◀▶ Horizontal Angle
```

‘Backsight’ Field

This field requires an existing point number to use as the backsight point. This field disappears when the horizontal angle field label is set to *Azimuth* or *Bearing* since it becomes unnecessary, but is required for an *Angle Right* or *Angle Left* calculation.



```
POINT TRAVERSE Sideshot
From Point:    6

◀ Azimuth    ▶ 74°25'00"
Hz Distance:   50.000m
Offset:        0.000m

◀▶ Horizontal Angle
```

‘Angle Right’ / ‘Angle Left’ / ‘Azimuth’ or ‘Bearing’ Field

The label for this field changes when the user toggles the input types for this field with the ◀ and ▶ cursor keys. This field requires a real number angle when *Angle Right* or *Angle Left* is selected. Input types accepted when *Azimuth* or *Bearing* is selected:

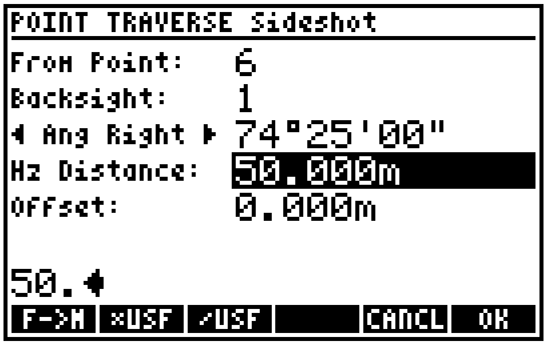
1. Azimuth or Bearing – The real number entered is interpreted based on the current angle unit and direction reference user settings.
2. Any of the standard directions input options.

‘Hz Distance’ Field

This field requires the distance to the new point from the FROM POINT. Input types accepted:

1. A distance – The number entered is used as the distance.
2. Any of the standard distances input options.

When the command line becomes active, i.e. a distance is being entered or edited, the softmenu will update to show more options:



```
POINT TRAVERSE Sideshot
From Point:    6
Backsight:     1
◂ Ang Right ▸ 74°25'00"
Hz Distance:   50.000m
Offset:        0.000m

50.◂
F->M xUSF /USF      CANCL  OK
```

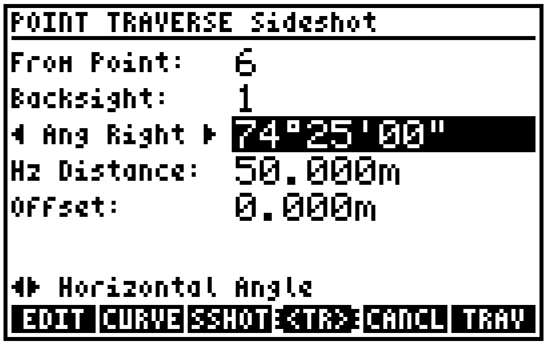
1. F1 F->M or M->F – Converts the input between metric and imperial units. The appearance of this softkey varies depending on your primary distance unit setting.
2. F2 xUSF – Multiplies the input by the user defined scale factor. THIS CAN BE USED FOR SCALING VALUES GRID<>GROUND AS YOU GO, OR COULD BE AN ALTERNATIVE METHOD TO CONVERT BETWEEN UNITS.
3. F3 /USF – Divides the input by the user defined scale factor.

'Offset' Field

This field accepts a perpendicular offset value from the line of direction. A positive offset is to the right while a negative offset is to the left. This field accepts the same types of inputs as the *Hz Distance* field.

The Menu

The **Point Traverse** *Sideshot* input form menu:



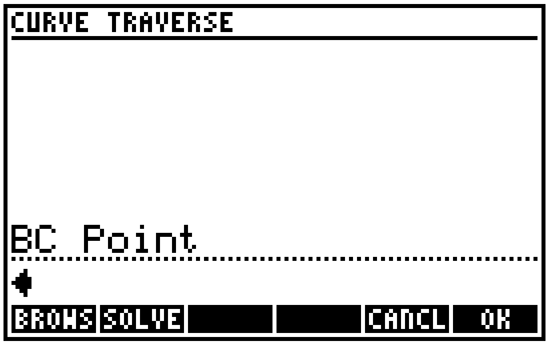
```
POINT TRAVERSE Sideshot
From Point:    6
Backsight:     1
◂ Ang Right ▸ 74°25'00"
Hz Distance:   50.000m
Offset:        0.000m

◂▸ Horizontal Angle
EDIT CURVE SSHOT <TR> CANCL TRAV
```

1. F1 EDIT – Edits the currently selected field by copying the field's contents to the command line and activating the command line.
2. F2 CURVE – Starts the Curve Traverse program.
3. F3 or F6 SSHOT – Takes the input provided and attempts to store the information as a sideshot. When successful, the STORE POINT screen is displayed to store the new point and the original **Point Traverse** *Sideshot* input form is re-displayed, unchanged.
4. F4 <TR> or <SS> – Toggles the location of the TRAV and SSHOT softkeys between F3 and F6.
5. F5 CANCL – Exits the **Point Traverse** *Sideshot* program.
6. F6 or F3 TRAV – Takes the input provided and attempts to store the information as a traverse point. When successful, the STORE POINT screen is displayed to store the new point and the original **Point Traverse** *Sideshot* input form is re-displayed, with the newly stored point in the *From Point* field, and the previous FROM POINT set as the BACKSIGHT point.

Curve Traverse

The **Curve Traverse** program is accessible from the main screen of **Point Traverse** *Standard* and **Point Traverse** *Sideshot*. The program allows the user to enter a beginning of curve (BC) point and a radius point, then enter a known curve element and curve direction to solve the end of curve (EC) point.

'BC Point' Screen

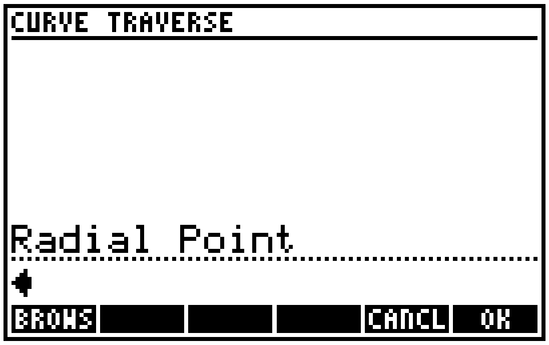


CURVE TRAVERSE

BC Point

This screen prompts the user to enter the beginning of curve point number. A point number is automatically suggested for this input, usually the previously stored point, which can be useful when calculating multiple points along the same arc. F2 **SOLVE** opens the horizontal curve solver.

'Radial Point' Screen

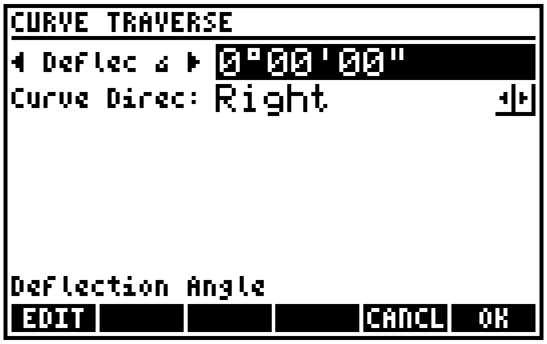


CURVE TRAVERSE

Radial Point

This screen prompts the user to enter the radius point number. The radius point input is remembered for the next use until the user quits the **Curve Traverse** program.

CURVE TRAVERSE Form



CURVE TRAVERSE

◀ Deflec ∠ ▶ 0°00'00"

Curve Direc: Right

Deflection Angle

This input form accepts the known element of the curve and the curve direction.

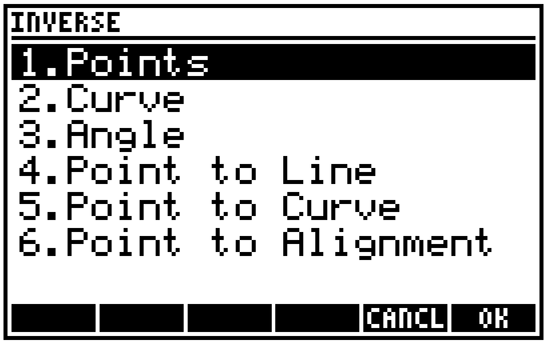
The label for the first field changes when the user toggles the input types for this field with the ◀ and ▶ cursor keys. Available input options include the curve deflection angle, arc length, chord length or the tangent length.

The second field is a Right or Left option for the curve direction. Direction 'Right' is always clockwise, while 'Left' is counter clockwise.

The standard STORE POINT screen follows valid input to store the EC point. The **Curve Traverse** program continues until cancelled.

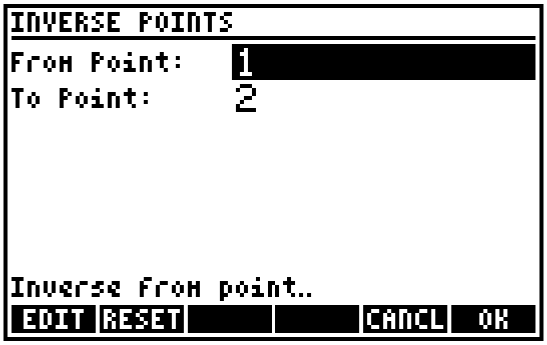
5.2 Inverse

Inverse Points, **Inverse Curve**, **Inverse Angle**, **Inverse Point to Line**, **Inverse Point to Curve** and **Inverse Point to Alignment** are available options for inversing with point coordinates in the current job database.

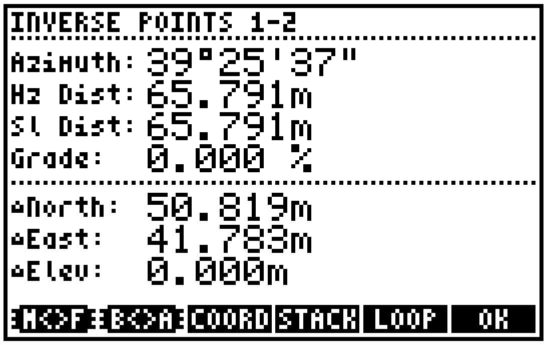

```
INVERSE
1.Points
2.Curve
3.Angle
4.Point to Line
5.Point to Curve
6.Point to Alignment
                CANCL  OK
```

Inverse Points

Enter the *From Point* and *To Point* to calculate the inverse information between any two points in the job database.

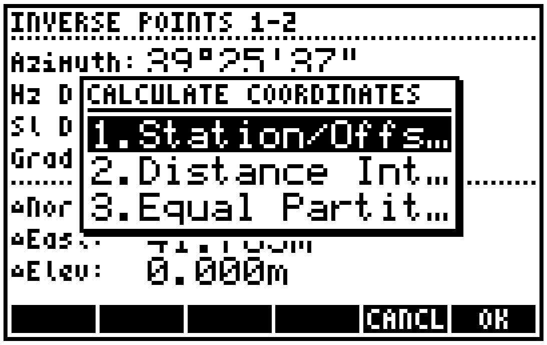

```
INVERSE POINTS
From Point:   1
To Point:     2

Inverse from point..
EDIT RESET      CANCL  OK
```

The results screen displays the azimuth/bearing, horizontal distance, slope distance, slope grade, and coordinate differences between the two points. The menu:


```
INVERSE POINTS 1-2
Azimuth: 39°25'37"
Hz Dist: 65.791m
Sl Dist: 65.791m
Grade:   0.000 %
ΔNorth:  50.819m
ΔEast:   41.783m
ΔElev:   0.000m
M<>F B<>A COORD STACK LOOP OK
```

1. F1 M<>F – Toggles metric/imperial.
2. F2 B<>A – Toggles bearings/azimuths.
3. F3 COORD – Presents three options to solve coordinates between the two points. The options are by **Station/Offset**, by **Distance Interval**, or by **Equal Partitions**.
4. F4 STACK – Exports the direction, horizontal and slope distances to the stack.
5. F5 LOOP - Loops through the current job database inversing sets of two points in the order they were stored.

Solve COORDinates



INVERSE POINTS 1-2

Azimuth: 39°25'37"

Hz D

Sl D

Grad

ΔNor

ΔEas:

ΔElev: 0.000m

CALCULATE COORDINATES

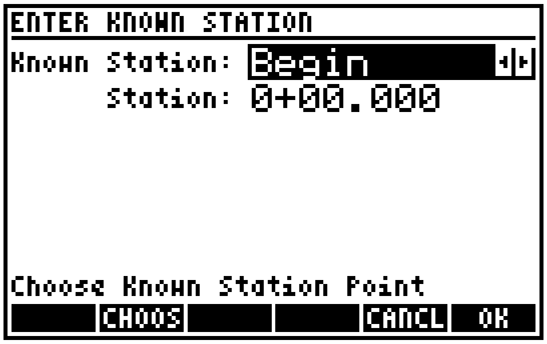
1.Station/Offs…

2.Distance Int…

3.Equal Partit…

Pick this option to calculate coordinates between the two points. **Station/Offset** accepts any station and offset to calculate the coordinate of a point, **Distance Interval** will store points at the given distance interval, and **Equal Partitions** will divide the line into equal partitions, storing a point at each calculated partition.

Station/Offset



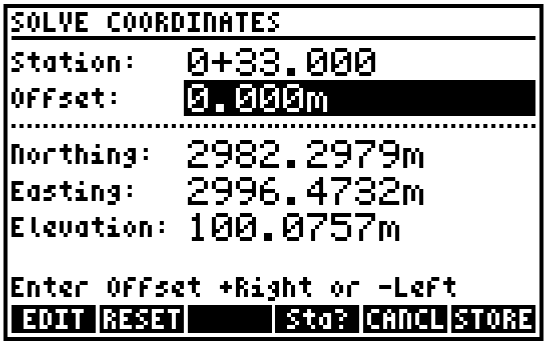
ENTER KNOWN STATION

Known Station: Begin

Station: 0+00.000

Choose Known Station Point

The first input form requires the selection of a *Known Station,* either the beginning (*From Point*) or the end (*To Point*) of the straight line, and a *Station* input at that point to calculate coordinates at any station and offset between the two points.



SOLVE COORDINATES

Station: 0+33.000

Offset: 0.000m

Northing: 2982.2979m

Easting: 2996.4732m

Elevation: 100.0757m

Enter Offset +Right or -Left

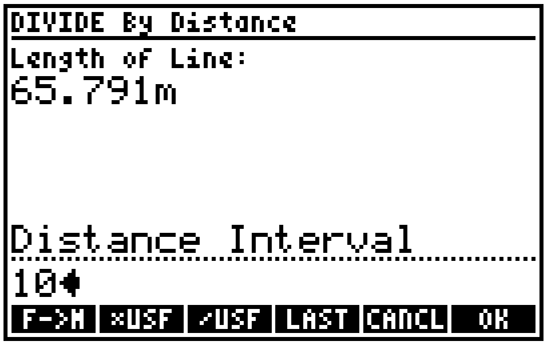
The second input form accepts a *Station* and a perpendicular *Offset* from the line. A positive offset is to the right while a negative offset is to the left. The 3D coordinates of the calculated point are immediately displayed on the screen. The menu:

1. F4 **Sta?** – Select a position to jump to on the line; beginning, mid-point or end of the line. Sets the *Station* field value to the selected station and sets the *Offset* field to zero.
2. F6 **STORE** – Stores the calculated coordinates as a point in the job database.

NOTE: Stations are displayed with a "<" character preceding the station when the station is less than the station of the beginning of the line, and displayed with a ">" character preceding the station when the station is greater than the end of the line. Coordinates can be calculated for any station along the projection of the defined line.

Distance Interval

The first input screen will display the length of the line and accept a distance interval value. Points will be created every specified distance starting at the *From Point* of the line until the remainder to the *To Point* of the line is less than or equal to the distance interval specified.



DIVIDE By Distance

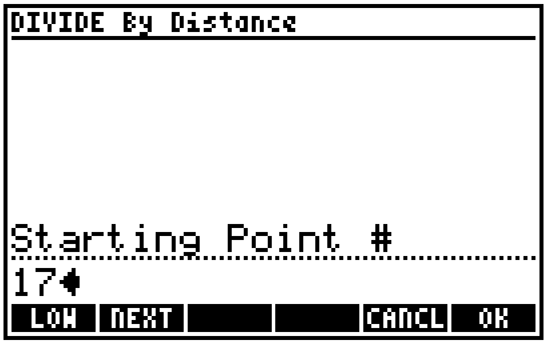
Length of Line:

65.791m

Distance Interval

10◆

The second input screen prompts for a Starting Point Number to be used for storing the points that will be calculated. Unused point number will be assigned to the positions starting at the specified point number. Point coordinates are calculated in three dimensions between the *From Point* and *To Point*.



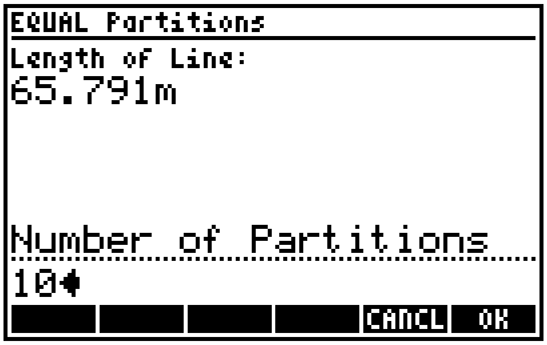
DIVIDE By Distance

Starting Point #

17◆

Equal Partitions

The first input screen will display the length of the line and accept the number of partitions to divide the line into. Points will be created at the calculated interval so that the specified number of equal partitions will be created between the *From Point* and *To Point*.



EQUAL Partitions

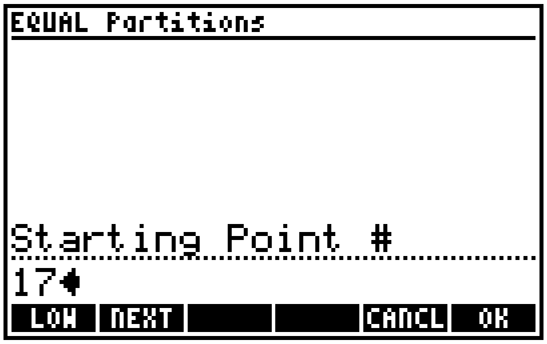
Length of Line:

65.791m

Number of Partitions

10◆

The second input screen prompts for a Starting Point Number to be used for storing the points that will be calculated. Unused point number will be assigned to the positions starting at the specified point number. Point coordinates are calculated in three dimensions between the *From Point* and *To Point*.



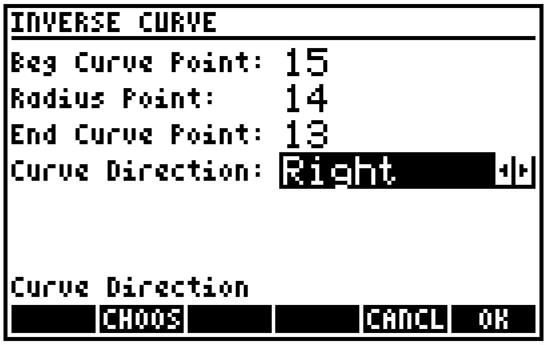
EQUAL Partitions

Starting Point #

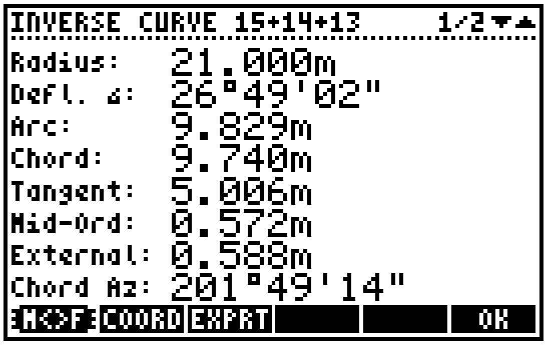
17◆

Inverse Curve

Enter the *Beginning of Curve Point*, the *Radius Point* and the *End of Curve Point*, and choose the *Curve Direction* to calculate the curve information.


INVERSE CURVE
Beg Curve Point: 15
Radius Point: 14
End Curve Point: 13
Curve Direction: Right
Curve Direction

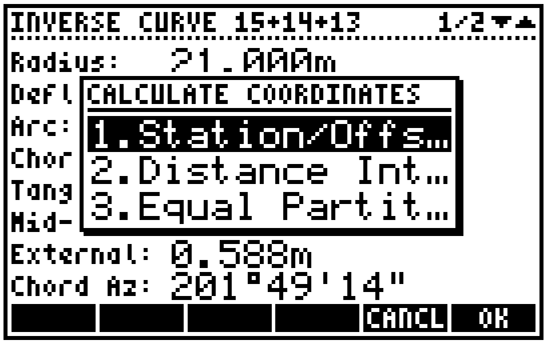
The results screen displays the radius, deflection angle, arc length, chord length, tangent length, mid-ordinate length and the external length. Use the ▼and ▲ keys to toggle between the first and second page of the results. Page 2 displays the sector, segment and fillet areas of the curve. The menu:


INVERSE CURVE 15+14+13 1/2
Radius: 21.000m
Defl. ∠: 26°49'02"
Arc: 9.829m
Chord: 9.740m
Tangent: 5.006m
Mid-Ord: 0.572m
External: 0.588m
Chord Az: 201°49'14"

1. F1 **M<>F** – Toggles metric/imperial.
2. F2 **COORD** – Presents three options to solve coordinates between the end points along the curve. The options are by **Station/Offset**, by **Distance Interval**, or by **Equal Partitions**.
3. F3 **EXPRT** – Exports the results to the stack, to an ASCII file, or copy to the clipboard.

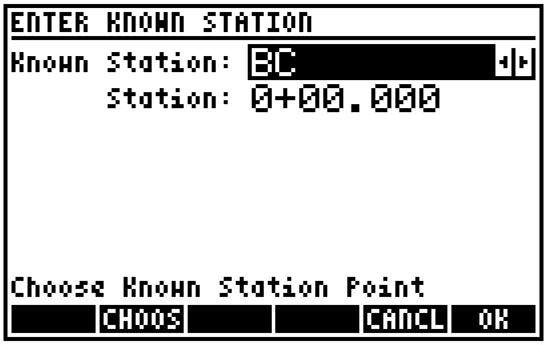
Solve COORDinates

Pick this option to calculate coordinates between the end points of the curve. **Station/Offset** accepts any station and offset to calculate the coordinates of a point, **Distance Interval** will store points at the given distance interval, and **Equal Partitions** will divide the curve into equal partitions, storing a point at each calculated partition.


INVERSE CURVE 15+14+13 1/2
Radius: 21.000m
Defl
Arc:
Chor
Tang
Mid-
External: 0.588m
Chord Az: 201°49'14"

Station/Offset

The first input form requires the selection of a *Known Station* (BC, PI or EC) on the curve, and a *Station* input at that point to calculate coordinates at any station and offset along the curve.



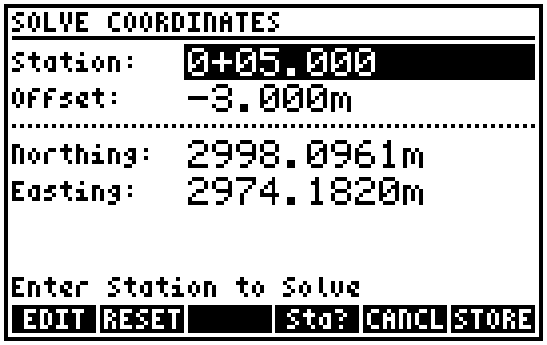
ENTER KNOWN STATION

Known Station: BC

Station: 0+00.000

Choose Known Station Point

The second input form accepts a *Station* and a perpendicular *Offset* from the curve. A positive offset is to the right while a negative offset is to the left. The coordinates of the point are immediately displayed on the screen. The menu:



SOLVE COORDINATES

Station: 0+05.000

Offset: -3.000m

Northing: 2998.0961m

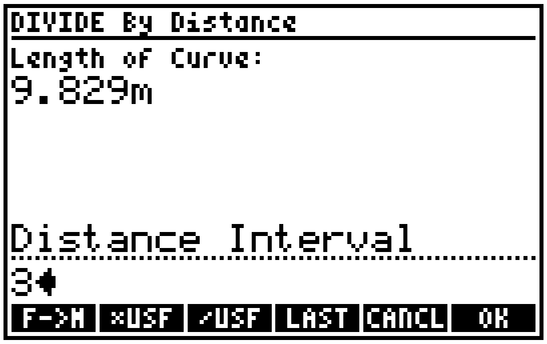
Easting: 2974.1820m

Enter Station to Solve

1. F4 **Sta?** – Select a position to jump to on the curve; BC, PI, EC, mid-point of curve, or the Radius Point. Sets the *Station* field to the selected station and sets the *Offset* field to zero. NOTE: THE PI STATION IS NOT ON THE CURVE.
2. F5 **STORE** – Stores the calculated coordinates as a point in the job database.

Distance Interval

The first input screen will display the length of the curve and accept a distance interval value. Points will be created every specified distance starting at the *Beginning of Curve Point* until the remainder to the *End of Curve Point* is less than or equal to the distance interval specified.



DIVIDE By Distance

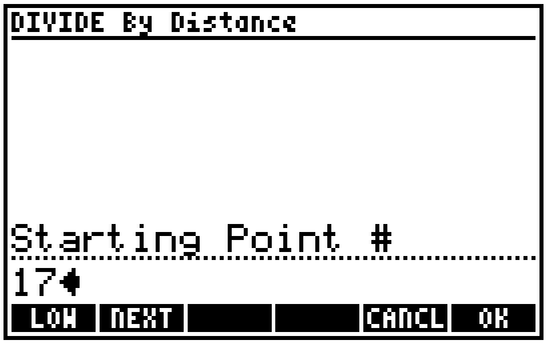
Length of Curve:

9.829m

Distance Interval

3◆

The second input screen prompts for a Starting Point Number to be used for storing the points that will be calculated. Unused point number will be assigned to the positions starting at the specified point number. Point coordinates are calculated in three dimensions between the *Beginning of Curve Point* and *End of Curve Point*.

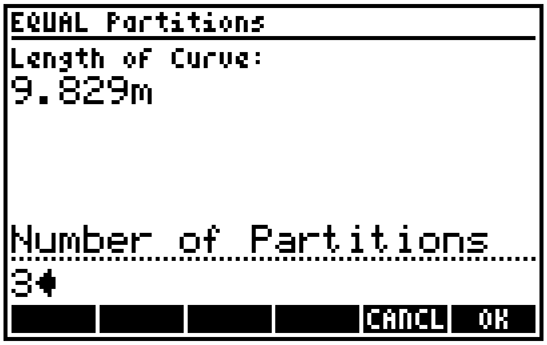


DIVIDE By Distance

Starting Point #

17◆

Equal Partitions


EQUAL Partitions

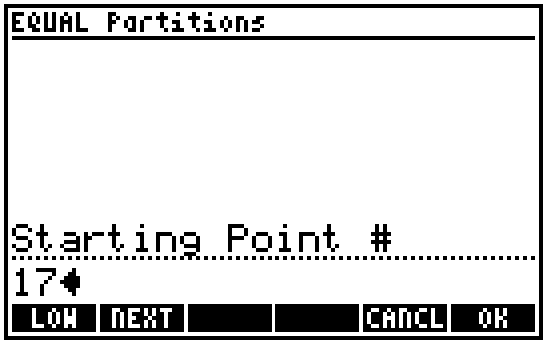
Length of Curve:

9.829m

Number of Partitions

3

The first input screen will display the length of the curve and accept the number of partitions to divide the curve into. Points will be created at the calculated interval so that the specified number of equal partitions will be created between the *Beginning of Curve Point* and *End of Curve Point*.


EQUAL Partitions

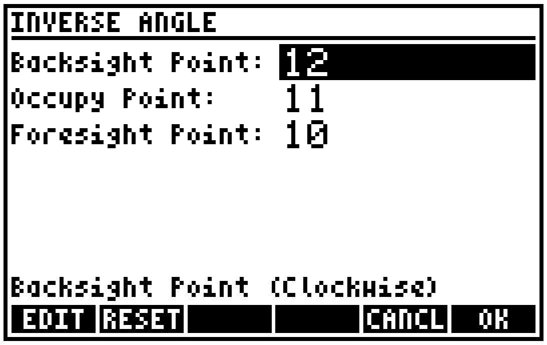
Starting Point #

17

The second input screen prompts for a Starting Point Number to be used for storing the points that will be calculated. Unused point number will be assigned to the positions starting at the specified point number. Point coordinates are calculated in three dimensions between the *Beginning of Curve Point* and *End of Curve Point*.

Inverse Angle

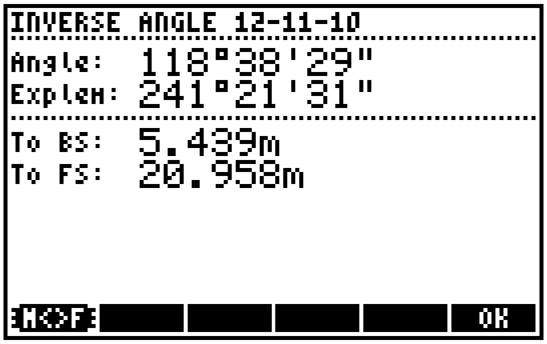
Enter the *Backsight Point*, the *Occupy Point* and the *Foresight Point* to calculate the included angle. The *Backsight Point* is always from clockwise resulting in an “angle right” calculation.


INVERSE ANGLE

Backsight Point: 12
Occupy Point: 11
Foresight Point: 10

Backsight Point (Clockwise)

The results screen displays the turned angle and its explementary angle as well as the distances to the backsight and foresight points from the occupy point. Use F1 **M<>F** to convert the distance results between metric and imperial units.

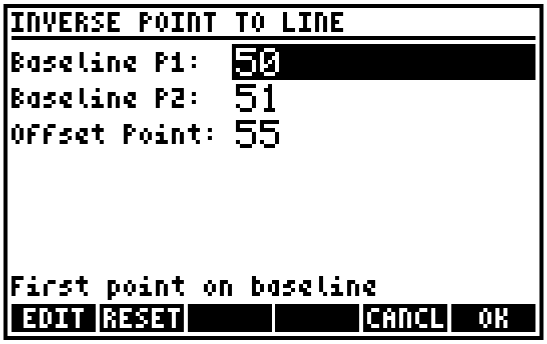

INVERSE ANGLE 12-11-10

Angle: 118°38'29"
Explem: 241°21'31"

To BS: 5.439m
To FS: 20.958m

Inverse Point to Line

Enter two points, *Baseline P1* and *Baseline P2*, to define a baseline and an *Offset Point* to calculate the perpendicular offset of the point to the line.

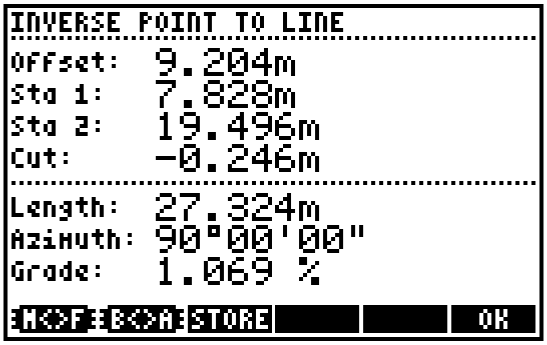

INVERSE POINT TO LINE

Baseline P1: 50
Baseline P2: 51
Offset Point: 55

First point on baseline

The results screen displays the offset from the baseline, Station 1 from *Baseline P1* to a point along the baseline that is perpendicular to the offset point, Station 2 to the same point from *Baseline P2*, the cut/fill to the baseline and the length, direction and the grade of the baseline. The menu:

1. F1 **M<>F** – Toggles metric/imperial.
2. F2 **B<>A** – Toggles bearings/azimuths.
3. F3 **STORE** – Stores the coordinates of the point along the baseline perpendicular to the offset point as a point in the job database.


INVERSE POINT TO LINE

Offset: 9.204m
Sta 1: 7.828m
Sta 2: 19.496m
Cut: -0.246m

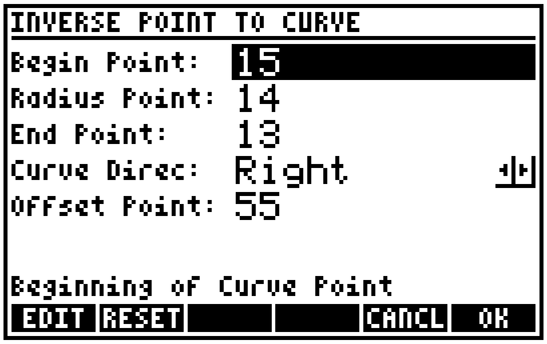
Length: 27.324m
Azimuth: 90°00'00"
Grade: 1.069 %

Inverse Point to Curve

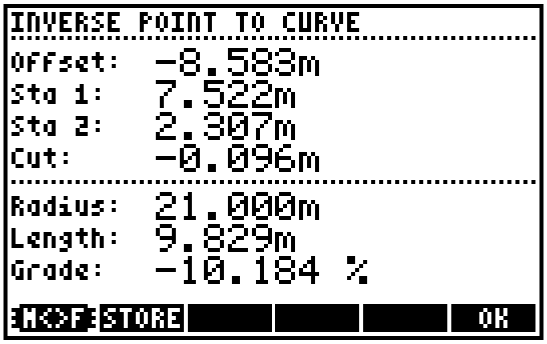
Enter a *Beginning of Curve Point*, a *Radius Point* and a *End of Curve Point*, choose a *Curve Direction*, and enter a *Offset Point* to calculate the perpendicular offset of the point to the curve.

The results screen displays the offset from the curve, Station 1 from the *Beginning of Curve* to a point along the curve that is perpendicular to the offset point, Station 2 to the same point from the *End of Curve*, the cut/fill to the curve, and the radius, length and grade of the curve. The menu:

1. F1 **M<>F** – Toggles metric/imperial.
2. F2 **STORE** – Stores the coordinates of the point along the curve perpendicular to the offset point as a point in the job database.

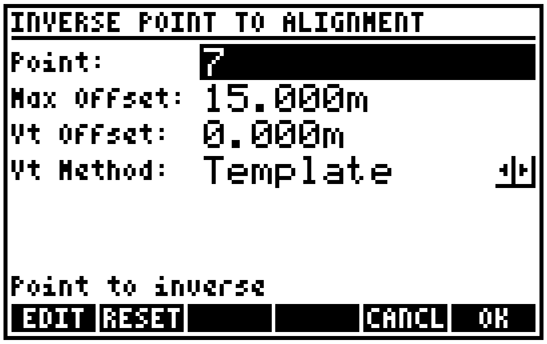


```
INVERSE POINT TO CURVE
Begin Point:  15
Radius Point: 14
End Point:    13
Curve Direc:  Right
Offset Point: 55

Beginning of Curve Point
EDIT RESET          CANCL OK
```



```
INVERSE POINT TO CURVE
Offset:  -8.583m
Sta 1:   7.522m
Sta 2:   2.307m
Cut:     -0.096m
Radius:  21.000m
Length:  9.829m
Grade:   -10.184 %
M<>F STORE               OK
```

Inverse Point to Alignment



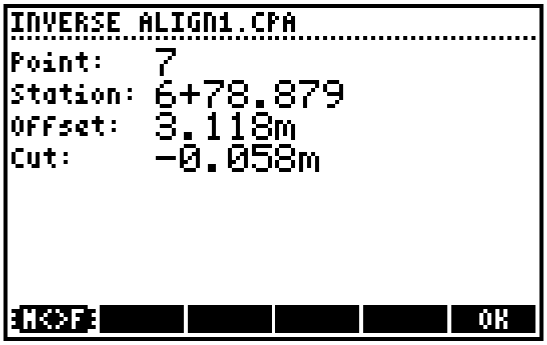
INVERSE POINT TO ALIGNMENT

Point: 7
Max Offset: 15.000m
Vt Offset: 0.000m
Vt Method: Template

Point to inverse

Choose an alignment from the list of available alignments to inverse a point to the alignment and determine the station, offset and cut/fill of the point in relation to the alignment design.

Enter a point number to inverse to the selected alignment, and specify a maximum offset from centerline. Optionally specify a vertical offset constant from the alignment design, and choose whether elevation values are calculated to the cross section template or to the centerline of the alignment.



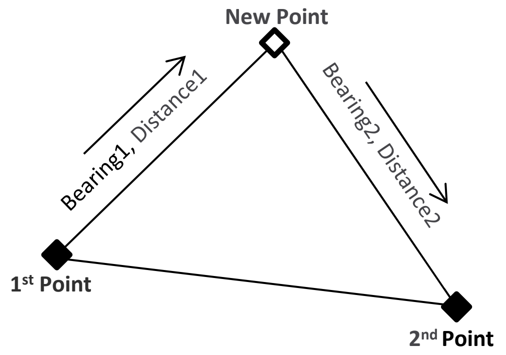
INVERSE ALIGN1.CPA

Point: 7
Station: 6+78.879
Offset: 3.118m
Cut: -0.058m

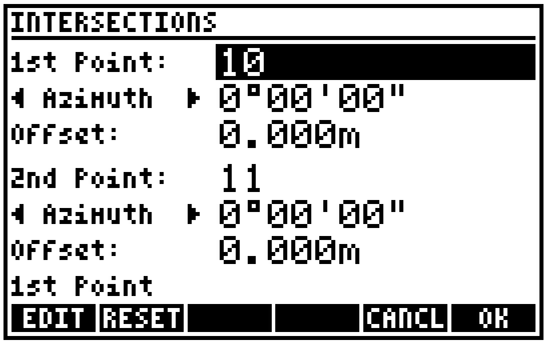
The maximum offset value is used to eliminate erroneous inverse results that may occur when alignments have significant geometry changes which could in turn provide offsets that are valid but not for the desired station.

The result includes the point number entered, the calculated station, offset and cut/fill. Use F1 M<>F to toggle the offset and cut/fill between metric/imperial.

5.3 Intersections



The **Intersections** program exclusively handles all types of intersections from a single input form. The words “Bearing” and “Azimuth” are interchangeable for naming the intersection types. In the diagram shown, the ‘1st Point’ and “2nd Point’ points are always known points and the ‘New Point’ point can be calculated when Bearing1 or Distance1 **and** Bearing2 or Distance2 is known.



```
INTERSECTIONS
1st Point:    10
◀ Azimuth  ▶ 0°00'00"
Offset:       0.000m
2nd Point:    11
◀ Azimuth  ▶ 0°00'00"
Offset:       0.000m
1st Point
```

‘1st Point’ and ‘2nd Point’ Fields

Both of these fields require an existing point number.

‘Azimuth’ or ‘Bearing’ / ’Distance’ Fields

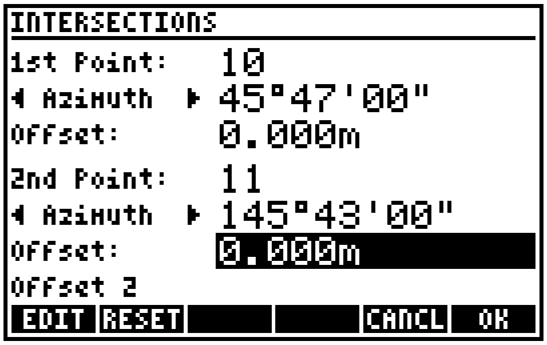
The labels for these fields change when the user toggles the input types for these fields with the ◀ and ▶ cursor keys. The label indicates how the input will be processed:

1. Azimuth/Bearing input is accepted using any of the standard directions input options.
2. Distance input is accepted using any of the standard distances input options.

‘Offset’ Fields

These fields are only visible when the corresponding Azimuth or Bearing is the selected known component. Enter offsets using any of the standard distances input options.

NOTE: The directions of Bearing1 and Bearing2 may be flipped 180° without consequence, however if you enter any offsets, these offsets will always be relative to your direction entered (left or right).



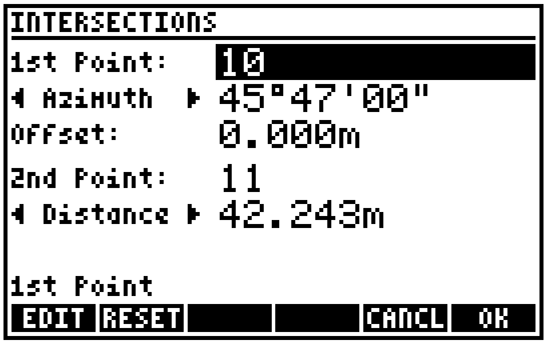
```
INTERSECTIONS
1st Point:    10
◀ Azimuth  ▶ 45°47'00"
Offset:       0.000m
2nd Point:    11
◀ Azimuth  ▶ 145°43'00"
Offset:       0.000m
Offset 2
```

Bearing-Bearing

A bearing-bearing intersection can be solved when Bearing1 and Bearing2 are known. Offsets may be entered for both azimuth/bearing inputs. Entering offsets makes it possible to create an offset intersection, for example.

Bearing-Distance

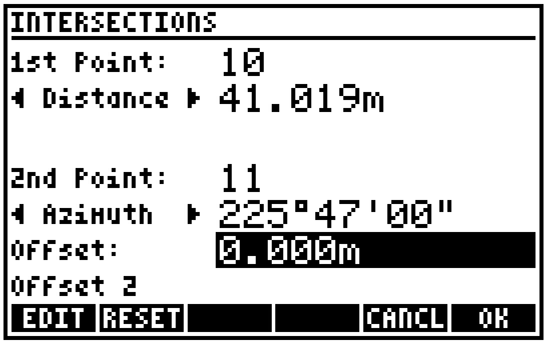
A bearing-distance intersection can be solved when Bearing1 and Distance2 are known. This type of intersection usually has two possible solutions. The user is prompted to choose which of the two solutions is desired by selecting one of the two possible Distance1 solutions. A “No Solution” error indicates that the intersection is not possible with the data provided.



```
INTERSECTIONS
1st Point:    10
◂ Azimuth  ▸  45°47'00"
Offset:       0.000m
2nd Point:    11
◂ Distance ▸  42.243m

1st Point
```

Distance-Bearing

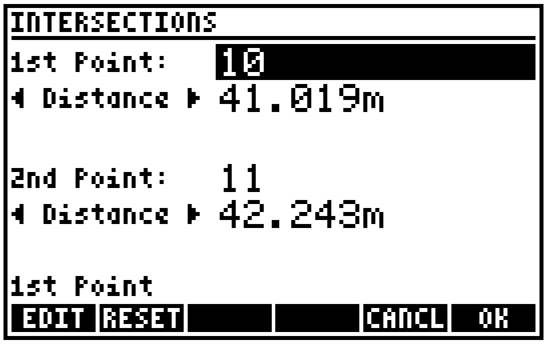
A distance-bearing intersection can be solved when Distance1 and Bearing2 are known. This type of intersection is similar to a bearing-distance intersection but started from the opposite direction, and usually also has two possible solutions. The user is prompted to choose which of the two solutions is desired by selecting one of the two possible Bearing1 solutions.



```
INTERSECTIONS
1st Point:    10
◂ Distance ▸  41.019m

2nd Point:    11
◂ Azimuth  ▸  225°47'00"
Offset:       0.000m
Offset 2
```

Distance-Distance

A distance-distance intersection can be solved when Distance1 and Distance2 are known. This type of intersection usually has two possible solutions. The user is prompted to choose which of the two solutions is desired by selecting one of the two possible Bearing1 solutions.



```
INTERSECTIONS
1st Point:    10
◂ Distance ▸  41.019m

2nd Point:    11
◂ Distance ▸  42.243m

1st Point
```

5.4 Area by Points

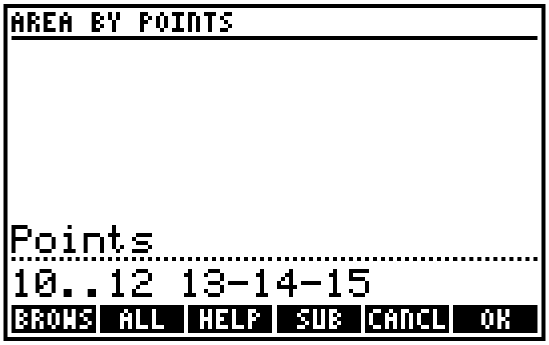
The **Area by Points** program calculates the area of a polygon when the points along the perimeter of the polygon are entered in sequence. The polygon can consist of straight segments and curves.

To enter straight segments:

1. Enter individual points separated by spaces. For example, *1 2 3 4 5 6*.
2. Enter a range of points in sequence and in numerical ascending order. For example, *1..6*.
3. Enter any combination of the above. For example, *1 3 9..15 18 20..29 33*.

To enter curves:

1. Curve 'Right' – Enter points separated by the '+' character so that the curve is defined by BC+CC+EC. For example, *1+2+3*.
2. Curve 'Left' – Enter points separated by the '-' character so that the curve is defined by BC-CC-EC. For example, *1-2-3*.
3. Compound curves and reverse curves – Enter each curve component separately so that each curve component is its own block of points, BC+CC+EC or BC-CC-EC, and each block is separated by a space. For example, *1+2+3 3-4-5* or *1+2+3 3+4+5*, etc.



AREA BY POINTS

Points

10..12 13-14-15

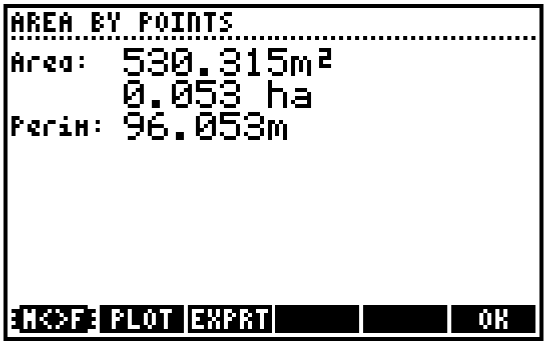
NOTE: AN ERROR OCCURS WHEN THE POINTS PROVIDED AS CURVE POINTS DO NOT ACTUALLY DEFINE A CURVE, I.E. THE RADIUS DIFFERS BY MORE THAN THE radius tolerance SETTING.

Any mix of straight segments and curves is accepted. For example, an area with straight segments and a curve could be *10 11 12 13-14-15* which could also be entered as *10..12 13-14-15*.

The menu on the input screen:

1. F1 **BROWS** - Opens the point browser to select/review points.
2. F2 **ALL** - Use all points in the current job, the points will be used in the sequence they were created. NOTE: BE CAREFUL WITH THIS OPTION, THIS OPTION COULD PROVIDE RESULTS YOU DO NOT EXPECT OR IT COULD TAKE SOME TIME TO CALCULATE WHEN SEVERAL HUNDRED POINTS EXIST.
3. F3 **HELP** - Displays a summary of how input is accepted.
4. F4 **SUB** - Subdivide a pre-determined area.

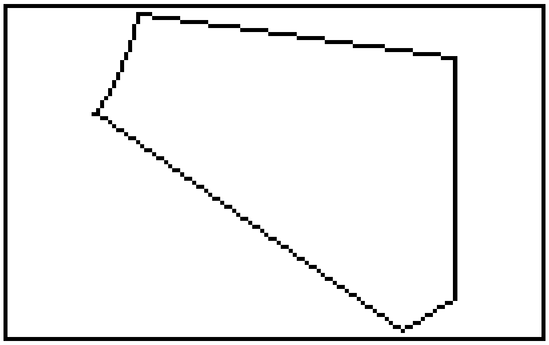
The area (square units and hectares/acres) and perimeter are displayed following a valid input. The program automatically determines the direction (clockwise or counter clockwise) the points were entered. The primary distance unit affects the results displayed.



```
AREA BY POINTS
Area:  530.315m²
       0.053 ha
Perim: 96.053m
```

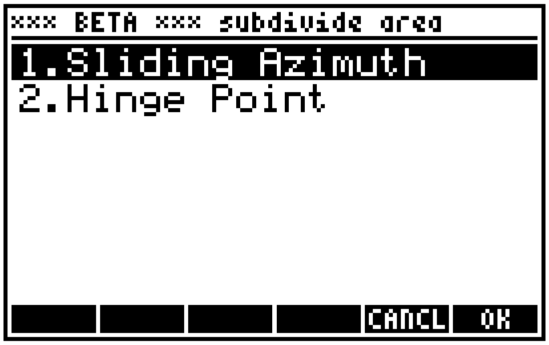
The menu on the results screen:

1. F1 **M<>F** - Toggles metric/imperial.
2. F2 **PLOT** - Draws the area polygon on the screen.
3. F3 **EXPRT** - Writes a DXF file of the polygon linework to save to the SD card COGOPLUS\ASCII directory or to the HOME directory on the calculator. DXF files can be opened with a CAD program.



Area Subdivisions

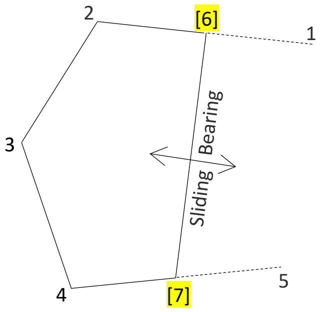
COGO+ Pro Version 3.57 introduced two BETA editions of area subdivision programs. Coordinates of the missing point(s) are calculated to complete the fixed-area polygon.



```
*** BETA *** subdivide area
1.Sliding Azimuth
2.Hinge Point
```

Sliding Bearing/Azimuth

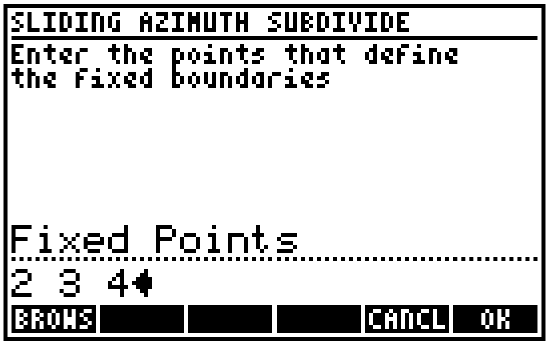
Calculate a specified area by sliding a line of fixed bearing/azimuth. For the diagram on the right, assume the known coordinates as listed and the fixed azimuth line to be 6°10'35" with a desired area of 100m². The program will calculate the coordinates for highlighted points 6 and 7 shown.



P,N,E Coordinates
1,100,100
2,101,90
3,96,86
4,89,89
5,90,99

First, enter the points that define the fixed boundaries, in this example those points are 2, 3 and 4.

Next, enter the directions from the first and from the last point that you entered for the fixed points. Use any of the standard directions input options.

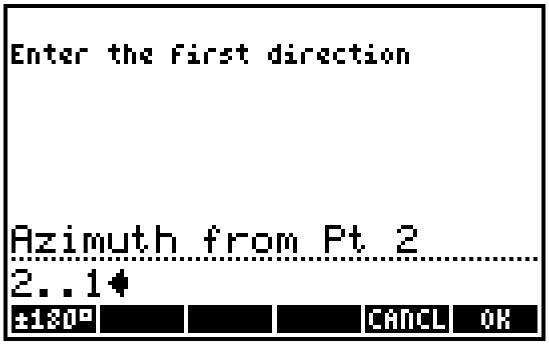


SLIDING AZIMUTH SUBDIVIDE

Enter the points that define the fixed boundaries

Fixed Points

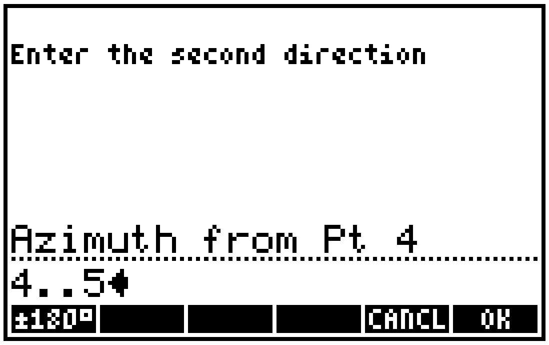
2 3 4◆



Enter the first direction

Azimuth from Pt 2

2..1◆

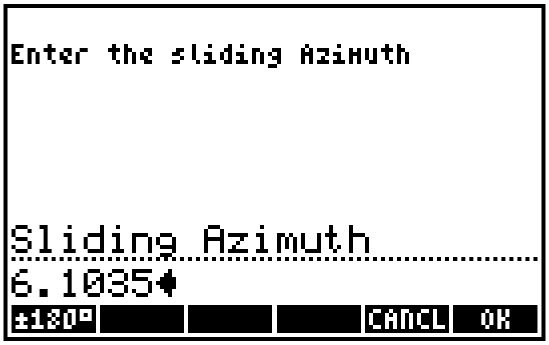


Enter the second direction

Azimuth from Pt 4

4..5◆

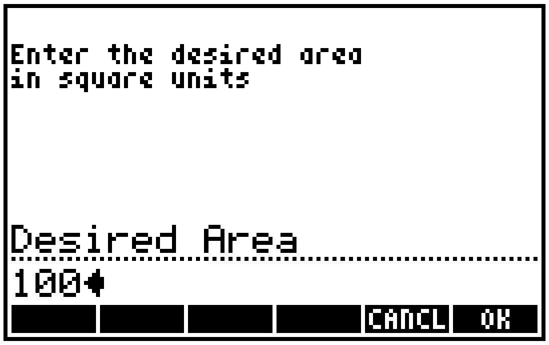
Next, enter the sliding azimuth/bearing followed by the desired area.



Enter the sliding Azimuth

Sliding Azimuth

6.1035◆

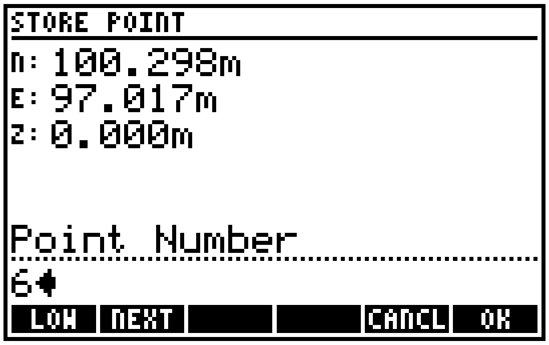


Enter the desired area in square units

Desired Area

100◆

The solution for each point is presented on the standard STORE POINT screen.

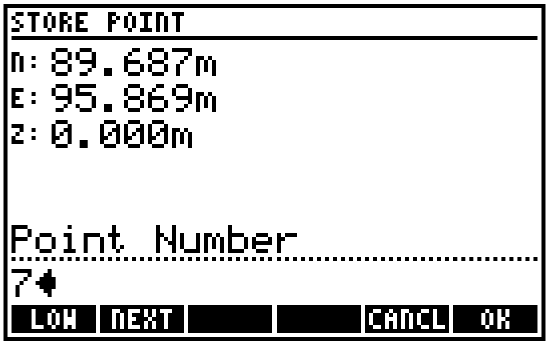


STORE POINT

N: 100.298m
E: 97.017m
Z: 0.000m

Point Number

6◆



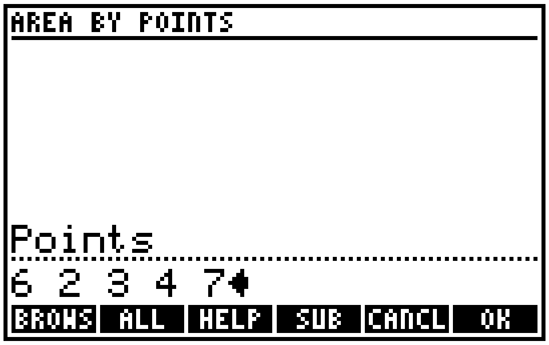
STORE POINT

N: 89.687m
E: 95.869m
Z: 0.000m

Point Number

7◆

When both points are stored in the job database a message appears on the screen "New Area String copied to Clipboard…". This allows the user to paste the clipboard contents into the Area by Points program to confirm the solution.

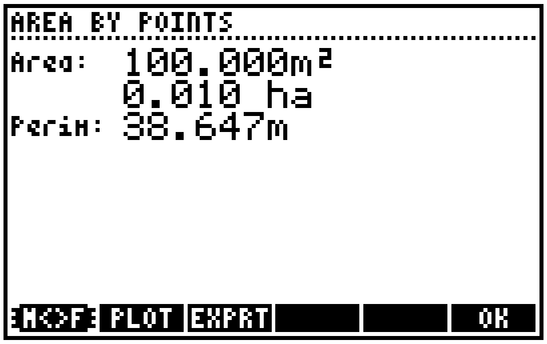

AREA BY POINTS

Points

6 2 3 4 7◆

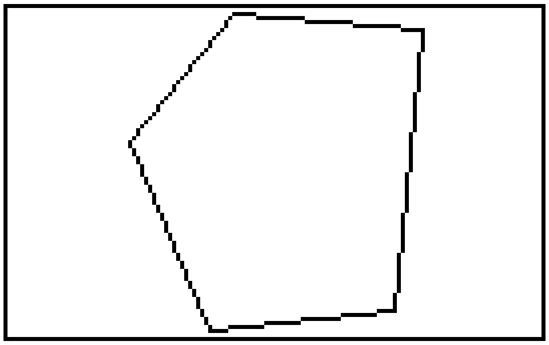
Use ↱ followed by PASTE to paste the clipboard contents into the Area by Points input screen.

The solution and plot should be as shown below.


AREA BY POINTS

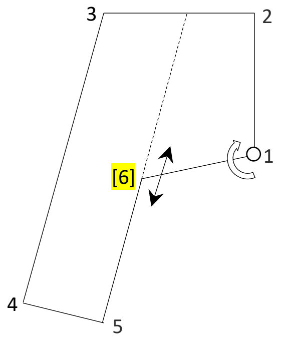
Area: 100.000m²
0.010 ha

Perim: 38.647m



Hinge Point

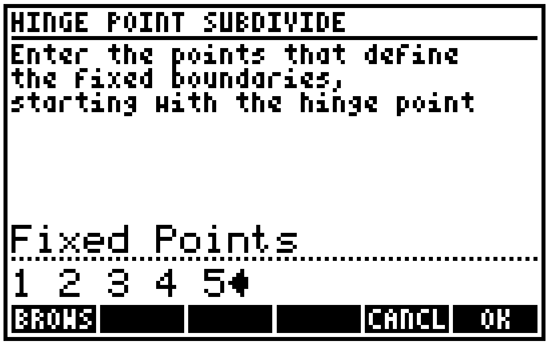
Calculate a specified area by swinging a line from a hinge point into another line of fixed bearing/azimuth. For the diagram on the right, assume the known coordinates as listed and the direction from Point 5 to be parallel to the line 4-3 with a desired area of 180m². The program will calculate the coordinates for highlighted point 6 shown.



P,N,E Coordinates

P	N	E
1	20	20
2	30	20
3	30	10
4	11	3
5	9	8

First, enter the points that define the fixed boundaries, starting with the hinge point. In this example those points are 1, 2, 3, 4 and 5.

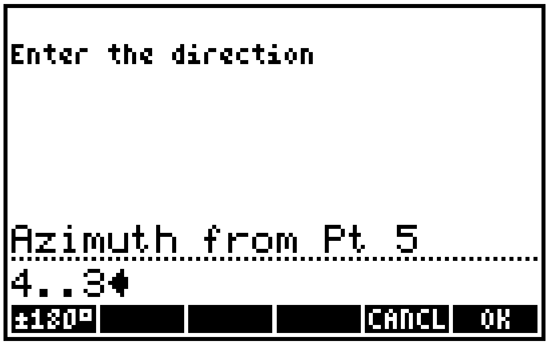

HINGE POINT SUBDIVIDE

Enter the points that define the fixed boundaries, starting with the hinge point

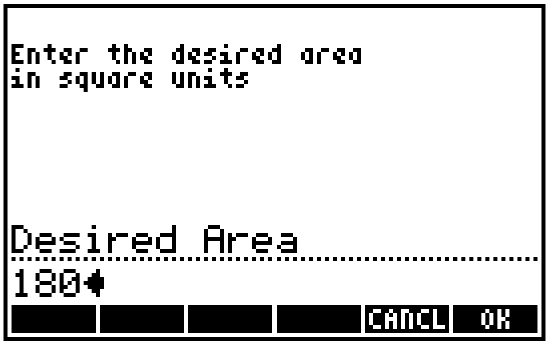
Fixed Points

1 2 3 4 5◆

Next, enter the direction from the last fixed point followed by the desired area. Use any of the standard directions input options.


Enter the direction

Azimuth from Pt 5
4..3

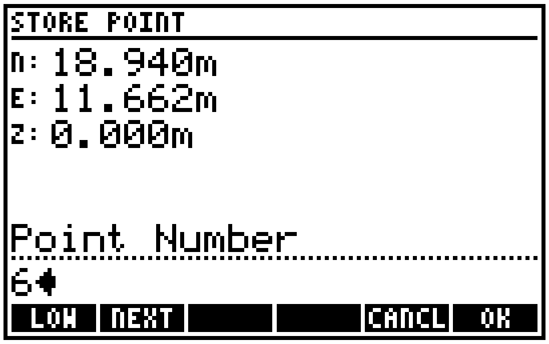

Enter the desired area
in square units

Desired Area
180

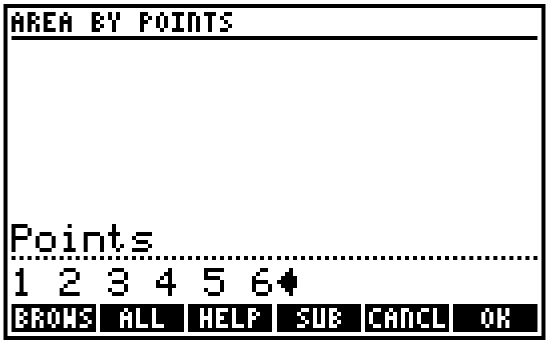
The solution for the point is presented on the standard STORE POINT screen.

When the point is stored in the job database a message appears on the screen "New Area String copied to Clipboard…". This allows the user to paste the clipboard contents into the Area by Points program to check to confirm the solution.

Use ↱ followed by PASTE to paste the clipboard contents into the Area by Points input screen.

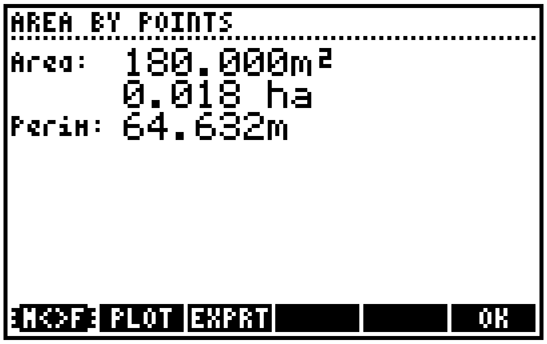

STORE POINT
N: 18.940m
E: 11.662m
Z: 0.000m

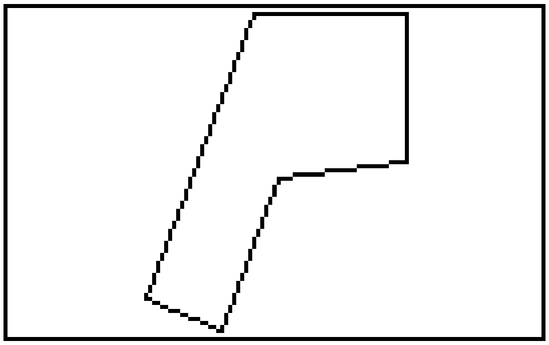
Point Number
6


AREA BY POINTS

Points
1 2 3 4 5 6

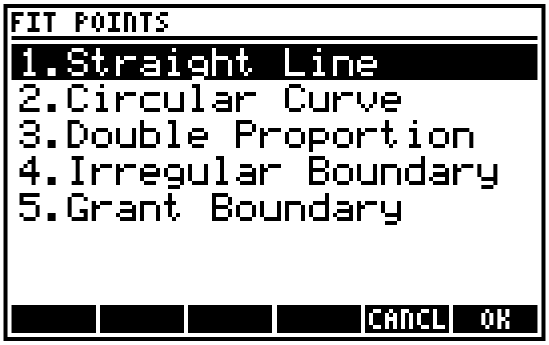
The solution and plot should be as shown below.


AREA BY POINTS
Area: 180.000m²
0.018 ha
Perim: 64.632m



5.5 Fit Points

The **Fit Points** program consists of:

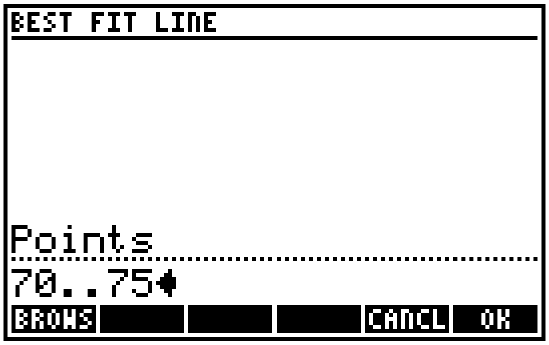


1. Best fit points to a straight line, linear regression.
2. Best fit points to a curve, similar to linear regression but fits points to a circular curve.
3. Solve the position of a lost corner using the **Double Proportionate Method**.
4. Solve the position of a lost corner using the **Irregular Boundary Adjustment Method**.
5. Solve the positions of lost angle points using the **Grant Boundary Adjustment Method**.

Best Fit Line

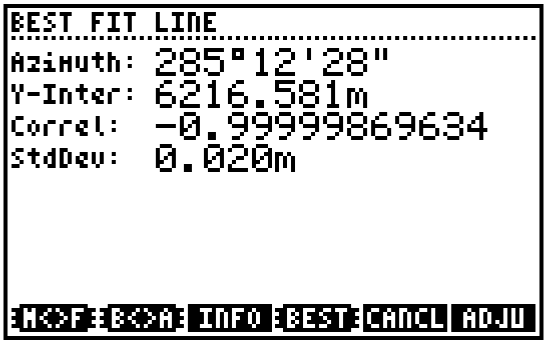
Enter a series of points to compute the least squares straight line that best fits the points. Enter point numbers using any of the point numbers input options.

An un-weighted linear regression method that minimizes BOTH X and Y residuals simultaneously is used to calculate the line.



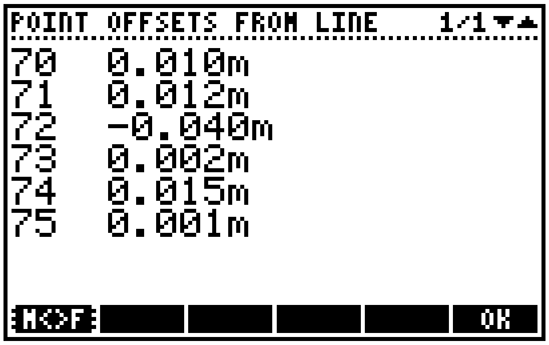
Points
70..75

The direction and Y-Intercept of the line, the correlation coefficient (a value between -1 and 1), and the point offsets standard deviation are computed and displayed. The menu:



```
Azimuth: 285°12'28"
Y-Inter: 6216.581m
Correl:  -0.9999869634
StdDev:  0.020m
```

1. F1 **M<>F** – Toggles metric/imperial.
2. F2 **B<>A** – Toggles bearings/azimuths.
3. F3 **INFO** – Displays the perpendicular offsets to the line for each point used in the computation.



```
POINT OFFSETS FROM LINE    1/1
70   0.010m
71   0.012m
72   -0.040m
73   0.002m
74   0.015m
75   0.001m
```

4. F4 **BEST** or **FIX** – The menu label indicates whether the calculation represents a best-fit line or a line with a fixed direction. It is possible to enter a direction for the line that

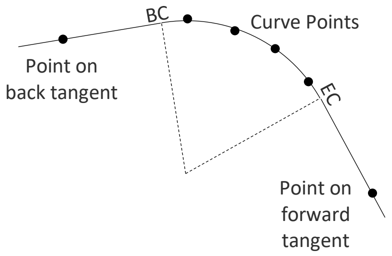
differs from the best-fit direction and recalculate with the fixed parameter.

5. F5 **CANCL** – Return to the points input screen without adjusting the points.
6. F6 **ADJU** – Proceed with adjusting the points.

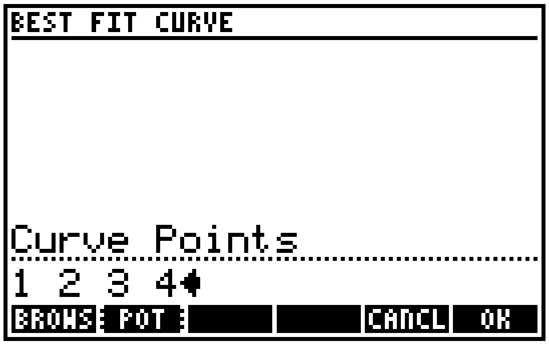
Choosing to apply the solution will either shift the points to fall on the line, which overwrites the existing points, or calculate new points re-numbered with an additive point number. The adjusted points setting controls the behaviour of overwrite/re-number.

Points are always shifted perpendicular to the best-fit line to minimize the shift.

Best Fit Curve



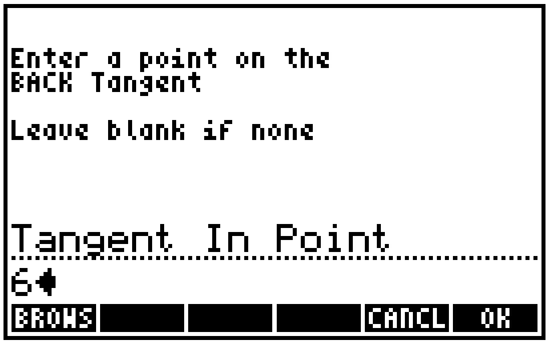
The main purpose of this program is to calculate the radius and the coordinates of the radius point of the least squares circle that best fits a series of points. When providing points along one or both tangents, the program will also solve the BC and/or EC point coordinates.


BEST FIT CURVE

Curve Points
1 2 3 4◀

In the main input form, enter a series of points along the curve to compute the circle that best fits the points. Enter point numbers using any of the point numbers input options. The menu:

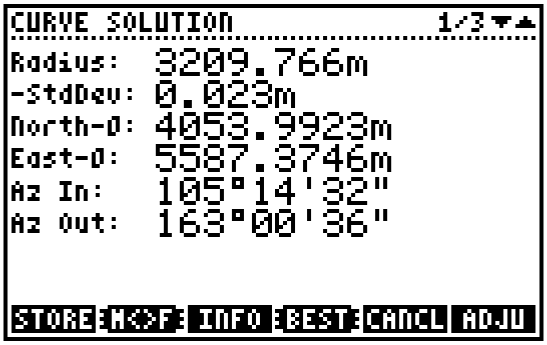
1. F1 **BROWS** - Opens the point browser to select/review points.
2. F2 **POT** - Toggle "Points On Tangents" input option. When enabled, the program will prompt for points on the back and forward tangent.


Enter a point on the
BACK Tangent

Leave blank if none

Tangent In Point
6◀

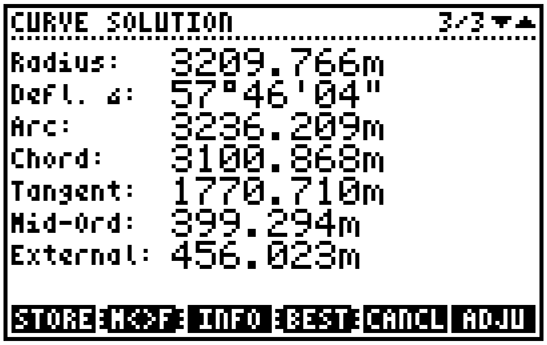
When the POT toggle is set, the program will also ask the user to choose the curve direction Right/Left.



```
CURVE SOLUTION          1/3
Radius:  3209.766m
-StdDev: 0.023m
North-O: 4053.9923m
East-O:  5587.3746m
Az In:   105°14'32"
Az Out:  163°00'36"
```

The first page of the curve solution displays the computed radius, the radius standard deviation, the coordinates of the radius point, and if applicable, the azimuths/bearings of the back and forward tangents.

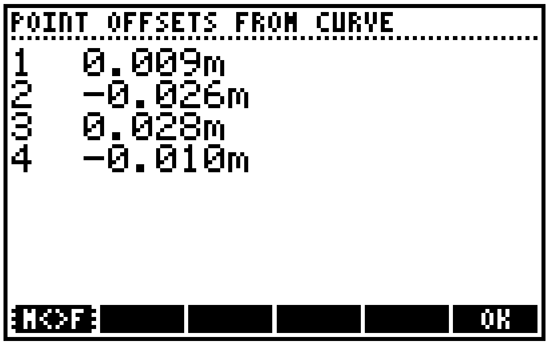
The second page of the curve solution displays the coordinates of the BC and EC points if possible to solve with the input provided. THIS PAGE IS NOT APPLICABLE WHEN THE POT TOGGLE IS NOT SET.



```
CURVE SOLUTION          3/3
Radius:   3209.766m
Defl. ∆:  57°46'04"
Arc:      3236.209m
Chord:    3100.868m
Tangent:  1770.710m
Mid-Ord:  399.294m
External: 456.023m
```

The third page of the curve solution displays the curve elements: radius, deflection angle, arc, chord, tangent, mid-ordinate and external values. THIS PAGE IS ONLY DISPLAYED WHEN BOTH POT INPUTS ARE PROVIDED. The menu:

1. F1 **STORE** – Stores the computed radius point coordinates as a point in the job database, or when possible to store BC and/or EC points, will ask the user to choose which point to store.
2. F2 **M<>F** – Toggles metric/imperial.
3. F3 **INFO** – Displays the radial offsets to the curve for each point used in the computation.



```
POINT OFFSETS FROM CURVE
1   0.009m
2   -0.026m
3   0.028m
4   -0.010m
```

4. F4 **BEST** or **FIX** – The menu label indicates whether the calculation represents a best-fit radius or a curve with a fixed radius. It is possible to enter a radius for the curve that differs from the best-fit radius and recalculate a new least squares circle with the fixed parameter.
5. F5 **CANCL** – Return to the points input screen without adjusting the points.
6. F6 **ADJU** – Proceed with adjusting the points.

Choosing to apply the solution will either shift the points to fall on the curve, which overwrites the existing points, or calculate new points re-numbered with an additive point number. The adjusted points setting controls the behaviour of overwrite/re-number.

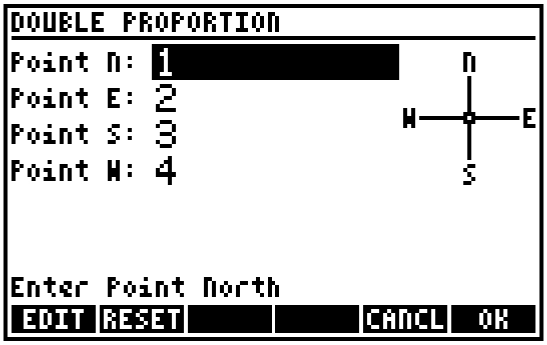
Points are always shifted radially to the best-fit curve to minimize the shift.

Double Proportion

“The term ‘double proportionate measurement’ is applied to a new measurement made between four known corners, two each on intersecting meridional and latitudinal lines, for the purpose of relating the cardinal equivalent intersection to both.” (Page 166, BLM Manual of Surveying Instructions 2009).

The **Double Proportion** program solves this type of problem by accepting point numbers for the four known corners, and the record measurements to the lost corner.

The first input form requires existing point numbers of the corners to the north, east, south and west of the lost corner.

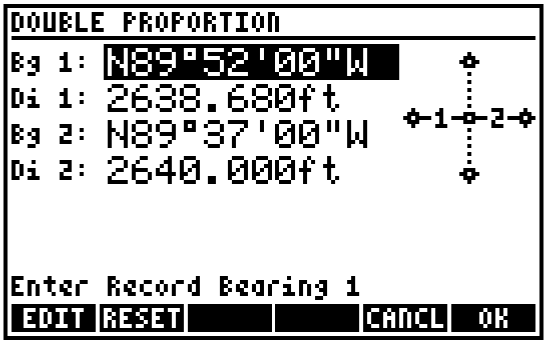


DOUBLE PROPORTION

Point N: 1
Point E: 2
Point S: 3
Point W: 4

Enter Point North

The second input form requires the **record** bearings and distances to the known corners west and east of the lost corner. When entering or editing a distance the menu offers the following functions:



DOUBLE PROPORTION

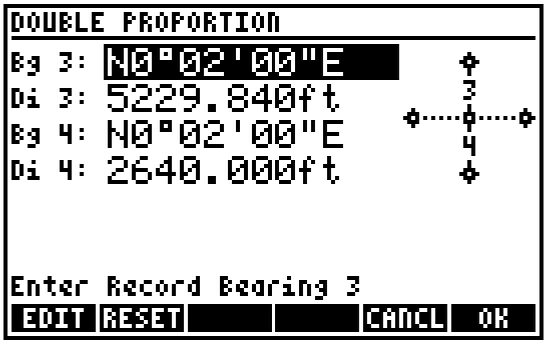
Bg 1: N89°52'00"W
Di 1: 2638.680ft
Bg 2: N89°37'00"W
Di 2: 2640.000ft

Enter Record Bearing 1

1. F1 **F->M** or **M->F** – Converts the input between metric and imperial units. The appearance of this softkey varies depending on your primary distance unit setting.
2. F2 **xUSF** – Multiplies the input by the user defined scale factor.
3. F3 **/USF** – Divides the input by the user defined scale factor.

The third input form requires the **record** bearings and distances to the known corners north and south of the lost corner.

Following input, the coordinates of the lost corner are calculated and the point may be stored to the job database.

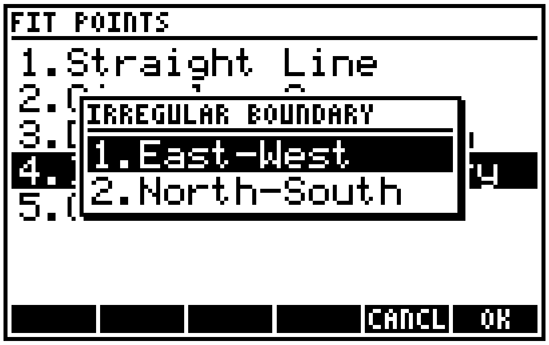


DOUBLE PROPORTION

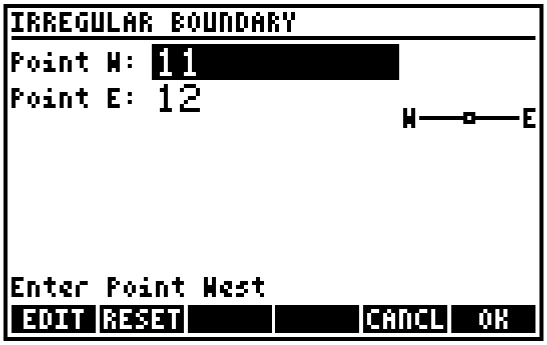
Bg 3: N0°02'00"E
Di 3: 5229.840ft
Bg 4: N0°02'00"E
Di 4: 2640.000ft

Enter Record Bearing 3

Irregular Boundary Adjustment



Irregular boundaries are the result of boundaries surveyed from opposite directions, or piecemeal surveys where resulting boundaries are not straight lines. “In order to restore one or more lost corners or angle points on such irregular exterior, a retracement between the nearest known corners is made on the record courses and distances to ascertain the direction and length of the closing distance. A position is calculated for each lost corner or angle point at the record position.” (Page 174, BLM Manual of Surveying Instructions 2009) A combination of single proportion and compass rule is then used to re-establish the lost corner. The direction (East-West or North-South) determines how the adjustment is performed.


IRREGULAR BOUNDARY

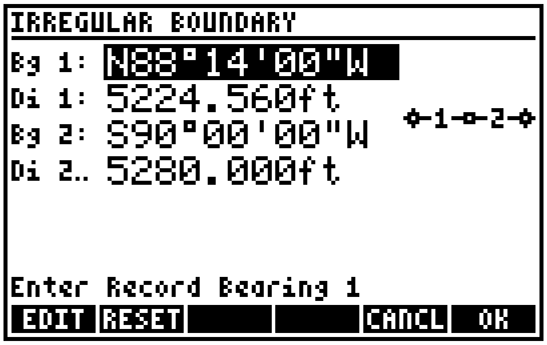
Point W: 11

Point E: 12

Enter Point West

The **Irregular Boundary** program prompts for the direction of the line, and then the first input form requires the point numbers for the surveyed points (W + E, or N + S).

The second input form requires the **record** dimensions to the corners on either side of the lost corner. When entering or editing a distance the menu offers the following functions:


IRREGULAR BOUNDARY

Bg 1: N88°14'00"W

Di 1: 5224.560ft

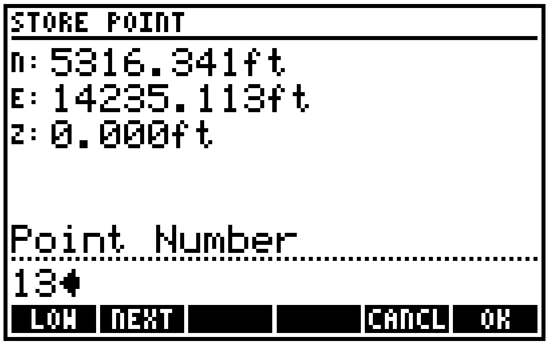
Bg 2: S90°00'00"W

Di 2.. 5280.000ft

Enter Record Bearing 1

1. F1 **F->M** or **N->F** – Converts the input between metric and imperial units. The appearance of this softkey varies depending on your primary distance unit setting.
2. F2 **xUSF** – Multiplies the input by the user defined scale factor.
3. F3 **/USF** – Divides the input by the user defined scale factor.

Following input, the coordinates of the lost corner are calculated and the point may be stored to the job database.


STORE POINT

N: 5316.341ft

E: 14235.113ft

Z: 0.000ft

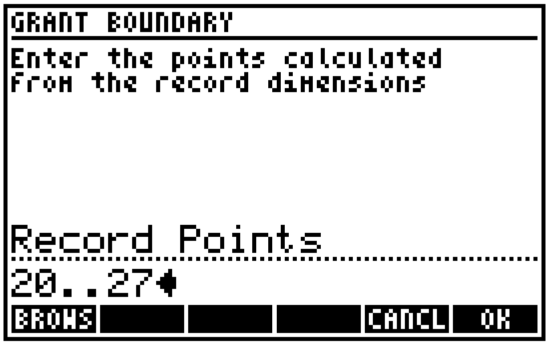
Point Number

13

Grant Boundary Adjustment

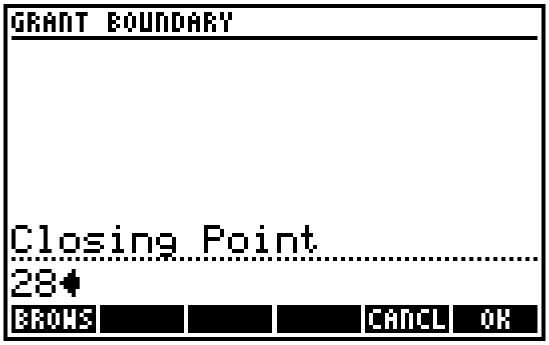
"In many of the States there are irregular grant and reservation boundaries that were established prior to the public rectangular surveys. In these cases, the township and section lines are regarded as the closing lines. The grant boundary field notes may call for natural objects, but these are often supplemented by metes-and-bounds descriptions. The natural calls are ordinarily given precedence then the existent angle points of the metes-and-bounds survey. The lost angle points are then restored by uniformly orienting the record courses to the left or right and adjusting the lengths of the lines on a constant ration." (Page 176, BLM Manual of Surveying Instructions 2009)

The **Grant Boundary** program requires a series of points that define the grant boundary, starting with a known (found) point followed by calculated points using **record** dimensions. The first input screen asks for these points. The second input screen asks for a *Closing Point*, which is a found original point representing the last calculated point that was entered. The spatial difference between the closing point and the last calculated point determines the adjustment parameters.


GRANT BOUNDARY

Enter the points calculated
from the record dimensions

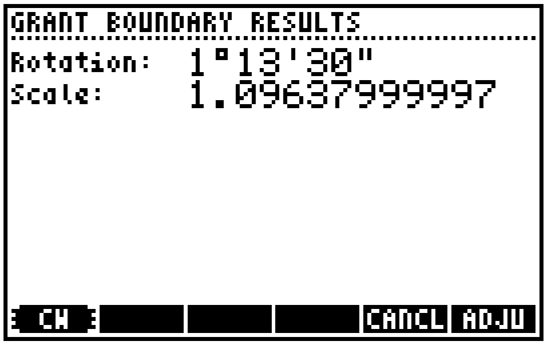
Record Points
20..27◆


GRANT BOUNDARY

Closing Point
28◆

The calculated rotation and scale are displayed on the results screen. The menu:

1. F1 **CW** or **CCW** – Toggles the rotation displayed as being clockwise or counter-clockwise.
2. F5 **CANCL** – Cancel the adjustment and return to the input screen.
3. F6 **ADJU** - Adjusts the record points. The adjustment will either update the existing point coordinates, which overwrites the existing points, or calculate new points re-numbered with an additive point number. The adjusted points setting controls the behaviour of overwrite/re-number.


GRANT BOUNDARY RESULTS

Rotation: 1°13'30"
Scale: 1.09637999997