HP-35S Calculator Program   Closure   1

Traverse Closure with Area Calculation and Co-ordinates

Programmer: Dr. Bill Hazelton

Date: October, 2007.

Line	Instruction	Display	User Instructions
A001	LBL A		Press XEQ A ENTER
A002	CLSTK		
A003	SF 10		
A004	USE COORDS	USE COORDS	(Key in using EQN RCL U, RCL S, etc.)
A005	PSE		
A006	INPUT C	C?	
A007	RCL C		
A008	x = 0?		
A009	GTO A021		
A010	ENTER N0	ENTER N0	(Key in using EQN RCL E, RCL N, etc.)
A011	PSE		
A012	INPUT N	N?	
A013	ENTER E0	ENTER E0	(Key in using EQN RCL E, RCL N, etc.)
A014	PSE		
A015	INPUT E	E?	
A016	RCL E		
A017	0 **i** 1		(Key in as 0, then **i**, then 1, press ENTER.)
A018	×		
A019	RCL+ N		
A020	STO F		
A021	360		
A022	STO B		
A023	CLSTK		
A024	STO P		
A025	STO Q		
A026	STO M		
A027	XEQ V001		
A028	STO R		
A029	STO+ P		
A030	ARG		
A031	RCL P		
A032	ARG		
A033	–		
A034	SIN		
A035	STO S		
A036	RCL R		
A037	ABS		
A038	STO× S		
A039	RCL P		

Traverse Closure with Area Calculation and Co-ordinates

A040	ABS		
A041	STO× S		
A042	RCL S		
A043	2		
A044	÷		
A045	STO+ Q		
A046	1		
A047	STO+ M		
A048	RCL C		
A049	x = 0?		
A050	GTO A067		
A051	RCL Q		
A052	ABS		
A053	RCL P		
A054	RCL+ F		
A055	RCL M		
A056	*x* **i** *y*		(Key in as ↰ DISPLAY, then select 9*xiy*)
A057	STOP		
A058	RCL P		
A059	ARG		
A060	x < 0?		
A061	RCL+ B		
A062	→HMS		
A063	RCL P		
A064	ABS		
A065	STOP		
A066	GTO A027		
A067	RCL Q		
A068	ABS		
A069	RCL P		
A070	ARG		
A071	x < 0?		
A072	RCL+ B		
A073	→HMS		
A074	RCL P		
A075	ABS		
A076	RCL M		
A077	STOP		
A078	GTO A027		

Notes

(1) Set the calculator into DEGREES mode (press MODE 1) before starting.

(2) This is a general traverse closure program that computes vector and area to each point around the traverse, together with co-ordinates, if desired. It also computes the misclosure of a closed traverse.

Traverse Closure with Area Calculation and Co-ordinates

(3) This program uses the V program as a sub-routine for data entry, so Program V must be in the calculator as program V (or the XEQ V001 at line A027 changed to reflect the changed label. The V program allows entry of azimuths in D.MMSS format (HP notation) and distances, while converting them to the internal complex number format. Program V is one of the HP-35s Utilities programs (Utility 3).

(4) The program allows the user to choose if the co-ordinates of each point are to be calculated. The user is prompted with USE COORDS briefly, followed by the C? prompt. If co-ordinates are desired, key in 1, if not, key in 0, then press R/S to continue.

(5) After each side (azimuth and distance) has been entered, the calculator produces the following output.

A. If the user has selected to use co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in the display in line 2, and the co-ordinates of the point in line 1.

Stack Register	Contents
T	
Z	Area of the traverse thus far
Y	Co-ordinate vector of current point (as a complex number)
X	Number of sides entered

The complex numbers will be displayed in rectangular form, with the Northing first, followed by the letter i, then the Easting. All this will be displayed as a single line of data.

The user can scroll through the stack, using the R↓ key, and can perform any other operation of interest to the data on the stack. This information is stored in memory registers for use later in the program, so the stack may be changed and worked with as needed.

To show the co-ordinates and the vector to the current point in polar form, change the display (use ↰ DISPLAY *10rθα*). The complex numbers will be displayed as r θ a, where r is the distance and a is the angle in decimal degrees.

When the user presses R/S, the calculator takes the vector to the current forward point from the starting point, and converts it into the distance (which is placed in line 2, the X register) and the azimuth in degrees, minutes and seconds (HP notation) in line 1 of the display (the Y register).

When the user presses R/S again, the calculator prompts for the azimuth of the next side to be entered. The azimuth should be entered in HP notation (DDD.MMSSss).

B. If the user has selected not to show co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in line 2 of the display (the X register) and the distance of the misclosure or the line connecting the starting point to the current point, in line 1 of the display (Y register).

Stack Register	Contents
T	Area of the traverse thus far
Z	Azimuth of the line from the start to current point (in HP notation)
Y	Distance of the line from the start to the current point
X	Number of sides entered

By pressing the R↓ key, the user can see the azimuth in register Z of the stack. Pressing R↓ again will show the area to the current point.

If the rectangular components of the misclosure are needed, press RCL P and make sure the display is in *x i y* mode (change it using ↰ DISPLAY 9*xiy*). The misclosure will be displayed as ΔN **i** ΔE.

When the user presses R/S again, the calculator prompts for the azimuth of the next side.

(6) Azimuths are entered and displayed by themselves in HP notation, i.e., DDD.MMSSss. If the azimuth is part of a complex number in r θ a form, the azimuth (on the right, after the θ) will be in decimal degrees.

(7) This program forms the basis of the two missing distances (2MD) program. Enter the known sides using this program to begin the 2MD computation process.

(8) In order to display the prompts, this program sets Flag 10. However, the program never ends, because it is up to the user to decide when to stop and move control elsewhere. So the program never clears Flag 10. If you require Flag 10 to be clear, in order to process equations, you must do this manually.

Theory

The traverse closure programs works by converting the entered azimuths (in DDD.MMSS, or HP, notation) and distances into complex numbers (which act as 2-D vectors), which are then added to compute the location of points around the traverse. The area is computed by triangles developed by each new side of the traverse and the line from the starting point to the current forward point, and is updated with each new side. So the area is that of the polygon formed by the traverse entered thus far and the line from the start to the current point. This allows areas to be incremented for lot splitting calculations.

The azimuth and distance of the line from the start to the current point is also placed on the stack after each line. This allows a connecting line to be computed easily between two points. The final azimuth and distance is the traverse misclosure and the area is that of the traverse.

If the user chooses, the co-ordinates of the starting point may be entered, and if this choice is made, the calculator displays the co-ordinates of each point, in addition to the other information. The co-ordinates are displayed as a complex number, in the form Northing **i** Easting.

Traverse Closure with Area Calculation and Co-ordinates

An arbitrary azimuth is satisfactory. Plane surveying assumptions apply. The program uses no error checking on entered data.

Sample Computation

Bearing	Distance
6° 53' 10"	72.00
112° 37' 20"	102.23
185° 39' 50"	29.04
181° 30' 00"	27.88
283° 54' 30"	102.38

Final Results

DE	=	0.023
DN	=	–0.002
Misclosure Length	=	0.023
Misclosure Bearing	=	95° 24' 15"
Area	=	6,378.4660

Stepping through the Calculation

A. Without Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS, the C?

Key in 0, then press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	72.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

0.0000	(area thus far)
6.5310	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	107.9004	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	72.3939	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	100.1841	(distance from start)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	88.0808	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	102.4027	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.6396	(area thus far)
	103.5423	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Traverse Closure with Area Calculation and Co-ordinates

Display shows:	0.0229	(distance from start, also linear misclosure)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.4660	(area thus far)
	95.2415	(azimuth from start, also azimuth of misclosure)

Pressing RCL P brings the misclosure vector to the X register as a complex number. It will be displayed in rectangular form as –0.0022 *i* 0.0228, in which case the misclosure component in the north-south direction is –0.0022, while that in the east-west direction is 0.0228. If the calculator is in rθα display mode, the display will show 0.0229 θ 95.4041. Convert this by using ↰ DISPLAY 9*xiy*.

B. Using Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS briefly, then C?

Key in 1, then press R/S.

Calculator prompts ENTER N0 briefly, then N?

Key in 1000.000, press R/S.

Calculator prompts ENTER E0 briefly, then E?

Key in 500.000, press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	1,071.4806 *i* 508.6325	(Co-ordinates of current forward point)
	1.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	0.0000	(area thus far)
	1,071.4806 *i* 508.6325	(Co-ordinates of current forward point)

Press R/S.

Display shows:	6.5310	(Azimuth for current forward point from starting point)
	72.0000	(Distance from starting point to current forward point)

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Traverse Closure with Area Calculation and Co-ordinates

Display shows:	1,032.1575 *i* 602.9971	(Co-ordinates of current forward point)
	2.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	3,542.3468	(area thus far)
	1,032.1575 *i* 602.9971	(Co-ordinates of current forward point)

Press R/S.

Display shows:	72.3939	(Azimuth for current forward point from starting point)
	107.9004	(Distance from starting point to current forward point)

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	1,003.2593 *i* 600.1310	(Co-ordinates of current forward point)
	3.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	4,984.4807	(area thus far)
	1,003.2593 *i* 600.1310	(Co-ordinates of current forward point)

Press R/S.

Display shows:	88.0808	(Azimuth for current forward point from starting point)
	100.1841	(Distance from starting point to current forward point)

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	975.3888 *i* 599.4012	(Co-ordinates of current forward point)
	4.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.6396	(area thus far)
	975.3888 *i* 599.4012	(Co-ordinates of current forward point)

Press R/S.

Display shows:	103.5423	(Azimuth for current forward point from starting point)
	102.4027	(Distance from starting point to current forward point)

Press R/S.

Traverse Closure with Area Calculation and Co-ordinates

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	999.9978 *i* 500.0228	(Co-ordinates of current forward point)
	5.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.4660	(area thus far)
	999.9978 *i* 500.0228	(Co-ordinates of current forward point)

Press R/S.

Display shows: 95.2415 (Azimuth for current forward point from starting point)
0.0229 (Distance from starting point to current forward point)

Press RCL P to bring the misclosure vector into the X register. If in rectangular mode, it will be displayed as –0.0022 *i* 0.0228, in which case the misclosure component in the north-south direction is –0.0022, while that in the east-west direction is 0.0228. If the calculator is in rθα display mode, the display will show 0.0229 θ 95.4041. Convert this by using ↰ DISPLAY 9*xiy*.

Storage Registers Used

A Used by the V program for the entered azimuth.

B Stores 360 for azimuth correction.

C Test for displaying co-ordinates: 1 = YES; 0 = NO.

D Used by the V program for the entered distance.

E Easting co-ordinates of the starting point.

F Co-ordinates of starting point, as a complex number.

I Used by the V program for indirect addressing.

M The number of sides entered.

N The Northing co-ordinate of the starting point.

P Current position of forward point, as a complex number.

Q Current area.

R Last side entered, as a complex number.

S Temporary storage for area calculation.

Statistical Registers: not used.

Other registers: 10 and 11 used by the V program.

Traverse Closure with Area Calculation and Co-ordinates

Labels Used

Label **A** Length = 268 Checksum = CED9

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

The program labeled V, which takes an azimuth in degrees, minutes and seconds (in HP notation), and a distance, and converts them to a complex number for processing in the calculator. This routines uses storage location A and D, but copies out and replaces the contents of these storage locations in order to preserve them. It also uses storage register I for indirect addressing, which uses registers 10, 11 and 12.

Label **V** Length = 128 Checksum = 39FE

Traverse Closure with Area Calculation and Co-ordinates

Programmer: Dr. Bill Hazelton

Date: October, 2007.

Line	Instruction	Display	User Instructions
A001	LBL A		Press XEQ A ENTER
A002	CLSTK		
A003	SF 10		
A004	USE COORDS	USE COORDS	(Key in using EQN RCL U, RCL S, etc.)
A005	PSE		
A006	INPUT C	C?	
A007	RCL C		
A008	x = 0?		
A009	GTO A021		
A010	ENTER N0	ENTER N0	(Key in using EQN RCL E, RCL N, etc.)
A011	PSE		
A012	INPUT N	N?	
A013	ENTER E0	ENTER E0	(Key in using EQN RCL E, RCL N, etc.)
A014	PSE		
A015	INPUT E	E?	
A016	RCL E		
A017	0 **i** 1		(Key in as 0, then **i**, then 1, press ENTER.)
A018	×		
A019	RCL+ N		
A020	STO F		
A021	360		
A022	STO B		
A023	CLSTK		
A024	STO P		
A025	STO Q		
A026	STO M		
A027	XEQ V001		
A028	STO R		
A029	STO+ P		
A030	ARG		
A031	RCL P		
A032	ARG		
A033	–		
A034	SIN		
A035	STO S		
A036	RCL R		
A037	ABS		
A038	STO× S		
A039	RCL P		

Traverse Closure with Area Calculation and Co-ordinates

A040	ABS		
A041	STO× S		
A042	RCL S		
A043	2		
A044	÷		
A045	STO+ Q		
A046	1		
A047	STO+ M		
A048	RCL C		
A049	x = 0?		
A050	GTO A068		
A051	RCL P		
A052	RCL+ F		
A053	XEQ X001		
A054	RCL M		
A055	RCL Q		
A056	ABS		
A057	R↓		
A058	STOP		
A059	RCL P		
A060	ARG		
A061	x < 0?		
A062	RCL+ B		
A063	→HMS		
A064	RCL P		
A065	ABS		
A066	STOP		
A067	GTO A027		
A068	RCL Q		
A069	ABS		
A070	RCL P		
A071	ARG		
A072	x < 0?		
A073	RCL+ B		
A074	→HMS		
A075	RCL P		
A076	ABS		
A077	RCL M		
A078	STOP		
A079	GTO A027		

This version of the program is designed to completely hide the complex number work that the calculator performs to compute the traverse. All values entered and returned are presented in much the same manner as with the HP-33S calculator.

Traverse Closure with Area Calculation and Co-ordinates

Notes

(1) Set the calculator into DEGREES mode (press MODE 1) before starting.

(2) This is a general traverse closure program that computes the azimuth and distance, plus area, to each point around the traverse, together with co-ordinates, if desired. It also computes the misclosure of a closed traverse.

(3) This program uses the V program as a sub-routine for data entry, so Program V must be in the calculator as program V (or the XEQ V001 at line A027 changed to reflect the changed label. The V program allows entry of azimuths in D.MMSS format (HP notation) and distances, while converting them to the internal format. Program V is one of the HP-35s Utilities programs (Utility 3).

(4) This program uses the X program as a sub-routine for co-ordinate return from the internal format., so program X must be in the calculator as program X (or the XEQ X001 at line A053 changed to reflect the label change). The X program allows co-ordinates to be returned from the internal processing format of the calculator. Program X is one of the HP-35s Utilities programs (Utility 4).

(5) The program allows the user to choose if the co-ordinates of each point are to be calculated. The user is prompted with USE COORDS briefly, followed by the C? prompt. If co-ordinates are desired, key in 1, if not, key in 0, then press R/S to continue.

(6) After each side (azimuth and distance) has been entered, the calculator produces the following output.

A. If the user has selected to use co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in the display in line 2, and the Easting co-ordinate of the point in line 1.

Stack Register	Contents
T	Area of the traverse thus far
Z	Northing co-ordinate of current point
Y	Easting co-ordinate of current point
X	Number of sides entered

The user can scroll through the stack, using the R↓ key, and can perform any other operation of interest to the data on the stack. This information is stored in memory registers for use later in the program, so the stack may be changed and worked with as needed. The user can also continue without viewing the stack.

When the user presses R/S, the calculator takes the line to the current forward point from the starting point, and converts it into the distance (which is placed in line 2, the X register) and the azimuth in degrees, minutes and seconds (HP notation) in line 1 of the display (the Y register).

Stack Register	Contents
T	
Z	
Y	Azimuth from start to current point (in D.MMSSss)
X	Distance from start to current point

When the user presses R/S again, the calculator prompts for the azimuth of the next side to be entered. The azimuth should be entered in HP notation (DDD.MMSSss).

B. If the user has selected not to show co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in line 2 of the display (the X register) and the distance of the misclosure or the line connecting the starting point to the current point, in line 1 of the display (Y register).

Stack Register	Contents
T	Area of the traverse thus far
Z	Azimuth of the line from the start to current point (in HP notation)
Y	Distance of the line from the start to the current point
X	Number of sides entered

By pressing the R↓ key, the user can see the azimuth in register Z of the stack. Pressing R↓ again will show the area to the current point.

If the rectangular components of the misclosure (or the line from the start to the current forward point) are needed, press RCL P, then XEQ X ENTER. The Easting component of the misclosure (ΔE) will be displayed in the X register (line 2), while the Northing component (ΔN) will be displayed in the Y register (line 1).

When the user presses R/S again, the calculator prompts for the azimuth of the next side.

(7) Azimuths are entered and displayed by themselves in HP notation, i.e., DDD.MMSSss.

(8) This program forms the basis of the two missing distances (2MD) program. Enter the known sides using this program to begin the 2MD computation process.

(9) In order to display the prompts, this program sets Flag 10. However, the program never ends, because it is up to the user to decide when to stop and move control elsewhere. So the program never clears Flag 10. If you require Flag 10 to be clear, in order to process equations, you must do this manually.

Traverse Closure with Area Calculation and Co-ordinates

Theory

The traverse closure programs works by converting the entered azimuths (in DDD.MMSS, or HP, notation) and distances into complex numbers (which act as 2-D vectors), which are then added to compute the location of points around the traverse. The area is computed by triangles developed by each new side of the traverse and the line from the starting point to the current forward point, and is updated with each new side. So the area is that of the polygon formed by the traverse entered thus far and the line from the start to the current point. This allows areas to be incremented for lot splitting calculations.

The azimuth and distance of the line from the start to the current point is also placed on the stack after each line. This allows a connecting line to be computed easily between two points. The final azimuth and distance is the traverse misclosure and the area is that of the traverse.

If the user chooses, the co-ordinates of the starting point may be entered, and if this choice is made, the calculator displays the co-ordinates of each point, in addition to the other information.

An arbitrary azimuth is satisfactory. Plane surveying assumptions apply. The program uses no error checking on entered data.

Sample Computation

Bearing	Distance
6° 53' 10"	72.00
112° 37' 20"	102.23
185° 39' 50"	29.04
181° 30' 00"	27.88
283° 54' 30"	102.38

Final Results			
	DE	=	0.0228
	DN	=	–0.0022
	Misclosure Length	=	0.0229
	Misclosure Bearing	=	95° 24' 15"
	Area	=	6,378.4660

Stepping through the Calculation

A. Without Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS, the C?

Key in 0, then press R/S.

Traverse Closure with Area Calculation and Co-ordinates

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	72.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	6.5310	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	107.9004	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	72.3939	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	100.1841	(distance from start)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	88.0808	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	102.4027	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.6396	(area thus far)
	103.5423	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	0.0229	(distance from start, also linear misclosure)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.4660	(area thus far)
	95.2415	(azimuth from start, also azimuth of misclosure)

Pressing RCL P, then XEQ X ENTER brings the misclosure to the display in rectangular form. The Northing component of the misclosure (ΔN) is in the Y register (line 1) and is –0.0022, while the Easting component of the misclosure (ΔE) is in the X register (line 2) and is 0.0228.

B. Using Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS briefly, then C?

Key in 1, then press R/S.

Calculator prompts ENTER N0 briefly, then N?

Key in 1000.000, press R/S.

Calculator prompts ENTER E0 briefly, then E?

Key in 500.000, press R/S.

Traverse Closure with Area Calculation and Co-ordinates

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	508.6325	(Easting co-ordinate of current forward point)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	1,071.4806	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	6.5310	(Azimuth for current forward point from starting point)
	72.0000	(Distance from starting point to current forward point)

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	602.9971	(Easting co-ordinate of current forward point)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	1,032.1575	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	72.3939	(Azimuth for current forward point from starting point)
	107.9004	(Distance from starting point to current forward point)

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	600.1310	(Easting co-ordinate of current forward point)
	3.0000	(number of sides entered)

Traverse Closure with Area Calculation and Co-ordinates

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	1,003.2593	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	88.0808	(Azimuth for current forward point from starting point)
	100.1841	(Distance from starting point to current forward point)

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	599.4012	(Easting co-ordinate of current forward point)
	4.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.6396	(area thus far)
	975.3888	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	103.5423	(Azimuth for current forward point from starting point)
	102.4027	(Distance from starting point to current forward point)

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	500.0228	(Easting co-ordinate of current forward point)
	5.0000	(number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.4660	(area thus far)
	999.9978	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	95.2415	(Azimuth for current forward point from starting point)
	0.0229	(Distance from starting point to current forward point)

Press RCL P, then XEQ X ENTER to bring the misclosure onto the stack in rectangular mode. The misclosure in Northings (latitude, ΔN) will be displayed in the Y register (line 1) as –0.0022. The misclosure in Eastings (departure, ΔE) will be displayed in the X register (line 2) and is 0.0228.

Traverse Closure with Area Calculation and Co-ordinates

Storage Registers Used

A Used by the V program for the entered azimuth.

B Stores 360 for azimuth correction.

C Test for displaying co-ordinates: 1 = YES; 0 = NO.

D Used by the V program for the entered distance.

E Easting co-ordinates of the starting point.

F Co-ordinates of starting point, as a complex number.

I Used by the V and X programs to address the additional storage registers.

M The number of sides entered.

N The Northing co-ordinate of the starting point.

P Current position of forward point, as a complex number.

Q Current area.

R Last side entered, as a complex number.

S Temporary storage for area calculation.

Statistical Registers: not used.

Other registers: 10, 11 and 12 used by the V program; 10 used by the X program.

Labels Used

Label **A** Length = 271 Checksum = 1175

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

The program labeled V, which takes an azimuth in degrees, minutes and seconds (in HP notation), and a distance, and converts them to a complex number for processing in the calculator. This routines uses storage location A and D, but copies out and replaces the contents of these storage locations in order to preserve them. It also uses storage register I for indirect addressing, which uses registers 10, 11 and 12.

The program labeled X, which takes a complex number and converts it to the rectangular components, is called for co-ordinate presentation in this program. It uses storage register I for indirect addressing, which uses register 10.

Label **V** Length = 128 Checksum = 39FE

Label **X** Length = 53 Checksum = C46D

Two Missing Distances (2MD) Calculation

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Version**: 3.0 **Mnemonic**: **M** for Two **M**issing Distances

Line	Instruction	Display	User Instructions
M001	LBL M		Enter known traverse sides using Program A.
M002	CLSTK		Press XEQ M ENTER.
M003	FS? 10		
M004	GTO M008		
M005	SF 1		
M006	SF 10		
M007	GTO M009		
M008	CF 1		
M009	ENTER 1ST AZ		(Key in using EQN RCL E, RCL N, etc.)
M010	PSE		
M011	INPUT U		
M012	RCL U		
M013	HMS→		
M014	STO U		
M015	ENTER 2ND AZ		(Key in using EQN RCL E, RCL N, etc.)
M016	PSE		
M017	INPUT V		
M018	RCL V		
M019	HMS→		
M020	STO V		
M021	RCL U		
M022	RCL– V		
M023	SIN		
M024	ABS		
M025	1/x		
M026	STO S		
M027	RCL P		
M028	ABS		
M029	STO× S		
M030	RCL P		
M031	ARG		
M032	RCL– V		
M033	SIN		
M034	ABS		
M035	RCL× S		
M036	STO A		
M037	1ST DIST		(Key in using EQN 1 RCL S, RCL T, etc.)
M038	PSE		
M039	VIEW A		Length of first missing side displayed

Two Missing Distances Calculation

M040	RCL P		
M041	ARG		
M042	RCL– U		
M043	SIN		
M044	ABS		
M045	RCL× S		
M046	STO B		
M047	2ND DIST		(Key in using EQN 2 RCL N, RCL D, etc.)
M048	PSE		
M049	VIEW B		Length of second missing side displayed
M050	FS? 1		
M051	CF 10		
M052	RTN		

Notes

(1) Calculator should be in DEGREES mode for this calculation.

(2) Enter all the known sides of the traverse using the program stored under A, i.e., the closure program with area (Closure 1).

(3) When all known sides have been entered and processed, press XEQ M ENTER. This will take you to the start of the 2MD program.

(4) The calculator will prompt with ENTER 1ST AZ, briefly, then prompt with U? Key in the azimuth of the first unknown line, in HP notation (DDD.MMSSss). Press R/S.

(5) The calculator will prompt with ENTER 2ND AZ, briefly, then prompt with V? Key in the azimuth of the second unknown line, in HP notation (DDD.MMSSss). Press R/S.

(6) The calculator will show 1ST DIST, briefly, then show A= and the distance for the first missing side. Press R/S.

(7) The calculator will show 2ND DIST, briefly, then show B= and the distance for the second missing side. Press R/S to finish the program..

(8) Azimuths are entered and displayed in HP notation, i.e., DDD.MMSS

(9) All the results are kept in storage, in case the user wants to access them again.

(10) The program sets Flag 10, so as to be able to display the prompts, and at the end of the program it resets Flag 10 to its previous value. Flag 1 is used to control this process.

Two Missing Distances Calculation

Theory

Once all the known sides have been entered (order does not matter), the resultant vector of the traverse is known. This forms one side of a triangle, with the two unknown lines forming the other two sides. We know the length of the resultant vector, and the azimuths of all three sides. So we can deduce all three angles.

The triangle is solved using the Sine Rule. The ratio of the sine of the angle opposite the resultant vector and the resultant vector's length are stored and used with the sines of the other two angles to compute the lengths of the two missing sides.

Note that all misclosure (errors) in the known part of the traverse will be included in the lengths of the unknown sides. The resulting traverse should close perfectly, but this is meaningless information as far as the traverse is concerned, as there is no redundant data to allow computation of a misclosure.

Sample Computations

1.

Azimuth	Distance
6° 53' 10"	72.00
112° 37' 20"	102.23
185° 39' 50"	29.04
181° 30' 00"	Missing distance 1
283° 54' 30"	Missing distance 2

Results Missing distance 1 = 27.883

Missing distance 2 = 102.403

2.

Azimuth	Distance
231° 06' 56"	199.123
7° 07' 30"	201.556
166° 30' 16"	Missing distance 1
63° 26' 06"	Missing distance 2

Results Missing distance 1 = 128.549

Missing distance 2 = 111.804

Two Missing Distances Calculation

Storage Registers Used

A Holds distance of first unknown side at end of computation.

B Holds distance of second unknown side at end of computation.

P Vector of known resultant side from program A, stored as a complex number.

S Sine ratio.

U Azimuth of first missing side.

V Azimuth of second missing side.

Plus those used by the Traverse Closure and Area program (A).

Statistical Registers: Not used.

Labels Used

Label **M** Length = 196 Checksum = 35F0

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Three-Point Horizontal Resection Reduction Program

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Version**: 1.0

Line	Instruction	Display	User Programming Instructions
R001	LBL R		
R002	CLSTK		
R003	FS? 10		
R004	GTO R008		
R005	SF 1		
R006	SF 10		
R007	GTO R009		
R008	CF 1		
R009	RESECTION		
R010	PSE		
R011	ENTER LEFT X		(Key in using EQN RCL E RCL N etc.)
R012	PSE		
R013	INPUT X		
R014	STO A		
R015	ENTER LEFT Y		(Key in using EQN RCL E RCL N etc.)
R016	PSE		
R017	INPUT Y		
R018	STO B		
R019	ENTER MID X		(Key in using EQN RCL E RCL N etc.)
R020	PSE		
R021	INPUT X		
R022	STO C		
R023	ENTER MID Y		(Key in using EQN RCL E RCL N etc.)
R024	PSE		
R025	INPUT Y		
R026	STO D		
R027	ENTER RIGHT X		(Key in using EQN RCL E RCL N etc.)
R028	PSE		
R029	INPUT X		
R030	STO E		
R031	ENTER RIGHT Y		(Key in using EQN RCL E RCL N etc.)
R032	PSE		
R033	INPUT Y		
R034	STO F		
R035	ENTER ALPHA		(Key in using EQN RCL E RCL N etc.)
R036	PSE		
R037	INPUT X		
R038	HMS→		
R039	STO G		

Three Point Horizontal Resection Reduction Program

R040	ENTER BETA		(Key in using EQN RCL E RCL N etc.)
R041	PSE		
R042	INPUT X		
R043	HMS→		
R044	STO H		
R045	RCL B		
R046	RCL– D		
R047	RCL A		
R048	RCL– C		
R049	0 **i** 1		Press the zero key, then **i**, then 1.
R050	×		
R051	+		
R052	STO L		
R053	RCL F		
R054	RCL– D		
R055	RCL E		
R056	RCL– C		
R057	0 **i** 1		Press the zero key, then **i**, then 1.
R058	×		
R059	+		
R060	STO K		
R061	360		
R062	STO Z		
R063	RCL L		
R064	ARG		
R065	RCL K		
R066	ARG		
R067	–		
R068	x < 0 ?		
R069	RCL+ Z		
R070	STO I		
R071	RCL+ G		
R072	RCL+ H		
R073	RCL Z		
R074	x < > y		
R075	–		
R076	STO S		
R077	RCL L		
R078	ABS		
R079	RCL H		
R080	SIN		
R081	×		
R082	RCL K		
R083	ABS		
R084	÷		
R085	RCL G		
R086	SIN		

Three Point Horizontal Resection Reduction Program

R087	÷		
R088	RCL S		
R089	SIN		
R090	÷		
R091	RCL S		
R092	TAN		
R093	1/x		
R094	+		
R095	1/x		
R096	ATAN		
R097	STO X		
R098	RCL L		
R099	ARG		
R100	180		
R101	+		
R102	RCL+ G		
R103	RCL+ X		
R104	STO Y		
R105	RCL L		
R106	ABS		
R107	RCL X		
R108	SIN		
R109	×		
R110	RCL G		
R111	SIN		
R112	÷		
R113	STO J		
R114	RCL Y		
R115	SIN		
R116	×		
R117	STO P		
R118	RCL J		
R119	RCL Y		
R120	COS		
R121	×		
R122	STO Q		
R123	RCL P		
R124	RCL+ C		
R125	STO X		
R126	UNKNOWN X =		(Key in using EQN RCL U RCL N etc.)
R127	PSE		
R128	VIEW X		
R129	UNKNOWN Y =		(Key in using EQN RCL U RCL N etc.)
R130	PSE		
R131	RCL Q		
R132	RCL+ D		
R133	STO Y		

Three Point Horizontal Resection Reduction Program

R134	VIEW Y		
R135	RCL I		
R136	RCL+ G		
R137	RCL+ H		
R138	→HMS		
R139	CHECK VALUE =		(Key in using EQN RCL C RCL H etc.)
R140	PSE		
R141	STOP		
R142	FS? 1		
R143	CF 10		
R144	RTN		

Notes

(1) Horizontal 3-point resection solution, based on measuring two angles between three known points at an unknown point, the location of which is to be computed.

(2) Brief prompts are provided before each requirement for data entry, as well as before results are displayed. Each prompt shows for about 1 second, and is then replaced by the value or request for input.

(3) Co-ordinates of the unknown point are displayed following brief prompts. They are also stored in storage registers X and Y for later retrieval.

(4) Angles are entered and displayed in HP notation, i.e., DDD.MMSS. Internal storage of angles and azimuths is in decimal degrees. Internal storage of lines uses the calculator's complex number format.

Theory

This 2-D resection uses Ormsby's solution. In the discussion below, A is the left point, B is the middle point, C is the right point, and P is the unknown point. The left angle is alpha (α) and the right angle is beta (β). The interior angle at B is gamma (γ). The angle at point A is x, which is the first objective of the solution. A diagram is shown on the next page.

α and β are angles observed from the point P to points A, B and C, whose co-ordinate are known.

$$BP = \frac{AB \sin x}{\sin \alpha} = \frac{BC \sin y}{\sin \beta}$$

and $(x + y) = (360° - (\alpha + \beta + \gamma)) = s$

$$\frac{AB}{\sin \alpha} \sin x = \frac{BC}{\sin \beta} \sin (s - x) = \frac{BC}{\sin \beta}(\sin s \cos x - \cos s \sin x)$$

$$\frac{AB}{\sin \alpha} \sin x = \frac{BC}{\sin \beta} \sin s \cos x - \frac{BC}{\sin \beta} \cos s \sin x$$

$$\sin x \left(\frac{AB}{\sin \alpha} + \frac{BC}{\sin \beta} \cos s\right) = \frac{BC}{\sin \beta} \sin s \cos x$$

Three Point Horizontal Resection Reduction Program

$$\left(\frac{AB}{\sin \alpha} + \frac{BC}{\sin \beta} \cos s\right) \frac{\sin \beta}{BC \sin s} = \cot x$$

$$\frac{AB \sin\beta}{BC \sin \alpha \sin s} + \frac{BC \cos s \sin \beta}{BC \sin s \sin \beta} = \cot x$$

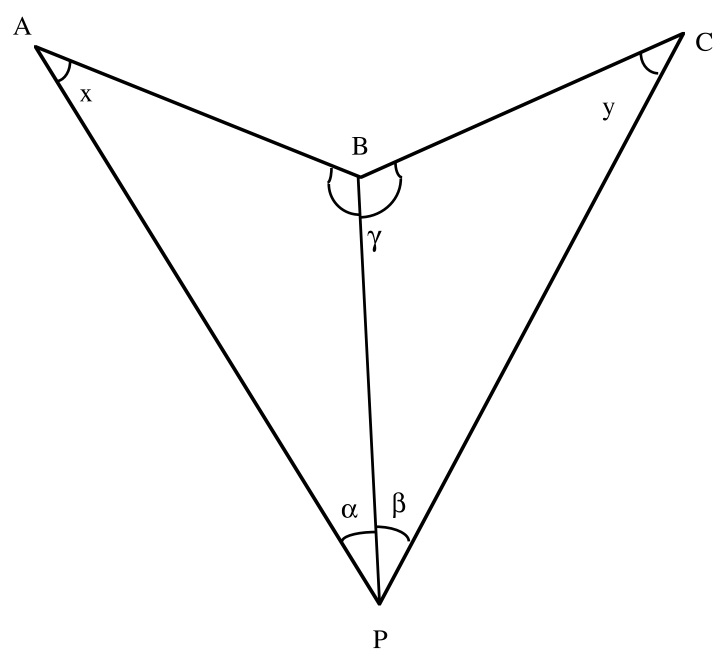
$$\frac{AB \sin \beta}{BC \sin \alpha \sin s} + \cot s = \cot x \qquad \text{[this is the equation solved first]}$$

$$y = s - x$$

With x and y determined, the sides AP, BP and CP can be calculated and hence the co-ordinates of P, as follows:

The azimuth of BP (Az_{BP}) can be determined using $Az_{BP} = Az_{AB} + \alpha + x$

The length of BP can be determined using $BP = \dfrac{AB \sin x}{\sin \alpha}$



Knowing the co-ordinates of B, Az_{BP} and BP, the co-ordinates of P can be easily computed. As a check, the equivalent solution can be obtain through the sides AP or CP, or using the angle y. Note that if P is close the danger circle, a solution will still be obtained, but the sum of $\alpha + \beta + \gamma$ will be close to 180°, probably in the range 175° to 185°. In this case, the solution will be highly sensitive to changes in α and β. If the solution is close to the danger circle, recomputed with the angles changed by about their precision and see how much the resulting co-ordinates change. It can be quite surprising! To facilitate this, press GTO R035, then R/S, to run the program with the same known points, but you can enter different observed angles.

Three Point Horizontal Resection Reduction Program

Azimuths in HP notation are used. Arbitrary co-ordinates are satisfactory. Plane surveying assumptions apply. The program uses no error checking on entered data. A check is made by showing the sum $\alpha + \beta + \gamma$. If this is close to 180°, the unknown point lies close to the danger circle and the result is highly suspect.

Sample Computation 1

Known Points

Point Name	X	Y
Point A	–25.336	778.136
Point B	–27.465	1179.927
Point C	–30.297	1555.643

Angles Left (α) = 136° 35' 26"

Right (β) = 27° 19' 24"

Results Unknown Point (P) X Co-ordinate = 26.009

Unknown Point (P) Y Co-ordinate = 1101.818

Check Angle = 344° 02' 32"

Sample Computation 2

Known Points

Point Name	X	Y
Point A	133.639	1548.712
Point B	158.065	1492.276
Point C	150.267	1353.056

Angles Left (α) = 5° 01' 48"

Right (β) = 3° 41' 29"

Results Unknown Point (P) X Co-ordinate = 116.784

Unknown Point (P) Y Co-ordinate = 1,186.818

Check Angle = 162° 06' 44"

This is not the ideal arrangement for a resection, as the measured angles are quite small. But the program will still produce an acceptable result.

This example is provided because the other example has negative co-ordinates and this tends to increase the chances of incorrect data entry. It happened to me, twice!

Three Point Horizontal Resection Reduction Program

Running the Program

Press XEQ R ENTER

Calculator displays RESECTION briefly, so that you know you are running the correct program.

Prompt ENTER LEFT X briefly, then X?

Enter X Co-ordinate for left known point.

Press R/S.

Prompt ENTER LEFT Y briefly, then Y?

Enter Y Co-ordinate for left known point.

Press R/S.

Prompt ENTER MID X briefly, then X?

Enter X Co-ordinate for middle known point.

Press R/S.

Prompt ENTER MID Y briefly, then Y?

Enter Y Co-ordinate for middle known point.

Press R/S.

Prompt ENTER RIGHT X briefly, then X?

Enter X Co-ordinate for right known point.

Press R/S.

Prompt ENTER RIGHT Y briefly, then Y?

Enter Y Co-ordinate for right known point.

Press R/S.

** Prompt ENTER ALPHA briefly, then X?

Enter left angle (α) in HP notation.

Press R/S.

Prompt ENTER BETA briefly, then X?

Enter right angle (β) in HP notation.

Press R/S.

Calculator displays RUNNING while doing the calculations.

Three Point Horizontal Resection Reduction Program

Prompt UNKNOWN X briefly, then X=

X co-ordinate of unknown point (P) is displayed.

Press R/S.

Prompt UNKNOWN Y briefly, then Y=

Y co-ordinate of unknown point (P) is displayed.

Press R/S.

Prompt CHECK VALUE briefly.

Sum $\alpha + \beta + \gamma$ is displayed in lower line of display in HP notation.

Check that value is not too close to 180°. At least 5 ° away, preferably 15° or more away.

Press R/S to clear flags. Program ends.

If you want to re-run the program with the same fixed points but different angles, press GTO R035, then R/S, and the program will start from the step labeled ** above, prompting with ENTER ALPHA. Changing the angles by small amounts can give you a good idea of the reliability of the solution.

Storage Registers Used

A	Left known point – X co-ordinate
B	Left known point – Y co-ordinate
C	Middle known point – X co-ordinate
D	Middle known point – Y co-ordinate
E	Right known point – X co-ordinate
F	Right known point – Y co-ordinate
G	Left measured angle — alpha (α)
H	Right measured angle — beta (β)
I	Interior angle at Middle known point — gamma (γ)
J	Distance from middle point to the unknown point
K	Vector from middle to right point (complex number format)
L	Vector from middle to left point (complex number format)
P	X co-ordinate of unknown point
Q	Y co-ordinate of unknown point
S	s = x + y in decimal degrees
X	Initial inputs, then angle x, then X co-ordinate of unknown point
Y	Initial inputs, then azimuth from middle to unknown point, then Y co-ordinate of unknown point
Z	360

Three Point Horizontal Resection Reduction Program

Labels Used

Label **R** Length = 581 Checksum = 17B0

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

The program sets flag 10, too allow equations to be displayed as prompts, and at the end of the program, resets flag 10 to its previous setting. The program uses flag 1 to record the state of flag 10 before the program started.

Co-ordinate–based Intersection Program (2-D)

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Version**: 1.0 **Mnemonic**: I for Intersection

Line	Instruction	Display	User Instructions
I001	LBL I		
I002	CLSTK		
I003	FS? 10		
I004	GTO I008		
I005	SF 1		
I006	SF 10		
I007	GTO I009		
I008	CF 1		
I009	INTERSECTION		(Key in EQN RCL I RCL N, etc.)
I010	PSE		
I011	ENTER X1		(Key in EQN RCL E RCL N etc.)
I012	PSE		
I013	INPUT X		
I014	ENTER Y1		(Key in EQN RCL E RCL N etc.)
I015	PSE		
I016	INPUT Y		
I017	RCL X		
I018	STO A		
I019	RCL Y		
I020	STO B		
I021	ENTER X2		(Key in EQN RCL E RCL N etc.)
I022	PSE		
I023	INPUT X		
I024	ENTER Y2		(Key in EQN RCL E RCL N etc.)
I025	PSE		
I026	INPUT Y		
I027	RCL X		
I028	STO C		
I029	RCL Y		
I030	STO D		
I031	ENTER ANG 1		(Key in EQN RCL E RCL N etc.)
I032	PSE		
I033	INPUT E		
I034	ENTER ANG 2		(Key in EQN RCL E RCL N etc.)
I035	PSE		
I036	INPUT F		
I037	RCL E		
I038	HMS→		
I039	STO E		

Co-ordinate–based Intersection Program (2-D)

I040	RCL F		
I041	HMS→		
I042	STO F		
I043	RCL B		
I044	RCL– D		
I045	RCL A		
I046	RCL F		
I047	TAN		
I048	÷		
I049	+		
I050	RCL C		
I051	RCL E		
I052	TAN		
I053	÷		
I054	+		
I055	RCL E		
I056	TAN		
I057	1/x		
I058	RCL F		
I059	TAN		
I060	1/x		
I061	+		
I062	STO G		
I063	÷		
I064	STO X		
I065	X OF POINT		(Key in EQN RCL X SPACE etc.)
I066	PSE		
I067	VIEW X		
I068	RCL C		
I069	RCL– A		
I070	RCL B		
I071	RCL F		
I072	TAN		
I073	÷		
I074	+		
I075	RCL D		
I076	RCL E		
I077	TAN		
I078	÷		
I079	+		
I080	RCL G		
I081	÷		
I082	STO Y		
I083	Y OF POINT		(Key in EQN RCL Y SPACE etc.)
I084	PSE		
I085	VIEW Y		
I086	RCL X		

Co-ordinate–based Intersection Program (2-D)

I087	RCL× G		
I088	RCL D		
I089	RCL– B		
I090	2		
I091	×		
I092	+		
I093	RCL÷ G		
I094	STO U		
I095	ALT X OF POINT		(Key in EQN RCL A RCL L etc.)
I096	PSE		
I097	VIEW U		
I098	RCL Y		
I099	RCL× G		
I100	RCL A		
I101	RCL– C		
I102	2		
I103	×		
I104	+		
I105	RCL÷ G		
I106	STO V		
I107	ALT Y OF POINT		(Key in EQN RCL A RCL L etc.)
I108	PSE		
I109	VIEW V		
I110	FS? 1		
I111	CF 10		
I112	RTN		

Notes

(1) Program assumes the use of co-ordinate for the location of the two known points, and produces an answer for the unknown point in the same co-ordinate system. The program is strictly 2-D, producing horizontal co-ordinates only.

(2) If co-ordinates are not known or needed, but angles at and a distance between the two known points are available, use the Triangles 1, Program 3 program, which solves a triangle in which two angles and the included side are known. This program will give the lengths of the two sides and the angle at the unknown point.

(3) Angles are entered in HP notation, i.e., DDD.MMSS. The angles at the known points are measured from the line between the known points.

(4) The calculator uses the notation X1, Y1, etc., while the theory part below, uses X_A, Y_A, etc. Assume that points A, B and C are the same as points 1, 2 and unknown.

(5) As will be appreciated, there are two possible solutions to this triangle, depending upon whether the point is to the 'left' or 'right' of the line between the known points. The alternative solution can be obtained by reversing the orientation of the situation, entering Point B co-ordinates as X1 and Y1, Point A co-ordinates as X2, Y2, the angle at Point B as ANG 1 and the angle at Point A

as ANG 2. The solution will follow. Note that entering the angles with negative signs will not give the correct result for the alternative solution.

The program computes the alternative solution for the given data. It is up to the user to decide on which side of the line the unknown point lies. The co-ordinates help with this.

(6) An alternative label to I may be used, if necessary, without any adverse consequences to the program.

Theory

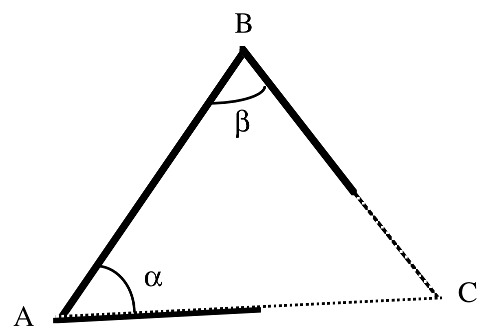
The intersection program is based on a triangle solution to the situation of knowing two angles and the included side. In this case, as co-ordinates are available, the solution can go directly to the co-ordinates of the unknown point.

The solution uses that of Richardus (1966), where:

$$X_C = \frac{(Y_A - Y_B) + X_A \cot\beta + X_B \cot\alpha}{\cot\alpha + \cot\beta}$$

$$Y_C = \frac{(X_B - X_A) + Y_A \cot\beta + Y_B \cot\alpha}{\cot\alpha + \cot\beta}$$

If the known points are A and B, the unknown point C, and the angles are α and A and β at B, then the co-ordinates of C can be computed directly from the above pair of equations. The situation is as shown in the figure.



Any plane co-ordinates may be used, using any units of distance measurement. Angles are expected in degrees, minutes and seconds, using HP notation. The solution assumes that plane surveying conditions apply, but this will usually be the case with this type of work and the precision expected will be such that any small differences caused by geodetic considerations will be within the range of likely random errors.

To compute the co-ordinates of a point that has been intersected from multiple known points, compute the various co-ordinates of the unknown point from each set of observations, and then take the mean of all the co-ordinates. It is possible to weight each co-ordinate on the basis of the intersection angle at the unknown point, then take a weighted mean but this is rather complex. It would be better to take the trouble to so a least squares adjustment in this case.

Co-ordinate–based Intersection Program (2-D)

Reference

RICHARDUS, P., 1966, *Project Surveying. General Adjustment and Optimization Techniques with Application to Engineering Surveying*. (Assisted by Allman, J.S.) New York : John Wiley and Sons, Inc.

Sample Computation 1

	X	Y	*Base angle*
A (1)	20 579·80	12 842·70	α (1) = 52° 37' 49"
B (2)	15 236·30	13 294·80	β (2) = 73° 22' 07"

Initial Solution:

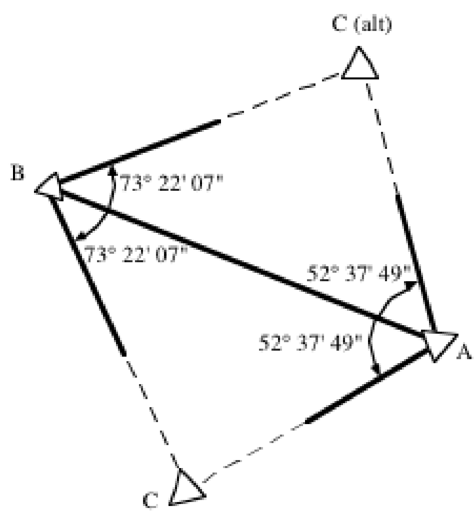
X_C = 16 313·13

Y_C = 8 138·18

Alternative Solution:

X_C = 17 164·20

Y_C = 18 197·20



The first solution is to the ‘left’ of the line from A to B, while the alternative solution is to the ‘right’ of the line from A to B. You should look at both solutions and check which of the solutions is the one you want.

To double-check, enter the data from the other end of the line and see if the same solutions are presented, but in reverse order.

Co-ordinate–based Intersection Program (2-D)

Sample Computation 2

Working the Alternative Solution

	X	Y	*Base angle*
B (1)	15 236·30	13 294·80	β (1) = 73° 22' 07"
A (2)	20 579·80	12 842·70	α (2) = 52° 37' 49"

Here the same situation is reversed, so that the alternative solution and the initial solution are swapped.

Initial Solution:

X_C = 17 164·20

Y_C = 18 197·20

Alternative Solution:

X_C = 16 313·13

Y_C = 8 138·18

Storage Registers Used

A X_1 (or X_A)
B Y_1 (or Y_A)
C X_2 (or X_B)
D Y_2 (or Y_B)
E Angle at A or 1, α
F Angle at B or 2, β
G cot α + cot β
U Storage for the alternative X_C co-ordinate.
V Storage for the alternative Y_C co-ordinate.
X Temporary storage for input X values and initial X_C location
Y Temporary storage for input Y values and initial Y_C location

Labels Used

Label **I** Length = 452 Checksum = FACE

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Co-ordinate–based Intersection Program (2-D)

Running the Program

With everything to hand, press XEQ I ENTER.

The program shows INTERSECTION briefly.

The calculator display ENTER X1 briefly, then prompts X? Key in the X value of the first point. Press R/S.

The calculator display ENTER Y1 briefly, then prompts Y? Key in the Y value of the first point. Press R/S.

The calculator display ENTER X2 briefly, then prompts X? Key in the X value of the second point. Press R/S.

The calculator display ENTER Y2 briefly, then prompts Y? Key in the Y value of the second point. Press R/S.

The calculator displays ENTER ANG 1 briefly, then prompts E? Key in the angle at the first point, Point A, or α. Use DD.MMSS (HP) notation. Press R/S.

The calculator displays ENTER ANG 2 briefly, then prompts F? Key in the angle at the second point, Point B, or β. Use DD.MMSS (HP) notation. Press R/S.

The calculator displays X OF POINT briefly, then displays X = and the initial X value of the unknown point. Press R/S to continue (the value is stored in the X memory register).

The calculator displays Y OF POINT briefly, then displays Y = and the initial Y value of the unknown point. Press R/S to continue (the value is stored in the Y memory register).

The calculator displays ALT X OF POINT briefly, then displays U = and the alternative X value of the unknown point. Press R/S to continue (the value is stored in the U memory register).

The calculator displays ALT Y OF POINT briefly, then displays V = and the alternative Y value of the unknown point. The value is stored in the V memory register.

Press R/S again. The calculator will clear the flags and end the program.

To run additional intersected points from the same pair of known points, press C to get out of the program, press GTO I031, then press R/S to start the program. The program prompts for the first angle, with the co-ordinates of the two known points already in the correct storage registers.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Programmer: Dr. Bill Hazelton

Date: December, 2007. **Version**: 1.0

Line	Instruction	Display	User Instructions
B001	LBL B		Press XEQ B ENTER
B002	CLSTK		
B003	SF 10		
B004	CF 3		
B005	TRAVERSE 2		(Key in using EQN RCL T, RCL R, etc.)
B006	PSE		
B007	USE COORDS 0-1	USE COORDS 0-1	(Key in using EQN RCL U, RCL S, etc.)
B008	PSE		
B009	INPUT C	C?	
B010	RCL C		
B011	x = 0?		
B012	GTO B024		
B013	ENTER N0	ENTER N0	(Key in using EQN RCL E, RCL N, etc.)
B014	PSE		
B015	INPUT N	N?	
B016	ENTER E0	ENTER E0	(Key in using EQN RCL E, RCL N, etc.)
B017	PSE		
B018	INPUT E	E?	
B019	RCL E		
B020	0 **i** 1		(Key in as 0, then **i**, then 1, press ENTER.)
B021	×		
B022	RCL+ N		
B023	STO F		
B024	360		
B025	STO B		
B026	CLSTK		
B027	STO P		
B028	STO Q		
B029	STO M		
B030	XEQ W001		
B031	STO U		
B032	FS? 3		
B033	GTO B086		
B034	STO T		
B035	STO+ P		
B036	ARG		
B037	RCL P		
B038	ARG		

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

B039	–		
B040	SIN		
B041	STO S		
B042	RCL U		
B043	ABS		
B044	STO× S		
B045	RCL P		
B046	ABS		
B047	STO× S		
B048	RCL S		
B049	2		
B050	÷		
B051	STO+ Q		
B052	1		
B053	STO+ M		
B054	RCL C		
B055	x = 0?		
B056	GTO B074		
B057	RCL P		
B058	RCL+ F		
B059	XEQ X001		
B060	RCL M		
B061	RCL Q		
B062	ABS		
B063	R↓		
B064	STOP		
B065	RCL P		
B066	ARG		
B067	x < 0?		
B068	RCL+ B		
B069	→HMS		
B070	RCL P		
B071	ABS		
B072	STOP		
B073	GTO B030		
B074	RCL Q		
B075	ABS		
B076	RCL P		
B077	ARG		
B078	x < 0?		
B079	RCL+ B		
B080	→HMS		
B081	RCL P		
B082	ABS		
B083	RCL M		
B084	STOP		
B085	GTO B030		

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

B086	ARC CHORD		
B087	PSE		
B088	0		
B089	STO Z		
B090	CONCAVE 0–1		
B091	PSE		
B092	INPUT Z		
B093	1		
B094	STO Y		
B095	TANGENT 0–1		
B096	PSE		
B097	INPUT Y		
B098	$x = 0$?		
B099	GTO B158		
B100	90		
B101	RCL T		
B102	ARG		
B103	RCL U		
B104	ARG		
B105	–		
B106	ABS		
B107	$x \leq y$?		
B108	GTO B112		
B109	RCL B		
B110	$x <> y$		
B111	–		
B112	STO H		
B113	RCL U		
B114	ABS		
B115	RCL H		
B116	SIN		
B117	÷		
B118	2		
B119	÷		
B120	STO R		
B121	x^2		
B122	RCL H		
B123	→RAD		
B124	RCL H		
B125	2		
B126	×		
B127	SIN		
B128	2		
B129	÷		
B130	–		
B131	×		
B132	ABS		

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

B133	STO S		
B134	RCL Z		
B135	$x = 0$?		
B136	GTO B140		
B137	RCL S		
B138	STO– Q		
B139	GTO B142		
B140	RCL S		
B141	STO+ Q		
B142	RADIUS =		
B143	PSE		
B144	VIEW R		
B145	ARC LENGTH =		
B146	PSE		
B147	RCL H		
B148	→RAD		
B149	2		
B150	×		
B151	RCL× R		
B152	STO L		
B153	VIEW L		
B154	CLSTK		
B155	CF 3		
B156	RCL U		
B157	GTO B034		
B158	ENTER RADIUS		
B159	PSE		
B160	INPUT R		
B161	RCL U		
B162	ABS		
B163	2		
B164	÷		
B165	RCL÷ R		
B166	ASIN		
B167	STO H		
B168	→RAD		
B169	2		
B170	RCL× H		
B171	SIN		
B172	2		
B173	÷		
B174	–		
B175	RCL R		
B176	x^2		
B177	×		
B178	STO S		
B179	RCL Z		

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

B180	$x = 0$?		
B181	GTO B185		
B182	RCL S		
B183	STO– Q		
B184	GTO B187		
B185	RCL S		
B186	STO+ Q		
B187	ARC LENGTH =		
B188	PSE		
B189	RCL H		
B190	→RAD		
B191	2		
B192	×		
B193	RCL× R		
B194	STO L		
B195	VIEW L		
B196	CLSTK		
B197	CF 3		
B198	RCL U		
B199	GTO B034		

This program is designed to completely hide the complex number work that the calculator performs to compute the traverse. All values entered and returned are presented in much the same manner as with the HP-33S calculator and its predecessors.

This program is a development and extension of the Closure 1A program. It requires the use of a slightly modified version of the V program, labeled W.

Notes

(1) Set the calculator into DEGREES mode (press MODE 1) before starting.

(2) This is a general traverse closure program that computes the azimuth and distance, plus area, to each point around the traverse, together with co-ordinates, if desired. It also computes the misclosure of a closed traverse.

(3) The program can also calculate the area between the chord and arc of a circular curve, and include this in the overall area. It can work with curves that meet the adjacent lines as tangents, or lines that intersect the arc at an angle. It can also work with areas that are added to or subtracted from the total area of the polygon, i.e., convex and concave curves.

(4) This program uses the W program as a sub-routine for data entry, so Program W must be in the calculator as program W (or the XEQ W001 at line B030 changed to reflect the changed label). The W program allows entry of azimuths in D.MMSS format (HP notation) and distances, while converting them to the internal format. Program W is one of the HP-35s Utilities programs (Utility 5). It differs from the V program in that it check for negative distances, converts them to positive distances, and sets Flag 3 if it finds a negative distance. This is used in the main program to indicate that a chord has been entered.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

(5) This program uses the X program as a sub-routine for co-ordinate return from the internal format, so program X must be in the calculator as program X (or the XEQ X001 at line B059 changed to reflect the label change). The X program allows co-ordinates to be returned from the internal processing format of the calculator. Program X is one of the HP-35s Utilities programs (Utility 4).

(6) The program allows the user to choose if the co-ordinates of each point are to be calculated. The user is prompted with USE COORDS briefly, followed by the C? prompt. If co-ordinates are desired, key in 1, if not, key in 0, then press R/S to continue.

(7) After each side (azimuth and distance) has been entered, the calculator produces the following output.

A. If the user has selected to use co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in the display in line 2, and the Easting co-ordinate of the point in line 1.

Stack Register	Contents
T	Area of the traverse thus far (including curves)
Z	Northing co-ordinate of current point
Y	Easting co-ordinate of current point
X	Number of sides entered

The user can scroll through the stack, using the R↓ key, and can perform any other operation of interest to the data on the stack. This information is stored in memory registers for use later in the program, so the stack may be changed and worked with as needed. The user can also continue without viewing the stack.

When the user presses R/S, the calculator takes the line to the current forward point from the starting point, and converts it into the distance (which is placed in line 2, the X register) and the azimuth in degrees, minutes and seconds (HP notation) in line 1 of the display (the Y register).

Stack Register	Contents
T	
Z	
Y	Azimuth from start to current point (in D.MMSSss)
X	Distance from start to current point

When the user presses R/S again, the calculator prompts for the azimuth of the next side to be entered. The azimuth should be entered in HP notation (DDD.MMSSss).

B. If the user has selected not to show co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in line 2 of the display (the X register) and the distance of the misclosure or the line connecting the starting point to the current point, in line 1 of the display (Y register).

Stack Register	Contents
T	Area of the traverse thus far
Z	Azimuth of the line from the start to current point (in HP notation)
Y	Distance of the line from the start to the current point
X	Number of sides entered

By pressing the R↓ key, the user can see the azimuth in register Z of the stack. Pressing R↓ again will show the area to the current point.

If the rectangular components of the misclosure (or the line from the start to the current forward point) are needed, press RCL P, then XEQ X ENTER. The Easting component of the misclosure (ΔE) will be displayed in the X register (line 2), while the Northing component (ΔN) will be displayed in the Y register (line 1).

When the user presses R/S again, the calculator prompts for the azimuth of the next side.

(8) Azimuths are entered and displayed by themselves in HP notation, i.e., DDD.MMSSss.

(9) This program forms the basis of the two missing distances (2MD) program. Enter the known sides using this program to begin the 2MD computation process.

(10) In order to display the prompts, this program sets Flag 10. However, the program never ends, because it is up to the user to decide when to stop and move control elsewhere. So the program never clears Flag 10. If you require Flag 10 to be clear, in order to process equations, you must do this manually.

(11) The program uses Flag 3 to see if a chord has been entered as the current side. If so, the area between the chord and the arc is calculated and added to the area calculation. The program then returns and calculated the area inside the polygon thus far, using the entered chord. So when each new point is displayed, the current area includes that between chords and arcs up to that point.

(12) The program will work with curves that meet the straight line at the start of the curve as a tangent without additional work, except indicating that an arc has been entered. This is done by entering a negative distance for the side being entered. The user is prompted about whether the curve is convex or concave, and whether the curve and the starting point meet as tangents.

(13) If the curve and the line at the starting point do not meet as tangents, the user is prompted to enter the radius of the curve, and calculation proceeds.

(14) The program calculates the arc length of the curve, as well as the radius if it was a ‘tangent’ curve. Note that the relationship (tangent or not) between the curve and the adjacent line at the end of the curve is irrelevant to the calculation of the

area, etc.; it is only the relationship between the curve and the line at the starting point that matters.

Theory

The traverse closure programs works by converting the entered azimuths (in DDD.MMSS, or HP notation) and distances into complex numbers (which act as 2-D vectors), which are then added to compute the location of points around the traverse. The area is computed by triangles developed by each new side of the traverse and the line from the starting point to the current forward point, and is updated with each new side. So the area is that of the polygon formed by the traverse entered thus far and the line from the start to the current point. This allows areas to be incremented for lot splitting calculations.

The azimuth and distance of the line from the start to the current point is also placed on the stack after each line. This allows a connecting line to be computed easily between two points. The final azimuth and distance thus shown is the traverse misclosure and the area is that of the traverse.

If the user chooses, the co-ordinates of the starting point may be entered, and if this choice is made, the calculator displays the co-ordinates of each point, in addition to the other information.

An arbitrary azimuth is satisfactory. Plane surveying assumptions apply. The program uses no error checking on entered data.

Dealing with Curves

When the program detects the presence of a chord, rather than a simple straight (indicated by entering a negative value for the distance of the chord), the program then prompts for whether or not the curve is concave or convex (convex is the default) with respect to the polygon, and whether or not the line entered as the side immediately before the chord formed a tangent to the curve (the default case is a tangent).

If the curve is convex, its area is added to the total being accumulated; if it is concave, the area is subtracted from the running total. This means that the area accumulated to the forward point (in this case, the end of the curve) is correct at the end of each line calculation.

If the previous side was a tangent to the curve, the program computes all the information needed from the deflection of the chord from the tangent and the tangent length. It displays the radius as a check (this is usually known), as well as the arc length. If the previous line was not a tangent to the curve, the program prompts for the radius, and computes the area. It also displays the arc length of the curve.

Once the area between the chord and the arc has been computed, the program computes the next line in the traverse, using the chord data, in the usual way. Note that it is irrelevant as to whether or not the line after the chord is tangential to the curve. This means that you can avoid having to enter the radius of a curve if only one end is not tangential, by computing around the traverse so that the tangential end comes first.

Tangent Case

In the case where the line coming into the curve (shown in black in Figure 1) is a tangent to the curve, the deflection of the chord (shown in blue in Figure 1) from that tangent is equal to half the deflection angle of the whole curve (θ), i.e., the angle at the center of the arc (where the radii meet). The radius of the curve can be calculated using the deflection angle and length of the chord, d, thus:

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

$$r = \frac{\left(\frac{d}{2}\right)}{\sin\left(\frac{\theta}{2}\right)}$$

The area of the segment of the circle between the chord and the arc (shown in red in Figure 1), A, can be calculated using:

$$A = \frac{\theta}{2}r^2 - \frac{1}{2}r^2\sin\theta = \left(\frac{\theta}{2} - \frac{\sin\theta}{2}\right)r^2$$

where θ indicates the size of the angle in radians.

The arc length, a, can be calculated using:

$$a = r\ \theta$$

where θ is in radians.

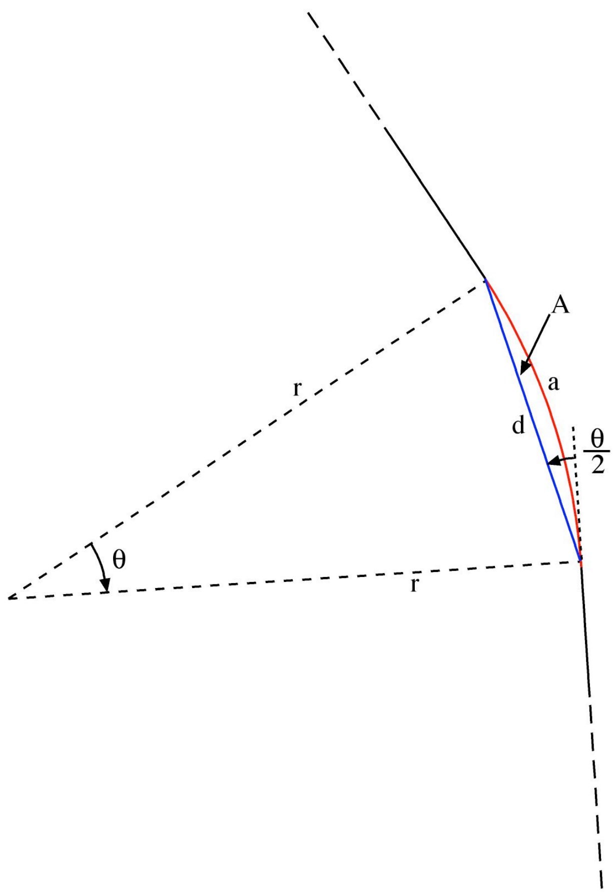


Figure 1 Diagram of the case where the line coming into the curve is a tangent to the curve.

Note that the state of tangency at the end of the curve had no affect on the calculations.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

The Non-Tangent Case

While it is more usual for curves to connect to other lines at tangent points, this tends to be for a large parcel, In many cases, a curved section of road forms a boundary to several parcels, and the boundary lines the separate these parcels meet the curve at other angles and are not tangential.

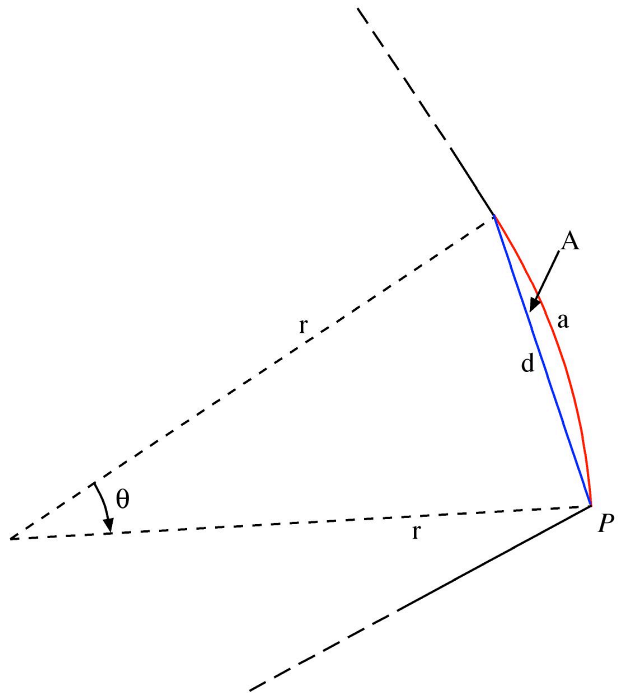


Figure 2 Diagram of the non-tangent case, where the line coming into the curve at point P is not tangential to the curve.

In Figure 2, the line that meets the curve at point P is clearly not a tangent to the curve. Therefore the deflection angle is no help in the calculations. All that is needed for calculating the area and the arc length are the radius (r) and the deflection angle of the curve (θ). So the user must supply the radius of the curve at this point in the calculation.

Knowing the radius and the chord length, half the deflection angle can be calculated thus:

$$\frac{\theta}{2} = \arcsin\left(\frac{d/2}{r}\right)$$

The segment area (A) and arc length (a) calculations can now be done using the same formulae as given for the tangent case.

Convex and Concave

It is important to know whether the area of the segment between the chord and the arc should be added to the area of the polygon derived using the chord, or subtracted. This depends upon whether the segment is part of a concave or convex structure.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

If the segment is outside the polygon of which the chord is a side, it is the convex case. If the segment is inside the polygon, then it is the concave case.

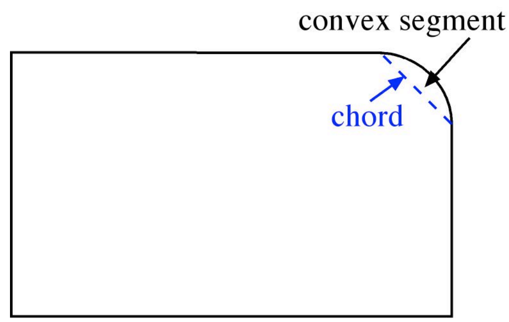


Figure 3 A convex segment, because the segment is outside the polygon

In Figure 3, the segment is outside the five-sided polygon formed by the chord and the other sides. The area of the segment must therefore be added to the area of the polygon. The convex case is probably more common, so it is set up to be the default in the program.

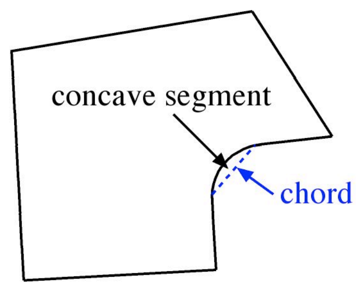


Figure 4 A concave segment, because the segment is inside the polygon.

In Figure 4, the segment is inside the seven-sided polygon formed by the chord and the other six sides. The area of the segment must therefore be subtracted from the area of the polygon.

Sample Computations

1. Without Curves

Azimuth	Distance
6° 53' 10"	72.00
112° 37' 20"	102.23
185° 39' 50"	29.04
181° 30' 00"	27.88
283° 54' 30"	102.38

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Final Results			
	DE	=	0.0228
	DN	=	–0.0022
	Misclosure Length	=	0.0229
	Misclosure Azimuth	=	95° 24' 15"
	Area	=	6,378.4660

Stepping through the Calculation

a. Without Co-ordinates

Press XEQ B ENTER

Calculator notes that this is the TRAVERSE 2 program.

Calculator prompts with USE COORDS 0–1, then C?

Key in 0, then press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	72.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	6.5310	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	107.9004	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	72.3939	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	100.1841	(distance from start)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	88.0808	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	102.4027	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.6396	(area thus far)
	103.5423	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	0.0229	(distance from start, also linear misclosure)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.4660	(area thus far)
	95.2415	(azimuth from start, also azimuth of misclosure)

Pressing RCL P, then XEQ X ENTER brings the misclosure to the display in rectangular form. The Northing component of the misclosure (ΔN) is in the Y register (line 1) and is –0.0022, while the Easting component of the misclosure (ΔE) is in the X register (line 2) and is 0.0228.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

b. Using Co-ordinates

Press XEQ B ENTER

Calculator notes that this is the TRAVERSE 2 program.

Calculator prompts with USE COORDS 0–1 briefly, then C?

Key in 1, then press R/S.

Calculator prompts ENTER N0 briefly, then N?

Key in 1000.000, press R/S.

Calculator prompts ENTER E0 briefly, then E?

Key in 500.000, press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	508.6325	(Easting co-ordinate of current forward point)
	1.0000	(Number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(Area thus far)
	1,071.4806	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	6.5310	(Azimuth for current forward point from starting point)
	72.0000	(Distance from starting point to current forward point)

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	602.9971	(Easting co-ordinate of current forward point)
	2.0000	(Number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(Area thus far)
	1,032.1575	(Northing co-ordinate of current forward point)

Press R/S.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Display shows:	72.3939	(Azimuth for current forward point from starting point)
	107.9004	(Distance from starting point to current forward point)

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	600.1310	(Easting co-ordinate of current forward point)
	3.0000	(Number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(Area thus far)
	1,003.2593	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	88.0808	(Azimuth for current forward point from starting point)
	100.1841	(Distance from starting point to current forward point)

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	599.4012	(Easting co-ordinate of current forward point)
	4.0000	(Number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.6396	(Area thus far)
	975.3888	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	103.5423	(Azimuth for current forward point from starting point)
	102.4027	(Distance from starting point to current forward point)

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Display shows:	500.0228	(Easting co-ordinate of current forward point)
	5.0000	(Number of sides entered)

Press the R↓ key, and the display becomes:

	6,378.4660	(Area thus far)
	999.9978	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	95.2415	(Azimuth for current forward point from starting point)
	0.0229	(Distance from starting point to current forward point)

Press RCL P, then XEQ X ENTER to bring the misclosure onto the stack in rectangular mode. The misclosure in Northings (latitude, ΔN) will be displayed in the Y register (line 1) as –0.0022. The misclosure in Eastings (departure, ΔE) will be displayed in the X register (line 2) and is 0.0228.

2. With Curves

a. Convex Tangent Case

In this example, the situation of using a curve that is tangential to the adjacent lines is used. The output for not using co-ordinates is provided.

Azimuth	Distance
0° 00' 00"	100.00
90° 00' 00"	100.00
180° 00' 00"	50.00
225° 00' 00"	70.71 chord
270° 00' 00"	50.00

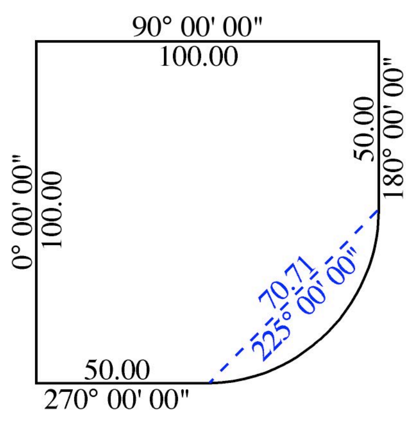


Figure 5 Polygon with a convex curve that is tangential to the adjacent polygon lines.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Press XEQ B ENTER

Calculator notes that this is the TRAVERSE 2 program.

Calculator prompts with USE COORDS 0–1, then C?

Key in 0, then press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 0.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 100.00, press R/S.

Display shows:	100.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	0.0000	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 90.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 100.00, press R/S.

Display shows:	141.4214	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	5,000.0000	(area thus far)
	45.0000	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 180.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 50.00, press R/S.

Display shows:	111.8034	(distance from start)
	3.0000	(number of sides entered)

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Press the R↓ key twice, and the display becomes:

7,500.0000	(area thus far)
63.2606	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4, the chord

Calculator prompts with A? for azimuth of side.

Key in 225.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in –70.71, press R/S. (Key in the number, then press the +/– key)

The calculator briefly display ARC CHORD to indicate that the side entered was a chord, and the '3' annunciator in the display should come on.

The calculator briefly displays CONCAVE 0–1, then prompts with Z?

As this is a convex curve, key in 0 and press R/S, or just press R/S as the 0 is already there. (0 means 'no' as the response to the question of concavity.)

The calculator briefly displays TANGENT 0–1, then prompts with Y?

As the last side is a tangent to the curve, key in 1 and press R/S, or just press R/S as the 1 is already there. (1 means 'yes' as the response to the question about the previous side being a tangent.)

The calculator briefly displays RADIUS =.

Display shows: R =

49.9995

Press R/S.

The calculator briefly displays ARC LENGTH =.

Display shows: L =

78.5391

Press R/S.

The '3' annunciator in the display should go out.

Display shows:	50.0005	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

9,463.4697	(area thus far)
89.5958	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Side 5

Calculator prompts with A? for azimuth of side.

Key in 270.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 50.00, press R/S.

Display shows:	0.0007	(distance from start)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	9,463.4577	(area thus far)
	45.0000	(azimuth from start in HP notation (D.MMSS))

Showing the Misclosure

Press RCL P, then XEQ X ENTER to bring the misclosure onto the stack in rectangular mode. The misclosure in Northings (latitude, ΔN) will be displayed in the Y register (line 1) as 0.0005. The misclosure in Eastings (departure, ΔE) will be displayed in the X register (line 2) and is 0.0005.

The process of computing around the traverse and computing co-ordinates is almost exactly the same, except an addition R/S is needed during the display of the co-ordinate data.

b. Concave Non-Tangent Case

Azimuth	Distance
0° 00' 00"	50.00
90° 00' 00"	100.00
180° 00' 00"	50.00 chord
270° 00' 00"	100.00

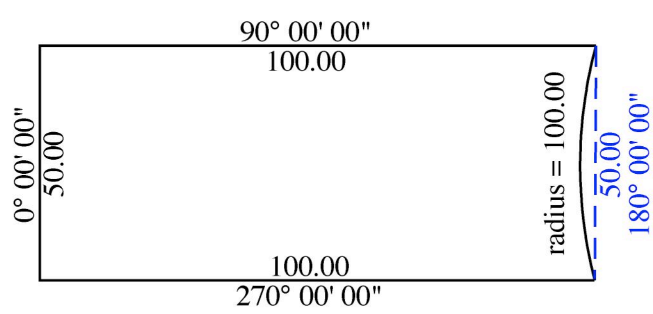


Figure 6 Polygon with a concave side that is not tangential to the adjacent sides.

In this example, co-ordinates are not displayed. The difference between the two processes are that an extra press of R/S is needed when displaying co-ordinates.

The radius of the curve is 100.00.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Press XEQ B ENTER

Calculator notes that this is the TRAVERSE 2 program.

Calculator prompts with USE COORDS 0–1, then C?

Key in 0, then press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 0.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 50.00, press R/S.

Display shows:	50.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	0.0000	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 90.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 100.00, press R/S.

Display shows:	111.8034	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	2,500.0000	(area thus far)
	63.2606	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 3, the chord

Calculator prompts with A? for azimuth of side.

Key in 180.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in –50.00, press R/S.

The calculator briefly displays ARC CHORD to show that the last side entered was a chord.

The calculator briefly displays CONCAVE 0–1, then prompts with Z?

In this case, the curve is concave, so key in 1 (for ‘yes’) and press R/S.

The calculator briefly displays TANGENT 0–1, then prompts with Y?

In this case, the previously entered line is not tangential to the curve, so key in 0 (for 'no') and press R/S.

The calculator briefly displays ENTER RADIUS, then prompts with R?

Key in 100.00 and press R/S.

The calculator briefly displays ARC LENGTH =

Display shows: L =

50.5361

Press R/S.

Display shows:	100.0000	(distance from start)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,893.8120	(area thus far)
	90.0000	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 270.0000, press R/S.

Calculator prompts with D? for distance of side.

Key in 100.00, press R/S.

Display shows:	0.0000	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,893.8120	(area thus far)
	0.0000	(azimuth from start in HP notation (D.MMSS))

Showing the Misclosure

Press RCL P, then XEQ X ENTER to bring the misclosure onto the stack in rectangular mode. The misclosure in Northings (latitude, ΔN) will be displayed in the Y register (line 1) as 0.0000. The misclosure in Eastings (departure, ΔE) will be displayed in the X register (line 2) and is 0.0000.

Storage Registers Used

A Used by the W program for the entered azimuth.

B Stores 360 for azimuth correction.

C Test for displaying co-ordinates: 1 = YES; 0 = NO.

D Used by the W program for the entered distance.

E Easting co-ordinates of the starting point.

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

F Co-ordinates of starting point, as a complex number.

H Half angle at center of arc (deflection angle, θ)

I Used by the W and X programs to address the additional storage registers.

L Arc length

M The number of sides entered.

N The Northing co-ordinate of the starting point.

P Current position of forward point, as a complex number.

Q Current area.

R Radius of curve

S Temporary storage for area calculation.

T Side before the last entered

U Last side entered, as a complex number.

Y Test for tangent to curve: 1 = YES (default); 0 = NO.

Z Test for convex or concave curve: 0 = CONVEX (default), 1 = CONCAVE.

Statistical Registers: not used.

Other registers: 10, 11 and 12 used by the W program; 10 used by the X program.

Labels Used

Label **B** Length = 732 Checksum = B9E5

Use the Length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

The program labeled W, which takes an azimuth in degrees, minutes and seconds (in HP notation), and a distance, and converts them to a complex number for processing in the calculator. This routines uses storage location A and D, but copies out and replaces the contents of these storage locations in order to preserve them. It also uses storage register I for indirect addressing, which uses registers 10, 11 and 12.

The program labeled X, which takes a complex number and converts it to the rectangular components, is called for co-ordinate presentation in this program. It uses storage register I for indirect addressing, which uses register 10.

Label **W** Length = 146 Checksum = A1A3

Label **X** Length = 53 Checksum = C46D

Traverse Closure with Area Calculation and Co-ordinates and Calculation of Curve Areas

Flags Used

The program sets Flag 10 to allow display of equations as prompts. Because the program never comes to a defined end (this is up to the user to decide), Flag 10 is never reset to its previous value. The user will have to do this if it is required.

The program uses Flag 3 to indicate that a chord has been entered, and so to calculate the segment area. This is done by noting the presence of a negative distance entered for the chord. Flag 3 is reset once the curve calculation has been completed. The state of Flag 3 is indicated in the display, with s small 3 at the top of the display turned on when the flag is set.

HP-35s Calculator Program — Closure 6A

Crandall's Adjustment for a Closed Traverse

Programmer: Dr. Bill Hazelton

Date: March, 2008. Version: 1,0 Mnemonic: **K** for '**K**randall' (sorry)

Line	Instruction	Display	User Instructions
K001	LBL K		LBL K
K002	CLSTK		CLEAR 5
K003	FS? 10		FLAGS 3 .0
K004	GTO K008		
K005	SF 1		FLAGS 1 1
K006	SF 10		FLAGS 1 .0
K007	GTO K009		
K008	CF 1		FLAGS 2 1
K009	CLΣ		CLEAR 4
K010	CRANDALL ADJ		(Key in using EQN RCL C, RCL R, etc.)
K011	PSE		PSE
K012	100		
K013	STO I		STO I
K014	STO (I)		STO (I) [(I) is on the 0 key]
K015	ENTER FIX SIDES		(Key in using EQN RCL E, RCL N, etc.)
K016	PSE		PSE
K017	ENTER FIX AZ		(Key in using EQN RCL E, RCL N, etc.)
K018	PSE		PSE
K019	INPUT Q	Q?	INPUT Q
K020	RCL Q		
K021	x < 0?		x?0 3
K022	GTO K036		
K023	HMS→		HMS→
K024	STO Q		STO Q
K025	ENTER FIX DIST		(Key in using EQN RCL E, RCL N, etc.)
K026	PSE		PSE
K027	INPUT D	D?	INPUT D
K028	RCL Q		
K029	SIN		
K030	RCL× D		
K031	RCL Q		
K032	COS		
K033	RCL× D		
K034	Σ+		
K035	GTO K017		
K036	—30		
K037	STO I		STO I
K038	0		
K039	STO (I)		STO (I) [(I) is on the 0 key]

Crandall's Adjustment for a Closed Traverse

Line	Instruction	Display	User Instructions
K040	−31		
K041	STO I		STO I
K042	0		
K043	STO (I)		STO (I)
K044	−32		
K045	STO I		STO I
K046	0		
K047	STO (I)		STO (I)
K048	STO K		STO K
K049	20		
K050	STO I		STO I
K051	21		
K052	STO J		STO J
K053	ENTER VAR SIDES		(Key in using EQN RCL E, RCL N, etc.)
K054	PSE		PSE
K055	ENTER VAR AZ		(Key in using EQN RCL E, RCL N, etc.)
K056	PSE		PSE
K057	INPUT Q	Q?	INPUT Q
K058	RCL Q		
K059	x < 0?		x?0 3
K060	GTO K082		
K061	HMS→		HMS→
K062	STO Q		STO Q
K063	STO (I)		STO (I)
K064	ENTER VAR DIST		(Key in using EQN RCL E, RCL N, etc.)
K065	PSE		PSE
K066	INPUT D	D?	INPUT D
K067	RCL D		
K068	STO (J)		STO (J) [(J) is on the . key]
K069	RCL Q		
K070	SIN		
K071	×		
K072	RCL Q		
K073	COS		
K074	RCL× D		
K075	Σ+		
K076	1		
K077	STO+ K		STO + K
K078	2		
K079	STO+ I		STO + I
K080	STO+ J		STO + J
K081	GTO K055		
K082	Σx^2		SUMS Σx^2
K083	Σy^2		SUMS Σy^2
K084	×		
K085	Σxy		SUMS Σxy

Crandall's Adjustment for a Closed Traverse

Line	Instruction	Display	User Instructions
K086	x^2		x^2
K087	–		
K088	STO D		STO D
K089	Σy		SUMS Σy
K090	Σxy		SUMS Σxy
K091	×		
K092	Σx		SUMS Σx
K093	Σy^2		SUMS Σy^2
K094	×		
K095	–		
K096	RCL÷ D		
K097	STO A		STO A
K098	Σx		SUMS Σx
K099	Σxy		SUMS Σxy
K100	×		
K101	Σy		SUMS Σy
K102	Σx^2		SUMS Σx^2)
K103	×		
K104	–		
K105	RCL÷ D		
K106	STO B		STO B
K107	0		
K108	STO V		STO V
K109	STO L		STO L
K110	MISCLOSURE		(Key in using EQN RCL M, RCL I, etc.)
K111	PSE		PSE
K112	LAT—NORTH		(Key in using EQN RCL L, RCL A, etc.)
K113	PSE		PSE
K114	Σx		SUMS Σx
K115	STO N		STO N
K116	VIEW N		VIEW N
K117	DEP—EAST		(Key in using EQN RCL D, RCL E, etc.)
K118	PSE		PSE
K119	Σy		SUMS Σy
K120	STO E		STO E
K121	VIEW E		VIEW E
K122	Σx		SUMS Σx
K123	x^2		x^2
K124	Σy		SUMS Σy
K125	x^2		x^2
K126	+		
K127	√x		
K128	STO D		STO D
K129	MISC DIST		(Key in using EQN RCL M, RCL I, etc.)
K130	PSE		PSE
K131	VIEW D		VIEW D

Crandall's Adjustment for a Closed Traverse

Line	Instruction	Display	User Instructions
K132	Σy		SUMS Σy
K133	Σx		SUMS Σx
K134	÷		
K135	ATAN		ATAN
K136	x ≥ 0?		x?0 5
K137	GTO K140		
K138	360		
K139	+		
K140	→HMS		→HMS
K141	STO Q		STO Q
K142	MISC AZ		(Key in using EQN RCL M, RCL I, etc.)
K143	PSE		PSE
K144	VIEW Q		VIEW Q
K145	ADJUSTED TRAV		(Key in using EQN RCL A, RCL D, etc.)
K146	PSE		PSE
K147	20		
K148	STO I		STO I
K149	21		
K150	STO J		STO J
K151	RCL (I)		
K152	STO Q		STO Q
K153	RCL (J)		
K154	STO D		STO D
K155	x < > y		
K156	SIN		
K157	×		
K158	RCL Q		
K159	COS		
K160	RCL× D		
K161	RCL× D		
K162	RCL× A		
K163	x < > y		
K164	RCL× D		
K165	RCL× B		
K166	+		
K167	STO C		STO C
K168	STO+ V		STO + V
K169	SIDE AZ		(Key in using EQN RCL S, RCL I, etc.)
K170	PSE		PSE
K171	RCL Q		
K172	→HMS		→HMS
K173	STO Q		STO Q
K174	VIEW Q		VIEW Q
K175	DIST CORRN		(Key in using EQN RCL D, RCL I, etc.)
K176	PSE		PSE
K177	VIEW C		VIEW C

Crandall's Adjustment for a Closed Traverse

Line	Instruction	Display	User Instructions
K178	RCL C		
K179	STO+ D		↱ STO + D
K180	CORRECTED DIST		(Key in using EQN RCL C, RCL O, etc.)
K181	PSE		↱ PSE
K182	VIEW D		↰ VIEW D
K183	RCL Q		
K184	HMS→		↰ HMS→
K185	STO Q		↱ STO Q
K186	SIN		
K187	RCL× C		
K188	RCL Q		
K189	COS		
K190	RCL× C		
K191	Σ+		
K192	1		
K193	STO+ L		↱ STO + L
K194	RCL K		
K195	RCL L		
K196	$x \geq y$?		↰ x?y 5
K197	GTO K202		
K198	2		
K199	STO+ I		↱ STO + I
K200	STO+ J		↱ STO + J
K201	GTO K151		
K202	SUM OF CORRNS		(Key in using EQN RCL S, RCL U, etc.)
K203	PSE		↱ PSE
K204	VIEW V		↰ VIEW V
K205	ADJ MISCLOSURE		(Key in using EQN RCL A, RCL D, etc.)
K206	PSE		↱ PSE
K207	LAT—NORTH		(Key in using EQN RCL L, RCL A, etc.)
K208	PSE		↱ PSE
K209	Σx		↱ SUMS Σx
K210	STO N		↱ STO N
K211	VIEW N		↰ VIEW N
K212	DEP—EAST		(Key in using EQN RCL D, RCL E, etc.)
K213	PSE		↱ PSE
K214	Σy		↱ SUMS Σy
K215	STO E		↱ STO E
K216	VIEW E		↰ VIEW E
K217	Σx		↱ SUMS Σx
K218	x^2		↱ x^2
K219	Σy		↱ SUMS Σy
K220	x^2		↱ x^2
K221	+		
K222	√x		
K223	STO D		↱ STO D

Crandall's Adjustment for a Closed Traverse

Line	Instruction	Display	User Instructions
K224	MISC DIST		(Key in using EQN RCL M, RCL I, etc.)
K225	PSE		↱ PSE
K226	VIEW D		↰ VIEW D
K227	Σy		↱ SUMS Σy
K228	Σx		↱ SUMS Σx
K229	÷		
K230	ATAN		↱ ATAN
K231	x ≥ 0?		↱ x?0 5
K232	GTO K235		
K233	360		
K234	+		
K235	→HMS		↱ →HMS
K236	STO Q		↱ STO Q
K237	MISC AZ		(Key in using EQN RCL M, RCL I, etc.)
K238	PSE		↱ PSE
K239	VIEW Q		↰ VIEW Q
K240	100		
K241	STO I		↱ STO I
K242	0		
K243	STO (I)		↱ STO (I)
K244	20		
K245	RCL I		
K246	x < y?		↰ x?y 3
K247	GTO K251		
K248	1		
K249	STO— I		↱ STO — (I)
K250	GTO K242		
K251	PROGRAM END		(Key in using EQN RCL P, RCL R, etc.)
K252	PSE		↱ PSE
K253	FS? 1		↰ FLAGS 3 1
K254	CF 10		↰ FLAGS 2 .0
K255	RTN		↰ RTN

Notes

(1) The program is designed to run in RPN mode. Its performance in ALG mode is unknown and may produce erroneous results.

(2) The program can work with traverses where all the sides are to be adjusted, where some of the sides are considered 'fixed' (not to be adjusted), and traverses between known points with co-ordinates. The last need the azimuth and distance between the fixed points derived from the co-ordinates, and treated as a fixed side.

(3) The sequence of operations is to enter the fixed sides (those not to be adjusted), and then the variable sides (those to be adjusted). The calculator stores the variable sides and retrieves them from memory to compute the adjusted lengths, and also re-computes the traverse misclosure. The memory limits the traverse to no more than 390 variable sides. If you want to run Crandall's Adjustment on a

traverse of more than 390 variable sides: (a) use a different computational device; (b) be aware that the errors in such a traverse may be too difficult to predict and work with in a meaningful way; (c) use a full least squares adjustment, plus use cross-ties on the ground; and (d) put your analyst on danger money!

(4) Provided that all the fixed sides are entered first, and then all the variable sides, it does not matter in which order the sides within each group are entered.

(5) Azimuths are entered in HP notation, i.e., DDD.MMSS

(6) The misclosure components of the adjusted traverse in X (or E or departure) and Y (or N or latitude) can be displayed by recalling Σy and Σx using the SUMS menu. (Note these are back-to-front.)

(7) This program was inspired by the program developed by the one-and-only Neil H. Bradbury, LS, in January, 1977, for the HP-25 calculator.

(8) The program assumes that the error in the length of each side is proportional to the length of the side, which is the situation with EDM. It will produce somewhat different results to a Crandall's adjustment based on the error increasing as the square root of the distance, which is the more traditional method based on error propagation using a tape.

(9) To indicate the end of the groups of sides of the traverse to be entered at any point, enter a negative value for the azimuth, –1 being easiest.

(10) At the end of the program, Flag 10 is reset to its value before the program started, and the 'indirect' memory is cleared and so de-allocated. This saves memory space and is simply good practice. If you quit before the program comes to the very end, you may leave quite a lot of memory used (from location 20 up to 100).

(11) This program does not use vectors or complex numbers for the misclosure calculation, because the components are required for the adjustment calculations. Consequently, the components are calculated and the vector sum is accumulated through the statistical registers, together with the various data items for the adjustment calculation.

(12) To avoid a problem with storing an azimuth (or distance) of zero, which may cause a memory error in the 'indirect' registers, an arbitrary value is stored in register 100. This should allow a 40-sided traverse. If you want to try a bigger traverse that may have an azimuth of zero in it, change the value of 100 at lines K012 and K240 to a value that is a little larger than twice the number of sides in the traverse plus 20. And make yourself some strong coffee! This will allocate enough 'indirect' memory at the start of the program, and clear and de-allocate it when the program ends. This particular issue came up in discussion with Mr. James Watson of RMIT, and brought to light the potential for a very subtle error in memory allocation, which this addition to the program should allow the user to avoid.

Theory

Crandall's Adjustment was developed by Charles Crandall, and published in 1914. It was the first serious rival to Bowditch's Method of Adjustment. The fundamental concepts were that angle measurement was far better than distance measurement, and that the small random errors in the angles of a traverse could be properly adjusted by simple distribution of the angular misclosure among the traverse's angles. This meant that all the remaining misclosure in the traverse, i.e., all the random errors, could be accounted for by the distances alone, so these were all that was adjusted by Crandall's Adjustment.

Crandall used least squares adjustment by condition equations for the adjustment. By avoiding having to adjust the angles as part of the adjustment proper, the number of conditions was reduced to two, which greatly simplified the calculations.

Angle Adjustment

The angular misclosure of the traverse can be adjusted by any reasonable means. The misclosure can be distributed evenly among all the angles, or larger amounts may be apportioned to angles that can be assumed to be less reliable, perhaps because of short lines or poor lines of sight. However the angles are adjusted, they should be adjusted to bring the misclosure to zero, and then the azimuths of each line calculated.

It is a good idea to run the traverse through one of the closure programs to make sure that there are no gross or systematic errors in the traverse, prior to adjustment. This is because the adjustment process assumes that only random errors are present in the data. ***If this assumption is not correct, the adjustment will produce erroneous results.***

Distance Adjustment

With the angular misclosure adjusted to zero, there are now only two conditions that must be met to have a consistent traverse. (The third condition, that the angular misclosure is zero, has been met by adjusting the angle ahead of this point.) Solution of a 2 × 2 matrix is straightforward, so a full least squares adjustment by condition equations can be performed. This adjustment changes only the lengths of the sides of the traverse. The angles are already adjusted and so held fixed.

If the length of each line in the traverse is M_i, while θ_i is its azimuth, then d_i will be the adjusted length of each line in the traverse, and v_i will be the correction to the measured length of each line to obtain the adjusted length:

$$d_i = M_i + v_i \qquad [1]$$

The conditions to be met can be expressed as:

$$d_1 \cos\theta_1 + d_2 \cos\theta_2 + d_3 \cos\theta_3 + \ldots = 0 \qquad \text{[Sum of latitudes equals zero]}$$

$$d_1 \sin\theta_1 + d_2 \sin\theta_2 + d_3 \sin\theta_3 + \ldots = 0 \qquad \text{[Sum of departures equals zero]}$$

Substituting Equation [1] into the above equations gives:

$$(M_1 + v_1)\cos\theta_1 + (M_2 + v_2)\cos\theta_2 + (M_3 + v_3)\cos\theta_3 + \ldots = 0$$

$$(M_1 + v_1)\sin\theta_1 + (M_2 + v_2)\sin\theta_2 + (M_3 + v_3)\sin\theta_3 + \ldots = 0$$

Re-arranging and gathering terms produces:

$$v_1 \cos\theta_1 + v_2 \cos\theta_2 + v_3 \cos\theta_3 + \ldots + \mathrm{q}_1 = 0$$

$$v_1 \sin\theta_1 + v_2 \sin\theta_2 + v_3 \sin\theta_3 + \ldots + \mathrm{q}_2 = 0 \qquad [2]$$

where $\mathrm{q}_1 = M_1 \cos\theta_1 + M_2 \cos\theta_2 + M_3 \cos\theta_3 + \ldots =$ misclosure in latitude

$\mathrm{q}_2 = M_1 \sin\theta_1 + M_2 \sin\theta_2 + M_3 \sin\theta_3 + \ldots =$ misclosure in departure

Crandall's Adjustment for a Closed Traverse

So q_1 and q_2 are obtained directly from the closure check of the traverse, done after the angles are adjusted and azimuths computed. Representing the departure of each line by D_i, and the latitude of each line by L_i, Equations [2] can be rewritten as:

$$v_1 \cos \frac{L_1}{d_1} + v_2 \cos \frac{L_2}{d_2} + v_3 \cos \frac{L_3}{d_3} + \ldots + q_1 = 0$$

$$v_1 \sin \frac{D_1}{d_1} + v_2 \sin \frac{D_2}{d_2} + v_3 \sin \frac{D_3}{d_3} + \ldots + q_2 = 0 \qquad [3]$$

The theory of least squares now allows the development of what are termed the normal equations (named after the normal distribution, which describes the distribution characteristics of the random errors). The details can be found elsewhere, but basically they are a combination of all the measured data, the precision values of the measurements, and the models that describe how they all fit together, i.e., the model of the conditions to be met (e.g., Equations [2] and [3]).

There are two cases that Crandall considered. The first is the situation where the error in the length of the sides increases in proportion to the square root of the length of the line, i.e., $\sqrt{d}$, which is a good assumption for taped distances. The second case is where the error in distance increases in proportion to the length of the line, which is more the case with short distances measured with rods, but also the case with EDM. This program is based on the second case, as it is more probable that EDM traverses will be used with this program.

If the error in the lengths of the sides of the traverse increases in proportion to the length of each side, then the normal equations for the adjustment are as follows:

$$A\sum_i L_i^2 + B\sum_i L_i D_i + q_1 = 0$$

$$A\sum_i L_i D_i + B\sum_i D_i^2 + q_2 = 0 \qquad [4]$$

Solving for A and B, which are called the correlatives:

$$\mathrm{A} = \frac{q_2\sum_i L_i D_i - q_1\sum_i D_i^2}{\sum_i D_i^2 \sum_i L_i^2 - \left(\sum_i L_i D_i\right)^2}$$

$$\mathrm{B} = \frac{q_1\sum_i L_i D_i - q_2\sum_i L_i^2}{\sum_i D_i^2 \sum_i L_i^2 - \left(\sum_i L_i D_i\right)^2} \qquad [5]$$

the corrections to the line lengths can be calculated using:

$$v_i = L_i \; d_i \; \mathrm{A} + D_i \; d_i \; \mathrm{B} \qquad [6]$$

As an aside, the corrections should all sum to zero if the traverse has no fixed sides, but that can be affected by rounding during the calculations, especially if the sums of the various components are large, so the result may be not quite zero. If the traverse has fixed sides, the sum of the corrections will be significantly different from zero, depending upon the length of the fixed side.

Including Fixed Sides and Points

In the event that the traverse includes more than one 'fixed' point (usually the starting point in calculations), Crandall's Method can still be used. A 'fixed' point means a point that has known co-ordinates, usually from a more precise prior survey, and will not be subject to adjustment in this program. This situation arises when a traverse connects to more than one known point.

If there are two fixed points, and these form the end points of one line of the traverse, then the traverse calculations are done in the normal way, except that the L_i, D_i, and L_iD_i values for the fixed line are not calculated and not included in the summations to solve the correlatives. This is the case in this program.

If there are more than two fixed points, or the fixed points are not adjacent, the traverse must be split into sections between fixed points, and the sections adjusted independently. This is not the ideal solution, as a full least squares adjustment should be used in this situation.

If the traverse runs between two known points, then the line (vector) between the two known points (computed from the co-ordinates of the two points) can be considered another side in the traverse. Computing around the traverse, the misclosure will reflect how well the traverse agrees with this computed vector. To adjust the traverse, don't include the L_i, D_i, and L_iD_i values for this line in the summations, and all the corrections will occur in the measured traverse lines.

For this program, any fixed lines should be worked out ahead of time, including the line between known end-points of the traverse (if applicable). The first part of the program allows the user to enter these sides, and they do not add to the data that sets up the adjustment. The second part of the program allows the user to enter the sides that do add to the adjustment. The final part of the program produces the results.

Sample Computations and Running the Program

1. All sides to be adjusted

The traverse to be adjusted is as follows:

Azimuth	Distance
0° 12'	156.41
90° 00'	211.65
165° 49'	173.82
250° 55'	176.60
308° 30'	112.26

Crandall's Adjustment for a Closed Traverse

The initial misclosure is:	DE	=	0.036
	DN	=	0.033
	Misclosure Length	=	0.049
	Misclosure Azimuth	=	47° 24' 15"

Running the Program

Press XEQ K and then ENTER to start the program.

The program briefly displays CRANDALL ADJ, then briefly displays ENTER FIX SIDE to indicate that the fixed sides should be entered first.

The program then prompts ENTER FIX AZ, to indicate that the azimuth of a fixed side should be entered in DDD.MMSS (HP) notation. The prompt is to enter Q?.

As there are no fixed sides in this case, key in 1 then press the +/– key, and press R/S.

The calculator shows ENTER VAR SIDE to indicate that the sides to be varied (adjusted) should be entered now.

The calculator then prompts ENTER VAR AZ to indicate that the azimuth of a variable sides should be entered in DDD.MMSS (HP) notation. The prompts to enter is Q?.

Key in the azimuth of the first side, 0.12, then press R/S.

The calculator prompts with ENTER VAR DIST, to indicate that the length of that variable side should be entered. The prompt is to enter D?.

Key in the distance 156.41, then press R/S.

The calculator prompts with ENTER VAR AZ, then the Q? prompt.

Key in the next azimuth, 90.0, and press R/S.

The calculator prompts with ENTER VAR DIST, then the D? prompt.

Key in the distance, 211.65, and press R/S.

The calculator prompts with ENTER VAR AZ, then the Q? prompt.

Key in the next azimuth, 165.49, and press R/S.

The calculator prompts with ENTER VAR DIST, then the D? prompt.

Key in the distance, 173.82, and press R/S.

The calculator prompts with ENTER VAR AZ, then the Q? prompt.

Key in the next azimuth, 250.55, and press R/S.

The calculator prompts with ENTER VAR DIST, then the D? prompt.

Key in the distance, 176.6, and press R/S.

The calculator prompts with ENTER VAR AZ, then the Q? prompt.

Key in the next azimuth, 308.3, and press R/S.

The calculator prompts with ENTER VAR DIST, then the D? prompt.

Key in the distance, 112.26, and press R/S.

The calculator prompts with ENTER VAR AZ, then the Q? prompt.

Crandall's Adjustment for a Closed Traverse

As this is the last variable side, key in a negative azimuth, –1, and press R/S.

[Note that the Q and D prompts contain the previous entries, albeit the Q value is in decimal degrees, which can help you keep track of where you are when entering sides.]

The calculator briefly displays MISCLOSURE, then briefly displays LAT–NORTH, then stops and displays:

N=
0.0330

which is the misclosure component in the latitude, or in the north-south direction. Press R/S to continue. The calculator briefly displays DEP–EAST, then stops and displays:

E=
0.0359

which is the misclosure in the departure or east-west direction. Press R/S to continue. The calculator briefly MISC DIST, then stops and displays:

D=
0.0488

which is the length of the misclosure vector. Press R/S to continue. The calculator briefly displays MISC AZ, then stops and displays:

Q=
47.2415

which is the azimuth of the misclosure vector, in HP notation (DDD.MMSS). Press R/S to continue.

The calculator briefly displays ADJUSTED TRAV and then proceeds around a loop displaying, in turn, the azimuth of the side, the correction to that side's distance, and the corrected distance for the side. The prompts before each value are SIDE AZ and Q= for the traverse side's azimuth (which is displayed in HP notation, DDD.MMSS), DIST CORRN and C= for the correction to the distance, and CORRECTED DIST and D= for the corrected side distance.

Pressing R/S after each value is displayed will cause the calculator to move on to the next value. The results are tabulated below. The sequence is moving along the rows from left-to-right, and then down. Note that the azimuth of the sides are unchanged: only the side lengths have been adjusted.

Side Azimuth	Distance Correction	Corrected Distance
0° 12' 00"	–0.0139	156.3961
90° 00' 00"	–0.0207	211.6293
165° 49' 00"	0.0132	173.8332
250° 55' 00"	0.0194	176.6194
308° 30' 00"	0.0001	112.2601

Crandall's Adjustment for a Closed Traverse

After the last corrected side distance has been displayed, the calculator then briefly displays SUM OF CORRNS, which is for the total of the corrections. The calculator stops and displays:

V=
–0.0019

which is the sum of all the distance corrections. This is pretty close to zero. Press R/S to continue. The calculator briefly displays ADJ MISCLOSURE, then LAT–NORTH briefly, then stops and displays:

N=
–9.0000E –16

which indicates that the misclosure of the adjusted traverse in the north-south direction, or in the latitudes, is exceedingly small. Press R/S to continue, and the calculator briefly displays DEP–EAST, then stops and displays:

E=
–1.1030E –13

which is the misclosure adjusted traverse in the east-west or departure direction, and is also very, very small. Press R/S to continue and the calculator displays MISC DIST briefly, then stops and displays:

D=
1.1030E –13

which is the length of the misclosure vector of the adjusted traverse which is also exceedingly tiny. Press R/S to continue, and the calculator briefly displays MISC AZ, then stops and displays:

Q=
89.3157

which is the azimuth of the misclosure vector of the adjusted traverse, in HP notation (DDD.MMSS). Press R/S to continue.

The calculator then shows RUNNING for several seconds as it clears the indirect memory, and then briefly displays PROGRAM END, and terminates.

You can see that it is handy to have the traverse data nicely tabulated, to make getting it into and out of the calculator quick, simple and error free.

2. One fixed side, a traverse between known points

For this traverse, one side is considered fixed, because the traverse is between two fixed points. The azimuth and distance between the two points are deduced from co-ordinates (using a different program), and are 59° 04' 33" for 1995.78. The angular misclosure is adjusted out of the traverse angles, and the azimuths of the sides calculated, on the same azimuth datum as the fixed side, i.e., the same co-ordinate system.

The program is started and the fixed side entered: 59.0433 and 1995.78.

Crandall's Adjustment for a Closed Traverse

As this is the only fixed side, the next azimuth is entered as –1, and the calculator moves to collect the data for the variable sides. These are as follows:

Azimuth	Distance
349° 47' 19"	429.20
109° 22' 12"	476.00
202° 58' 13"	1362.00
271° 07' 59"	210.10
259° 42' 45"	164.00
295° 34' 53"	319.80
237° 45' 39"	499.95
283° 57' 32"	482.10

The initial misclosure is: DE = 2.910

DN = –1.290

Misclosure Length = 3.184

Misclosure Azimuth = 293° 54' 23"

These are entered into the calculator as prompted. Once they have all been entered, an azimuth of –1 is entered, and the calculator shows the above misclosure data. Once this part is done, the calculator works through the adjustment of the sides, presenting the adjusted sides. The adjusted side values are:

Azimuth	Distance Correction	Distance
349° 47' 19"	0.4683	429.668
109° 22' 12"	–0.9666	475.033
202° 58' 13"	–0.4317	1361.568
271° 07' 59"	0.1719	210.272
259° 42' 45"	0.0930	164.093
295° 34' 53"	0.4393	320.239
237° 45' 39"	0.5639	500.514
283° 57' 32"	0.9761	483.076

Pressing R/S again, the calculator displays 1.314 as the sum of the corrections.

The final adjusted misclosure is: DE = 1.600E –11

DN = 2.500E –11

Misclosure Length = 2.968E –11

Misclosure Azimuth = 132° 37' 09"

The misclosure is now very much smaller than the finest level of distance measurement.

Crandall's Adjustment for a Closed Traverse

Comments

An important point to note about Crandall's Adjustment is that because all the corrections are placed in the distances, any non-random errors (e.g., gross errors) in the angles will produce very large corrections in the distances. This is an indication that there may be gross errors in the angles, and this should be checked. Similarly, gross errors in the distances will also produce some large corrections.

In the second example, the original lengths were in many cases almost approximate in their apparent precision, while the misclosure was quite large, being about 1: 1,900. This led to the large corrections in the distances seen in the results.

Note also that in the second example, with the fixed side, the sum of the corrections was 1.314, well away from the zero. This is characteristic of adjusting traverses with fixed sides.

Storage Registers Used

A Correlative A

B Correlative B

C Correction to side length

D Current side length, intermediate result in adjustment calculation, adjusted length, misclosure length

E Misclosure in departure, or the east-west direction.

I Used to address the indirect registers, via (I)

J Used to address the indirect registers, via (J)

K Counter for the number of sides

L Counter for the number of sides

N Misclosure in latitude or the north-south direction

Q Current side azimuth, azimuth of misclosure

V Sum of corrections to the lengths

Statistical Registers:

Σx = Sum of the latitudes

Σy = Sum of the departures

Σx^2 = Sum of the squares of the latitudes of the sides to be adjusted

Σy^2 = Sum of the squares of the departures of the sides to be adjusted

Σxy = Sum of the products of the latitudes and departures of the sides to be adjusted

Labels Used

Label **K** Length = 1058 Checksum = 3D86

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computations to check proper operation after entry.

Crandall's Adjustment for a Closed Traverse

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Reference

CRANDALL, Charles L., 1914. *Geodesy and Least Squares*. New York : John Wiley and Sons.

HP-35s Calculator Program   Closure   7A

Traverse Program using Latitude and Longitude and the Gauss Mid-Latitude Formulae

Programmer: Dr. Bill Hazelton

Date: March, 2008.   **Version**: 1.0

Line	Instruction	Display	User Programming Instructions
J001	LBL J		↱ LBL J
J002	CLSTK		↱ CLEAR 5
J003	FS? 10		↰ FLAGS 3 .0
J004	GTO J008		
J005	SF 1		↰ FLAGS 1 1
J006	SF 10		↰ FLAGS 1 .0
J007	GTO J009		
J008	CF 1		↰ FLAGS 2 1
J009	GAUSS M–L TRAV		(Key in using EQN RCL G, RCL A, etc.)
J010	PSE		↱ PSE
J011	6378137		
J012	STO A		↱ STO A
J013	6.699438 E–3		
J014	STO E		↱ STO E
J015	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
J016	PSE		↱ PSE
J017	INPUT A		↰ INPUT A
J018	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
J019	PSE		↱ PSE
J020	INPUT E		↰ INPUT E
J021	1		
J022	STO M		↱ STO M
J023	CHOOSE UNITS		(Key in using EQN RCL C, RCL H, etc.)
J024	PSE		↱ PSE
J025	FEET–METERS		(Key in using EQN RCL F, RCL E, etc.)
J026	PSE		↱ PSE
J027	INPUT M		↰ INPUT M
J028	0		
J029	STO H		↱ STO H
J030	STO F		↱ STO F
J031	STO L		↱ STO L
J032	AV HT ALL LINES		(Key in using EQN RCL A, RCL V, etc.)
J033	PSE		↱ PSE
J034	ENTER HT		(Key in using EQN RCL E, RCL N, etc.)
J035	PSE		↱ PSE
J036	INPUT H		↰ INPUT H
J037	START POSN		(Key in using EQN RCL S, RCL T, etc.)
J038	PSE		↱ PSE
J039	ENTER LAT		(Key in using EQN RCL E, RCL N, etc.)

Line	Instruction	Display	User Programming Instructions
J040	PSE		PSE
J041	INPUT F		INPUT F
J042	ENTER LONG		(Key in using EQN RCL E, RCL N, etc.)
J043	PSE		PSE
J044	INPUT L		INPUT L
J045	RCL F		
J046	HMS→		HMS→
J047	STO F		STO F
J048	STO Y		STO Y
J049	RCL L		
J050	HMS→		HMS→
J051	STO L		STO L
J052	STO X		STO X
J053	1		
J054	RCL H		
J055	RCL× M		
J056	1.571 E–7		
J057	×		
J058	–		
J059	STO H		STO H
J060	ENTER AZIMUTH		(Key in using EQN RCL E, RCL N, etc.)
J061	PSE		PSE
J062	INPUT Z		INPUT Z
J063	ENTER DISTANCE		(Key in using EQN RCL E, RCL N, etc.)
J064	PSE		PSE
J065	INPUT D		INPUT D
J066	RCL Z		
J067	HMS→		HMS→
J068	STO Z		STO Z
J069	RCL M		
J070	STO× D		STO × D
J071	RCL H		
J072	STO× D		STO × D
J073	RCL Z		
J074	SIN		
J075	RCL× D		
J076	RCL Z		
J077	COS		
J078	RCL× D		
J079	RCL÷ A		
J080	→ DEG		→DEG
J081	STO S		STO S
J082	2		
J083	÷		
J084	STO B		STO B
J085	x <> y		

Line	Instruction	Display	User Programming Instructions
J086	RCL F		
J087	RCL+ B		
J088	COS		
J089	÷		
J090	RCL÷ A		
J091	→ DEG		→DEG
J092	STO T		STO T
J093	2		
J094	÷		
J095	STO C		STO C
J096	1.004		
J097	STO Q		STO Q
J098	RCL F		
J099	RCL+ B		
J100	SIN		
J101	RCL C		
J102	TAN		
J103	×		
J104	RCL B		
J105	COS		
J106	÷		
J107	ATAN		ATAN
J108	STO G		STO G
J109	1		
J110	RCL– E		
J111	RCL× A		
J112	RCL F		
J113	RCL+ B		
J114	SIN		
J115	RCL× E		
J116	1		
J117	x <> y		
J118	–		
J119	1.5		
J120	y^x		
J121	÷		
J122	STO R		STO R
J123	RCL A		
J124	1		
J125	RCL F		
J126	RCL+ B		
J127	SIN		
J128	RCL× E		
J129	–		
J130	√x		
J131	÷		

Traverse Closure Using Latitude and Longitude

Line	Instruction	Display	User Programming Instructions
J132	STO N		↱ STO N
J133	RCL D		
J134	RCL÷ R		
J135	RCL Z		
J136	RCL+ G		
J137	COS		
J138	×		
J139	→ DEG		↱ →DEG
J140	STO S		↱ STO S
J141	RCL Z		
J142	RCL+ G		
J143	SIN		
J144	RCL× D		
J145	RCL÷ N		
J146	RCL F		
J147	RCL+ B		
J148	COS		
J149	÷		
J150	→ DEG		↱ →DEG
J151	STO T		↱ STO T
J152	RCL S		
J153	2		
J154	÷		
J155	STO B		↱ STO B
J156	RCL T		
J157	2		
J158	÷		
J159	STO C		↱ STO C
J160	ISG Q		↰ ISG Q
J161	GTO J098		
J162	RCL S		
J163	STO+ F		↱ STO + F
J164	RCL T		
J165	STO+ L		↱ STO + L
J166	RCL F		
J167	→ HMS		↱ →HMS
J168	STO F		↱ STO F
J169	CURRENT POINT		(Key in using EQN RCL C, RCL U, etc.)
J170	PSE		↱ PSE
J171	LATITUDE		(Key in using EQN RCL L, RCL A, etc.)
J172	PSE		↱ PSE
J173	VIEW F		↰ VIEW F
J174	RCL L		
J175	→ HMS		↱ →HMS
J176	STO L		↱ STO L
J177	LONGITUDE		(Key in using EQN RCL L, RCL O, etc.)

Line	Instruction	Display	User Programming Instructions
J178	PSE		PSE
J179	VIEW L		VIEW L
J180	RCL Z		
J181	RCL+ G		
J182	RCL+ G		
J183	→ HMS		→HMS
J184	STO Q		STO Q
J185	FWD AZIMUTH		(Key in using EQN RCL F, RCL W, etc.)
J186	PSE		PSE
J187	VIEW Q		VIEW Q
J188	RCL F		
J189	HMS→		HMS→
J190	STO F		STO F
J191	RCL L		
J192	HMS→		HMS→
J193	STO L		STO L
J194	NEW LINE (0–1)		(Key in using EQN RCL N, RCL E, etc.)
J195	PSE		PSE
J196	1		
J197	STO J		STO J
J198	INPUT J		INPUT J
J199	RCL J		
J200	x > 0?		x ? 0 4
J201	GTO J060		
J202	RCL F		
J203	RCL– Y		
J204	→ HMS		→HMS
J205	STO B		STO B
J206	RCL L		
J207	RCL– X		
J208	→ HMS		→HMS
J209	STO C		STO C
J210	MISCLOSURE		(Key in using EQN RCL M, RCL I, etc.)
J211	PSE		PSE
J212	MISC LATITUDE		(Key in using EQN RCL M, RCL I, etc.)
J213	PSE		PSE
J214	VIEW B		VIEW B
J215	MISC LONG		(Key in using EQN RCL M, RCL I, etc.)
J216	PSE		PSE
J217	VIEW C		VIEW C
J218	PROGRAM END		(Key in using EQN RCL P, RCL R, etc.)
J219	PSE		PSE
J220	FS? 1		FLAGS 3 1
J221	CF 10		FLAGS 2 .0
J222	RTN		RTN

Notes

(1) Simple computation of co-ordinates around a traverse, together with a simple computations of misclosure, where the traverse point locations are latitudes and longitudes, and the lines have their starting azimuth and linear distance available.

(2) Brief prompts are provided before each requirement for data entry, as well as before results are displayed. Each prompt shows for about 1 second, and is then replaced by the value or request for input.

(3) Co-ordinates of the traverse points (latitude and longitude) are not stored, and so must be written down to record them.

(4) Angles, including latitudes, longitudes and azimuths, are entered and displayed in HP notation, i.e., DDD.MMSS. Internal storage of angles and azimuths is in decimal degrees.

(5) Latitudes are positive North of the Equator, negative South of the Equator. Longitudes are positive to the East of the Greenwich meridian, negative to the West of the Greenwich meridian. This means that in the US, latitudes are positive and longitudes are negative.

(5) The program computes the latitude and longitude of the next traverse point from the azimuth and distance of the line from the current point. This is the classical 'forward' line computation problem in geodesy. The program uses the Gauss Mid-Latitude Formulae for the calculation, iterating each line four times. The program can also be used for stand-alone Gauss Mid-Latitude computations.

(6) The forward azimuth of the line at the end of each line is displayed. This allows the user to compute the azimuth of the next line, using the back azimuth of the current line and the angle measured at the traverse point, if this is required.

(7) The Gauss Mid-Latitude Formulae take care of any spherical excess in the figure, leaving the measured angle misclosure in the resulting azimuths.

(8) The formulae are designed to work with the WGS84/NAD83/GRS80 ellipsoid. If a different ellipsoid is required, the a and e^2 values can be changed at the start of the program. If computations on a spherical figure are required, enter the required radius for a, and set e^2 to zero. Values for some other ellipsoids are given later in this document.

(9) Users can enter an average height for lines, to allow lines to be reduced to the ellipsoid, but this is an average value for the entire traverse. If the lines have already been reduced to the ellipsoid, or this is not required, enter zero for the height.

(10) Users can elect to enter distances in feet by entering 0.3048 when prompted for FEET–METERS early in the program. To use meters, retain a value of 1, the program default. Other units can be used, if a conversion factor to meters is entered instead of the 0.3048. Long lines may need 0.30480061 for US Survey feet, while 0.3048 is used for International feet and approximate work in feet.

(11) The resulting misclosure is expressed in angular terms, in HP notation, and is the amount by which the final latitude and longitude miss the starting values. To convert these values to meters, multiply the number of seconds of latitude by 30, and the number of seconds of longitude by 30 cos ϕ. To get the values in feet multiply by 100 and 100 cos ϕ, respectively, instead. Note that these will be approximate.

Theory

This program uses the Gauss Mid-Latitude formulae to calculate the position of the point at the end of a line, given the starting position (in latitude, ϕ, and longitude, λ), the forward azimuth at the known point, and the distance (in either feet or meters). This is the classical 'forward' problem of geodetic line computation.

For the forward solution, the Gauss Mid-Latitude formulae require iteration to reach a solution, but are the simplest and quickest geodetic formulae for this type of task. The formulae are accurate to better than 0.001" in latitude and longitude (0.3 m, 0.1 ft), for worst-case lines up to 20 miles (32 km). If greater precision is required, use a different geodetic long-line formula (e.g., Robbins' or Rudoe's formulae).

Within the program, when a line's azimuth, θ, and distance, d, are first entered, they are converted to an initial, approximate latitude and longitude differences ($\Delta\phi_0$ and $\Delta\lambda_0$, respectively), using:

$$\Delta\phi_0 = \frac{d \cos\theta}{a} \qquad \Delta\lambda_0 = \frac{d \sin\theta}{a \cos\phi}$$

The mid-latitude of the line, ϕ_m, is computed using: $\phi_m = \phi + \frac{\Delta\phi}{2}$ [1]

The change in azimuth over the length of the line, $\Delta\theta$, is computed using:

$$\tan\frac{\Delta\theta}{2} = \tan\frac{\Delta\lambda}{2} \sin\phi_m \sec\frac{\Delta\phi}{2} \qquad [2]$$

The radii of curvature in the meridian and prime vertical at the mid-point of the line, ρ_m and ν_m, respectively, are calculated, using:

$$\rho_m = \frac{a(1-e^2)}{(1-e^2\sin^2\phi_m)^{\frac{3}{2}}} \qquad [3]$$

$$\nu_m = \frac{a}{\sqrt{(1-e^2\sin^2\phi_m)}} \qquad [4]$$

The differences in latitude and longitude are then calculated, using:

$$\Delta\phi = \frac{d}{\rho_m}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \qquad [5]$$

$$\Delta\lambda = \frac{d}{\nu_m}\sin\left(\theta + \frac{\Delta\theta}{2}\right)\sec\phi_m \qquad [6]$$

The values from equations [5] and [6] are returned to equation [1], and the process re-iterated until the changes in latitude and longitude are too small to worry about. In most cases, three iterations are sufficient, but the program uses four iterations, just to be sure.

Distances are converted to ellipsoidal distances using the average height (h_m) for the region (or single line) entered. Entering zero for the height means the height scale factor has no effect on line length. The formula used is as follows, which is good to about 1 in 10,000 if the height is good to within 60 meters. The ellipsoidal distance is equal to the 'horizontal' distance times the height scale factor.

$$\text{Height Scale Factor} = 1 - (h_m \times 0.1571 \times 10^{-6})$$

At the end of each line, the program displays the latitude and longitude of the end point, as well as the forward azimuth of the line at this point. For most lines of any significant length, this will differ from the forward azimuth at the start of the line. By converting this forward bearing to a back bearing (by adding or subtracting 180°), an angle measured at the end point can be used to obtain the forward azimuth of the next line. This is done manually by the user, and is not included in the program, as it is not something that will be needed by all users.

The program stores the initial point values, so that a comparison can be made at the end of a traverse, if desired. The difference is calculated and shown to the user.

Sample Computation

Traverse Data and Results

Point	Line	Azimuth	Distance	Latitude	Longitude
A				40° 02' 25".000	–83° 01' 25".000
	A–B	47° 51' 27"	14,302.785		
B				40° 07' 35".189	–82° 53' 57".427
	B–C	140° 32' 56"	12,821.076		
C				40° 02' 14".812	–82° 48' 14".056
	C–D	235° 28' 29"	15,093.269		
D				39° 57' 37".772	–82° 56' 57".579
	D–E	274° 28' 12"	6,394.974		
E				39° 57' 53".807	–83° 01' 26".012
	E–A	0° 09' 50"	8,383.815		
A				40° 02' 25".001	–83° 01' 25".001

Misclosure Latitude (ϕ) = 0° 00' 00".001 = 0".0007 = 0.021 m

Longitude (λ) = –0° 00' 00".001 = –0".0008 = –0.018 m

The angular misclosure around this figure was initially 2", of which about 1" was spherical excess. The linear misclosure when computed by other means is very close to the figures above.

Note that the misclosure is determined from very small differences at the least significant end of a long number, and so is affected by the limitations in the calculator's internal precision. Calculation of the same traverse using different equipment (e.g., a spreadsheet) should give the same results for locations and azimuths, but there may be some small differences in the misclosure.

Storage Registers Used

A	a = semi-major axis of ellipsoid = 6378137 m for WGS84/NAD83/GRS80
B	$\Delta\phi/2$ = half the latitude difference for the line
C	$\Delta\lambda/2$ = half the longitude difference for the line
D	Distance, i.e., length of the line
E	e^2 = eccentricity of ellipsoid = 0.006699438 for WGS84/NAD83/GRS80
F	ϕ, latitude of starting point of each line
G	$\Delta\theta/2$ = half the azimuth change for the line
H	Height above ellipsoid, then the height-scale factor for line lengths. By default, height = 0
J	Response variable for whether another line is to be processed
L	λ, longitude of starting point of each line
M	unit to meters conversion factor; by default 1.0 for meters
N	ν_m = radius of curvature of the ellipsoid in the prime vertical at the mid-point of the line
Q	counter for calculation loop (1.004 by default), then forward azimuth of line
R	ρ_m = radius of curvature of the ellipsoid in the meridian at the mid-point of the line
S	$\Delta\phi$, the difference in latitude for the line
T	$\Delta\lambda$, the difference in longitude for the line
X	λ_0, longitude of the initial point of the traverse
Y	ϕ_0, the latitude of the initial point of the traverse
Z	θ, azimuth at the start of the line

Labels Used

Label **J**	Length = 959	Checksum = D8B4

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after program entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Ellipsoidal Values

WGS84/NAD83/GRS80	a = 6 378 137 m	e^2 = 0.006 699 438 00
Clark 1866	a = 6 378 206.4 m	e^2 = 0.006 768 658
WGS72	a = 6 378 135 m	e^2 = 0.006 694 317 778
ANS (Australian)	a = 6 378 160 m	e^2 = 0.006 694 541 855

Running the Program

To start the program, press XEQ J, then press ENTER.

The calculator briefly displays GAUSS M–L TRAV, then briefly displays CHECK–ENTER A.

The calculator stops and prompts with A?

Key in a value for ellipsoid semi-major axis, or ignore to retain default value (the WGS84 value).

Press R/S to continue.

The calculator displays CHECK–ENTER E briefly, then stops and prompts with E?

Key in a value for ellipsoid eccentricity, or ignore to retain default value (the WGS84 value).

Press R/S to continue.

The calculator briefly displays CHOOSE UNITS, the briefly displays FEET-METERS, then stops and prompts with M?

Enter unit conversion value, or ignore to retain setting for distances in meters (value of 1). Enter 0.3048 for International feet; 0.30480061 for US Survey feet; 0.201168 for chains, etc.

Press R/S to continue.

The calculator briefly displays AV HT ALL LINES (although the S won't be visible), then briefly displays ENTER HT, then stops and prompts with H?

Enter average height above the ellipsoid for all lines to be processed. Ignore to retail default value of zero. Enter the height in the units you selected at the FEET–METERS prompt.

Press R/S to continue.

The calculator briefly displays START POSITION, then briefly displays ENTER LAT, then stops and prompts with F?

Enter the latitude (ϕ) of the starting point, in degrees, minutes and seconds, in HP notation (D.MMSSsss). Remember to include a negative sign, if in the southern hemisphere.

Press R/S to continue.

The calculator briefly displays ENTER LONG, then stops and prompts with L?

Enter the longitude (λ) of the starting point, in degrees, minutes and seconds, in HP notation (D.MMSSsss). Remember to include a negative sign, if in the western hemisphere.

Press R/S to continue.

Top of Loop Point

The calculator briefly displays ENTER AZIMUTH, then stops and prompts with Z?

Enter azimuth of the line at the starting point (θ) in HP notation.

Press R/S to continue.

Traverse Closure Using Latitude and Longitude

The calculator briefly displays ENTER DISTANCE, then stops and prompts with D?

Enter the length of the line in the units previous selected.

Press R/S to continue.

The calculator displays RUNNING for a while. Then the calculator briefly displays CURRENT POINT, the briefly displays LATITUDE, then stops and displays F= and the latitude of the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator displays LONGITUDE briefly, then stops and displays L= and the longitude of the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator displays FWD AZIMUTH briefly, then stops and displays Q= and the forward azimuth of the line at the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator briefly displays NEW LINE (0–1), then stops and prompts with J? and the default value of 1. To go on to do the next line in the traverse, press R/S and the program will go to the *Top of Loop Point*, above. If all the sides have been entered, key in 0 and press R/S to calculate the misclosure.

The calculator displays MISCLOSURE briefly, the briefly displays MISC LATITUDE, then stops and displays B= and the misclosure in latitude (difference between start and end latitudes), displayed in HP notation.

Press R/S to continue.

The calculator briefly displays MISC LONGITUDE, then stops and shows C= and the misclosure in longitude (difference between start and end longitudes), displayed in HP notation.

Pressing R/S again will reset the flags, briefly display PROGRAM END, and end the program. If the program was called from another location, control will return to that point.

The misclosure in latitude will remain in the Y register, and the misclosure in longitude will remain in the X register, on the screen of the calculator. They can now be converted to whatever units interest you, after having converted them to decimal degrees using the HMS→ function.

Note that this misclosure is being determined from small differences at the least significant end of the calculator's storage capability, and so will be approximate at best. It will give an idea of the degree of magnitude of the misclosure, rather than an exact amount.

HP-35s Calculator Program Closure 7A

Traverse Program using Latitude and Longitude and the Gauss Mid-Latitude Formulae

Programmer: Dr. Bill Hazelton

Date: March, 2008. **Version**: 1.0

Line	Instruction	Display	User Programming Instructions
J001	LBL J		↱ LBL J
J002	CLSTK		↱ CLEAR 5
J003	FS? 10		↰ FLAGS 3 .0
J004	GTO J008		
J005	SF 1		↰ FLAGS 1 1
J006	SF 10		↰ FLAGS 1 .0
J007	GTO J009		
J008	CF 1		↰ FLAGS 2 1
J009	GAUSS M–L TRAV		(Key in using EQN RCL G, RCL A, etc.)
J010	PSE		↱ PSE
J011	6378137		
J012	STO A		↱ STO A
J013	6.699438 E–3		
J014	STO E		↱ STO E
J015	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
J016	PSE		↱ PSE
J017	INPUT A		↰ INPUT A
J018	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
J019	PSE		↱ PSE
J020	INPUT E		↰ INPUT E
J021	1		
J022	STO M		↱ STO M
J023	CHOOSE UNITS		(Key in using EQN RCL C, RCL H, etc.)
J024	PSE		↱ PSE
J025	FEET–METERS		(Key in using EQN RCL F, RCL E, etc.)
J026	PSE		↱ PSE
J027	INPUT M		↰ INPUT M
J028	0		
J029	STO H		↱ STO H
J030	STO F		↱ STO F
J031	STO L		↱ STO L
J032	AV HT ALL LINES		(Key in using EQN RCL A, RCL V, etc.)
J033	PSE		↱ PSE
J034	ENTER HT		(Key in using EQN RCL E, RCL N, etc.)
J035	PSE		↱ PSE
J036	INPUT H		↰ INPUT H
J037	START POSN		(Key in using EQN RCL S, RCL T, etc.)
J038	PSE		↱ PSE
J039	ENTER LAT		(Key in using EQN RCL E, RCL N, etc.)

Line	Instruction	Display	User Programming Instructions
J040	PSE		PSE
J041	INPUT F		INPUT F
J042	ENTER LONG		(Key in using EQN RCL E, RCL N, etc.)
J043	PSE		PSE
J044	INPUT L		INPUT L
J045	RCL F		
J046	HMS→		HMS→
J047	STO F		STO F
J048	STO Y		STO Y
J049	RCL L		
J050	HMS→		HMS→
J051	STO L		STO L
J052	STO X		STO X
J053	1		
J054	RCL H		
J055	RCL× M		
J056	1.571 E–7		
J057	×		
J058	–		
J059	STO H		STO H
J060	ENTER AZIMUTH		(Key in using EQN RCL E, RCL N, etc.)
J061	PSE		PSE
J062	INPUT Z		INPUT Z
J063	ENTER DISTANCE		(Key in using EQN RCL E, RCL N, etc.)
J064	PSE		PSE
J065	INPUT D		INPUT D
J066	RCL Z		
J067	HMS→		HMS→
J068	STO Z		STO Z
J069	RCL M		
J070	STO× D		STO × D
J071	RCL H		
J072	STO× D		STO × D
J073	RCL Z		
J074	SIN		
J075	RCL× D		
J076	RCL Z		
J077	COS		
J078	RCL× D		
J079	RCL÷ A		
J080	→ DEG		→DEG
J081	STO S		STO S
J082	2		
J083	÷		
J084	STO B		STO B
J085	x < > y		

Traverse Closure Using Latitude and Longitude

Line	Instruction	Display	User Programming Instructions
J086	RCL F		
J087	RCL+ B		
J088	COS		
J089	÷		
J090	RCL÷ A		
J091	→ DEG		→DEG
J092	STO T		STO T
J093	2		
J094	÷		
J095	STO C		STO C
J096	1.004		
J097	STO Q		STO Q
J098	RCL F		
J099	RCL+ B		
J100	SIN		
J101	RCL C		
J102	TAN		
J103	×		
J104	RCL B		
J105	COS		
J106	÷		
J107	ATAN		ATAN
J108	STO G		STO G
J109	1		
J110	RCL– E		
J111	RCL× A		
J112	RCL F		
J113	RCL+ B		
J114	SIN		
J115	RCL× E		
J116	1		
J117	x <> y		
J118	–		
J119	1.5		
J120	y^x		
J121	÷		
J122	STO R		STO R
J123	RCL A		
J124	1		
J125	RCL F		
J126	RCL+ B		
J127	SIN		
J128	RCL× E		
J129	–		
J130	√x		
J131	÷		

Line	Instruction	Display	User Programming Instructions
J132	STO N		↱ STO N
J133	RCL D		
J134	RCL÷ R		
J135	RCL Z		
J136	RCL+ G		
J137	COS		
J138	×		
J139	→ DEG		↱ →DEG
J140	STO S		↱ STO S
J141	RCL Z		
J142	RCL+ G		
J143	SIN		
J144	RCL× D		
J145	RCL÷ N		
J146	RCL F		
J147	RCL+ B		
J148	COS		
J149	÷		
J150	→ DEG		↱ →DEG
J151	STO T		↱ STO T
J152	RCL S		
J153	2		
J154	÷		
J155	STO B		↱ STO B
J156	RCL T		
J157	2		
J158	÷		
J159	STO C		↱ STO C
J160	ISG Q		↰ ISG Q
J161	GTO J098		
J162	RCL S		
J163	STO+ F		↱ STO + F
J164	RCL T		
J165	STO+ L		↱ STO + L
J166	RCL F		
J167	→ HMS		↱ →HMS
J168	STO F		↱ STO F
J169	CURRENT POINT		(Key in using EQN RCL C, RCL U, etc.)
J170	PSE		↱ PSE
J171	LATITUDE		(Key in using EQN RCL L, RCL A, etc.)
J172	PSE		↱ PSE
J173	VIEW F		↰ VIEW F
J174	RCL L		
J175	→ HMS		↱ →HMS
J176	STO L		↱ STO L
J177	LONGITUDE		(Key in using EQN RCL L, RCL O, etc.)

Traverse Closure Using Latitude and Longitude

Line	Instruction	Display	User Programming Instructions
J178	PSE		↱ PSE
J179	VIEW L		↰ VIEW L
J180	RCL Z		
J181	RCL+ G		
J182	RCL+ G		
J183	→ HMS		↱ →HMS
J184	STO Q		↱ STO Q
J185	FWD AZIMUTH		(Key in using EQN RCL F, RCL W, etc.)
J186	PSE		↱ PSE
J187	VIEW Q		↰ VIEW Q
J188	RCL F		
J189	HMS→		↰ HMS→
J190	STO F		↱ STO F
J191	RCL L		
J192	HMS→		↰ HMS→
J193	STO L		↱ STO L
J194	NEW LINE (0–1)		(Key in using EQN RCL N, RCL E, etc.)
J195	PSE		↱ PSE
J196	1		
J197	STO J		↱ STO J
J198	INPUT J		↰ INPUT J
J199	RCL J		
J200	x > 0?		↱ x ? 0 4
J201	GTO J060		
J202	RCL F		
J203	RCL– Y		
J204	→ HMS		↱ →HMS
J205	STO B		↱ STO B
J206	RCL L		
J207	RCL– X		
J208	→ HMS		↱ →HMS
J209	STO C		↱ STO C
J210	MISCLOSURE		(Key in using EQN RCL M, RCL I, etc.)
J211	PSE		↱ PSE
J212	MISC LATITUDE		(Key in using EQN RCL M, RCL I, etc.)
J213	PSE		↱ PSE
J214	VIEW B		↰ VIEW B
J215	MISC LONG		(Key in using EQN RCL M, RCL I, etc.)
J216	PSE		↱ PSE
J217	VIEW C		↰ VIEW C
J218	PROGRAM END		(Key in using EQN RCL P, RCL R, etc.)
J219	PSE		↱ PSE
J220	FS? 1		↰ FLAGS 3 1
J221	CF 10		↰ FLAGS 2 .0
J222	RTN		↰ RTN

Traverse Closure Using Latitude and Longitude

Notes

(1) Simple computation of co-ordinates around a traverse, together with a simple computations of misclosure, where the traverse point locations are latitudes and longitudes, and the lines have their starting azimuth and linear distance available.

(2) Brief prompts are provided before each requirement for data entry, as well as before results are displayed. Each prompt shows for about 1 second, and is then replaced by the value or request for input.

(3) Co-ordinates of the traverse points (latitude and longitude) are not stored, and so must be written down to record them.

(4) Angles, including latitudes, longitudes and azimuths, are entered and displayed in HP notation, i.e., DDD.MMSS. Internal storage of angles and azimuths is in decimal degrees.

(5) Latitudes are positive North of the Equator, negative South of the Equator. Longitudes are positive to the East of the Greenwich meridian, negative to the West of the Greenwich meridian. This means that in the US, latitudes are positive and longitudes are negative.

(5) The program computes the latitude and longitude of the next traverse point from the azimuth and distance of the line from the current point. This is the classical 'forward' line computation problem in geodesy. The program uses the Gauss Mid-Latitude Formulae for the calculation, iterating each line four times. The program can also be used for stand-alone Gauss Mid-Latitude computations.

(6) The forward azimuth of the line at the end of each line is displayed. This allows the user to compute the azimuth of the next line, using the back azimuth of the current line and the angle measured at the traverse point, if this is required.

(7) The Gauss Mid-Latitude Formulae take care of any spherical excess in the figure, leaving the measured angle misclosure in the resulting azimuths.

(8) The formulae are designed to work with the WGS84/NAD83/GRS80 ellipsoid. If a different ellipsoid is required, the a and e^2 values can be changed at the start of the program. If computations on a spherical figure are required, enter the required radius for a, and set e^2 to zero. Values for some other ellipsoids are given later in this document.

(9) Users can enter an average height for lines, to allow lines to be reduced to the ellipsoid, but this is an average value for the entire traverse. If the lines have already been reduced to the ellipsoid, or this is not required, enter zero for the height.

(10) Users can elect to enter distances in feet by entering 0.3048 when prompted for FEET–METERS early in the program. To use meters, retain a value of 1, the program default. Other units can be used, if a conversion factor to meters is entered instead of the 0.3048. Long lines may need 0.30480061 for US Survey feet, while 0.3048 is used for International feet and approximate work in feet.

(11) The resulting misclosure is expressed in angular terms, in HP notation, and is the amount by which the final latitude and longitude miss the starting values. To convert these values to meters, multiply the number of seconds of latitude by 30, and the number of seconds of longitude by 30 cos ϕ. To get the values in feet multiply by 100 and 100 cos ϕ, respectively, instead. Note that these will be approximate.

Theory

This program uses the Gauss Mid-Latitude formulae to calculate the position of the point at the end of a line, given the starting position (in latitude, ϕ, and longitude, λ), the forward azimuth at the known point, and the distance (in either feet or meters). This is the classical 'forward' problem of geodetic line computation.

For the forward solution, the Gauss Mid-Latitude formulae require iteration to reach a solution, but are the simplest and quickest geodetic formulae for this type of task. The formulae are accurate to better than 0.001" in latitude and longitude (0.3 m, 0.1 ft), for worst-case lines up to 20 miles (32 km). If greater precision is required, use a different geodetic long-line formula (e.g., Robbins' or Rudoe's formulae).

Within the program, when a line's azimuth, θ, and distance, d, are first entered, they are converted to an initial, approximate latitude and longitude differences ($\Delta\phi_0$ and $\Delta\lambda_0$, respectively), using:

$$\Delta\phi_0 = \frac{d\cos\theta}{a} \qquad \Delta\lambda_0 = \frac{d\sin\theta}{a\cos\phi}$$

The mid-latitude of the line, ϕ_m, is computed using: $\phi_m = \phi + \frac{\Delta\phi}{2}$ [1]

The change in azimuth over the length of the line, $\Delta\theta$, is computed using:

$$\tan\frac{\Delta\theta}{2} = \tan\frac{\Delta\lambda}{2}\sin\phi_m\sec\frac{\Delta\phi}{2} \qquad [2]$$

The radii of curvature in the meridian and prime vertical at the mid-point of the line, ρ_m and ν_m, respectively, are calculated, using:

$$\rho_m = \frac{a(1-e^2)}{(1-e^2\sin^2\phi_m)^{\frac{3}{2}}} \qquad [3]$$

$$\nu_m = \frac{a}{\sqrt{(1-e^2\sin^2\phi_m)}} \qquad [4]$$

The differences in latitude and longitude are then calculated, using:

$$\Delta\phi = \frac{d}{\rho_m}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \qquad [5]$$

$$\Delta\lambda = \frac{d}{\nu_m}\sin\left(\theta + \frac{\Delta\theta}{2}\right)\sec\phi_m \qquad [6]$$

The values from equations [5] and [6] are returned to equation [1], and the process re-iterated until the changes in latitude and longitude are too small to worry about. In most cases, three iterations are sufficient, but the program uses four iterations, just to be sure.

Traverse Closure Using Latitude and Longitude

Distances are converted to ellipsoidal distances using the average height (h_m) for the region (or single line) entered. Entering zero for the height means the height scale factor has no effect on line length. The formula used is as follows, which is good to about 1 in 10,000 if the height is good to within 60 meters. The ellipsoidal distance is equal to the 'horizontal' distance times the height scale factor.

$$\text{Height Scale Factor} = 1 - (h_m \times 0.1571 \times 10^{-6})$$

At the end of each line, the program displays the latitude and longitude of the end point, as well as the forward azimuth of the line at this point. For most lines of any significant length, this will differ from the forward azimuth at the start of the line. By converting this forward bearing to a back bearing (by adding or subtracting 180°), an angle measured at the end point can be used to obtain the forward azimuth of the next line. This is done manually by the user, and is not included in the program, as it is not something that will be needed by all users.

The program stores the initial point values, so that a comparison can be made at the end of a traverse, if desired. The difference is calculated and shown to the user.

Sample Computation

Traverse Data and Results

Point	Line	Azimuth	Distance	Latitude	Longitude
A				40° 02' 25".000	–83° 01' 25".000
	A–B	47° 51' 27"	14,302.785		
B				40° 07' 35".189	–82° 53' 57".427
	B–C	140° 32' 56"	12,821.076		
C				40° 02' 14".812	–82° 48' 14".056
	C–D	235° 28' 29"	15,093.269		
D				39° 57' 37".772	–82° 56' 57".579
	D–E	274° 28' 12"	6,394.974		
E				39° 57' 53".807	–83° 01' 26".012
	E–A	0° 09' 50"	8,383.815		
A				40° 02' 25".001	–83° 01' 25".001

Misclosure Latitude (ϕ) = 0° 00' 00".001 = 0".0007 = 0.021 m

Longitude (λ) = –0° 00' 00".001 = –0".0008 = –0.018 m

The angular misclosure around this figure was initially 2", of which about 1" was spherical excess. The linear misclosure when computed by other means is very close to the figures above.

Note that the misclosure is determined from very small differences at the least significant end of a long number, and so is affected by the limitations in the calculator's internal precision. Calculation of the same traverse using different equipment (e.g., a spreadsheet) should give the same results for locations and azimuths, but there may be some small differences in the misclosure.

Storage Registers Used

A	a = semi-major axis of ellipsoid = 6378137 m for WGS84/NAD83/GRS80
B	$\Delta\phi/2$ = half the latitude difference for the line
C	$\Delta\lambda/2$ = half the longitude difference for the line
D	Distance, i.e., length of the line
E	e^2 = eccentricity of ellipsoid = 0.006699438 for WGS84/NAD83/GRS80
F	ϕ, latitude of starting point of each line
G	$\Delta\theta/2$ = half the azimuth change for the line
H	Height above ellipsoid, then the height-scale factor for line lengths. By default, height = 0
J	Response variable for whether another line is to be processed
L	λ, longitude of starting point of each line
M	unit to meters conversion factor; by default 1.0 for meters
N	ν_m = radius of curvature of the ellipsoid in the prime vertical at the mid-point of the line
Q	counter for calculation loop (1.004 by default), then forward azimuth of line
R	ρ_m = radius of curvature of the ellipsoid in the meridian at the mid-point of the line
S	$\Delta\phi$, the difference in latitude for the line
T	$\Delta\lambda$, the difference in longitude for the line
X	λ_0, longitude of the initial point of the traverse
Y	ϕ_0, the latitude of the initial point of the traverse
Z	θ, azimuth at the start of the line

Labels Used

Label **J**	Length = 959	Checksum = D8B4

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after program entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Ellipsoidal Values

WGS84/NAD83/GRS80	a = 6 378 137 m	e^2 = 0.006 699 438 00
Clark 1866	a = 6 378 206.4 m	e^2 = 0.006 768 658
WGS72	a = 6 378 135 m	e^2 = 0.006 694 317 778
ANS (Australian)	a = 6 378 160 m	e^2 = 0.006 694 541 855

Running the Program

To start the program, press XEQ J, then press ENTER.

The calculator briefly displays GAUSS M–L TRAV, then briefly displays CHECK–ENTER A.

The calculator stops and prompts with A?

Key in a value for ellipsoid semi-major axis, or ignore to retain default value (the WGS84 value).

Press R/S to continue.

The calculator displays CHECK–ENTER E briefly, then stops and prompts with E?

Key in a value for ellipsoid eccentricity, or ignore to retain default value (the WGS84 value).

Press R/S to continue.

The calculator briefly displays CHOOSE UNITS, the briefly displays FEET-METERS, then stops and prompts with M?

Enter unit conversion value, or ignore to retain setting for distances in meters (value of 1). Enter 0.3048 for International feet; 0.30480061 for US Survey feet; 0.201168 for chains, etc.

Press R/S to continue.

The calculator briefly displays AV HT ALL LINES (although the S won't be visible), then briefly displays ENTER HT, then stops and prompts with H?

Enter average height above the ellipsoid for all lines to be processed. Ignore to retail default value of zero. Enter the height in the units you selected at the FEET–METERS prompt.

Press R/S to continue.

The calculator briefly displays START POSITION, then briefly displays ENTER LAT, then stops and prompts with F?

Enter the latitude (ϕ) of the starting point, in degrees, minutes and seconds, in HP notation (D.MMSSsss). Remember to include a negative sign, if in the southern hemisphere.

Press R/S to continue.

The calculator briefly displays ENTER LONG, then stops and prompts with L?

Enter the longitude (λ) of the starting point, in degrees, minutes and seconds, in HP notation (D.MMSSsss). Remember to include a negative sign, if in the western hemisphere.

Press R/S to continue.

Top of Loop Point

The calculator briefly displays ENTER AZIMUTH, then stops and prompts with Z?

Enter azimuth of the line at the starting point (θ) in HP notation.

Press R/S to continue.

Traverse Closure Using Latitude and Longitude

The calculator briefly displays ENTER DISTANCE, then stops and prompts with D?

Enter the length of the line in the units previous selected.

Press R/S to continue.

The calculator displays RUNNING for a while. Then the calculator briefly displays CURRENT POINT, the briefly displays LATITUDE, then stops and displays F= and the latitude of the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator displays LONGITUDE briefly, then stops and displays L= and the longitude of the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator displays FWD AZIMUTH briefly, then stops and displays Q= and the forward azimuth of the line at the far end of the line just entered, displayed in HP notation.

Press R/S to continue.

The calculator briefly displays NEW LINE (0–1), then stops and prompts with J? and the default value of 1. To go on to do the next line in the traverse, press R/S and the program will go to the *Top of Loop Point*, above. If all the sides have been entered, key in 0 and press R/S to calculate the misclosure.

The calculator displays MISCLOSURE briefly, the briefly displays MISC LATITUDE, then stops and displays B= and the misclosure in latitude (difference between start and end latitudes), displayed in HP notation.

Press R/S to continue.

The calculator briefly displays MISC LONGITUDE, then stops and shows C= and the misclosure in longitude (difference between start and end longitudes), displayed in HP notation.

Pressing R/S again will reset the flags, briefly display PROGRAM END, and end the program. If the program was called from another location, control will return to that point.

The misclosure in latitude will remain in the Y register, and the misclosure in longitude will remain in the X register, on the screen of the calculator. They can now be converted to whatever units interest you, after having converted them to decimal degrees using the HMS→ function.

Note that this misclosure is being determined from small differences at the least significant end of the calculator's storage capability, and so will be approximate at best. It will give an idea of the degree of magnitude of the misclosure, rather than an exact amount.

Traverse Closure with Area Calculation and Co-ordinates and Proportional Misclosure

Programmer: Dr. Bill Hazelton

Date: October, 2010.

Line	Instruction	Display	User Instructions
A001	LBL A		Press XEQ A ENTER
A002	CLSTK		
A003	SF 10		
A004	USE COORDS	USE COORDS	(Key in using EQN RCL U, RCL S, etc.)
A005	PSE		
A006	INPUT C	C?	
A007	RCL C		
A008	x = 0?		
A009	GTO A021		
A010	ENTER N0	ENTER N0	(Key in using EQN RCL E, RCL N, etc.)
A011	PSE		
A012	INPUT N	N?	
A013	ENTER E0	ENTER E0	(Key in using EQN RCL E, RCL N, etc.)
A014	PSE		
A015	INPUT E	E?	
A016	RCL E		
A017	0 **i** 1		(Key in as 0, then **i**, then 1, press ENTER.)
A018	×		
A019	RCL+ N		
A020	STO F		
A021	360		
A022	STO B		
A023	CLSTK		
A024	STO P		
A025	STO Q		
A026	STO M		
A027	STO T		
A028	XEQ V001		
A029	STO R		
A030	STO+ P		
A031	ARG		
A032	RCL P		
A033	ARG		
A034	–		
A035	SIN		
A036	STO S		
A037	RCL R		
A038	ABS		

Traverse Closure, Area Calc., Co-ordinates & Prop.Misc.

A039	STO+ T		
A040	STO× S		
A041	RCL P		
A042	ABS		
A043	STO× S		
A044	RCL S		
A045	2		
A046	÷		
A047	STO+ Q		
A048	1		
A049	STO+ M		
A050	RCL C		
A051	x = 0?		
A052	GTO A074		
A053	RCL P		
A054	RCL+ F		
A055	XEQ X001		
A056	RCL M		
A057	RCL Q		
A058	ABS		
A059	R↓		
A060	STOP		
A061	RCL T		
A062	RCL P		
A063	ABS		
A064	÷		
A065	RCL P		
A066	ARG		
A067	x < 0?		
A068	RCL+ B		
A069	→HMS		
A070	RCL P		
A071	ABS		
A072	STOP		
A073	GTO A028		
A074	RCL Q		
A075	ABS		
A076	RCL P		
A077	ARG		
A078	x < 0?		
A079	RCL+ B		
A080	→HMS		
A081	RCL P		
A082	ABS		
A083	RCL M		
A084	STOP		
A085	GTO A028		

Traverse Closure, Area Calc., Co-ordinates & Prop.Misc.

This version of the program is designed to completely hide the complex number work that the calculator performs to compute the traverse. All values entered and returned are presented in much the same manner as with the HP-33S calculator.

Notes

(1) Set the calculator into DEGREES mode (press MODE 1) before starting.

(2) This is a general traverse closure program that computes the azimuth and distance, plus area, to each point around the traverse, together with co-ordinates, if desired. It also computes the misclosure of a closed traverse.

(3) This program uses the V program as a sub-routine for data entry, so Program V must be in the calculator as program V (or the XEQ V001 at line A027 changed to reflect the changed label. The V program allows entry of azimuths in D.MMSS format (HP notation) and distances, while converting them to the internal format. Program V is one of the HP-35s Utilities programs (Utility 3).

(4) This program uses the X program as a sub-routine for co-ordinate return from the internal format., so program X must be in the calculator as program X (or the XEQ X001 at line A053 changed to reflect the label change). The X program allows co-ordinates to be returned from the internal processing format of the calculator. Program X is one of the HP-35s Utilities programs (Utility 4).

(5) The program allows the user to choose if the co-ordinates of each point are to be calculated. The user is prompted with USE COORDS briefly, followed by the C? prompt. If co-ordinates are desired, key in 1, if not, key in 0, then press R/S to continue.

(6) After each side (azimuth and distance) has been entered, the calculator produces the following output.

A. If the user has selected to use co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in the display in line 2, and the Easting co-ordinate of the point in line 1.

Stack Register	Contents
T	Area of the traverse thus far
Z	Northing co-ordinate of current point
Y	Easting co-ordinate of current point
X	Number of sides entered

The user can scroll through the stack, using the R↓ key, and can perform any other operation of interest to the data on the stack. This information is stored in memory registers for use later in the program, so the stack may be changed and worked with as needed. The user can also continue without viewing the stack.

When the user presses R/S, the calculator takes the line to the current forward point from the starting point, and converts it into the distance (which is placed in line 2, the X register) and the azimuth in degrees, minutes and seconds (HP notation) in line 1 of the display (the Y register). The calculator also includes the proportional misclosure in the Z register of the stack.

Traverse Closure, Area Calc., Co-ordinates & Prop.Misc.

Stack Register	Contents
T	
Z	Proportional Misclosure (Representative Fraction)
Y	Azimuth from start to current point (in D.MMSSss)
X	Distance from start to current point

When the user presses R/S again, the calculator prompts for the azimuth of the next side to be entered. The azimuth should be entered in HP notation (DDD.MMSSss).

B. If the user has selected not to show co-ordinates, the calculator has the following data on the stack. It will stop and display this information, with the number of sides entered in line 2 of the display (the X register) and the distance of the misclosure or the line connecting the starting point to the current point, in line 1 of the display (Y register).

Stack Register	Contents
T	Area of the traverse thus far
Z	Azimuth of the line from the start to current point (in HP notation)
Y	Distance of the line from the start to the current point
X	Number of sides entered

By pressing the R↓ key, the user can see the azimuth in register Z of the stack. Pressing R↓ again will show the area to the current point.

If the rectangular components of the misclosure (or the line from the start to the current forward point) are needed, press RCL P, then XEQ X ENTER. The Easting component of the misclosure (ΔE) will be displayed in the X register (line 2), while the Northing component (ΔN) will be displayed in the Y register (line 1).

When the user presses R/S again, the calculator prompts for the azimuth of the next side.

(7) Azimuths are entered and displayed by themselves in HP notation, i.e., DDD.MMSSss.

(8) This program forms the basis of the two missing distances (2MD) program. Enter the known sides using this program to begin the 2MD computation process.

(9) In order to display the prompts, this program sets Flag 10. However, the program never ends, because it is up to the user to decide when to stop and move control elsewhere. So the program never clears Flag 10. If you require Flag 10 to be clear, in order to process equations, you must do this manually.

(10) The program calculates the proportional misclosure (representative fraction) of the traverse, and displays it in the stack. It only does this of the Co-ordinates

-

option is selected If you don't want to run the Co-ordinates option, but still want the proportional misclosure, do the following. When you have completed processing the traverse, press: RCL T RCL P **ABS** ÷

The denominator will show in the display. If the proportional misclosure is 1 in *X*, then *X* will be displayed.

Theory

The traverse closure programs works by converting the entered azimuths (in DDD.MMSS, or HP, notation) and distances into complex numbers (which act as 2-D vectors), which are then added to compute the location of points around the traverse. The area is computed by triangles developed by each new side of the traverse and the line from the starting point to the current forward point, and is updated with each new side. So the area is that of the polygon formed by the traverse entered thus far and the line from the start to the current point. This allows areas to be incremented for lot splitting calculations.

The azimuth and distance of the line from the start to the current point is also placed on the stack after each line. This allows a connecting line to be computed easily between two points. The final azimuth and distance is the traverse misclosure and the area is that of the traverse.

If the user chooses, the co-ordinates of the starting point may be entered, and if this choice is made, the calculator displays the co-ordinates of each point, in addition to the other information.

An arbitrary azimuth is satisfactory. Plane surveying assumptions apply. The program uses no error checking on entered data.

Sample Computation

Bearing	Distance
6° 53' 10"	72.00
112° 37' 20"	102.23
185° 39' 50"	29.04
181° 30' 00"	27.88
283° 54' 30"	102.38

Final Results

DE	=	0.0228
DN	=	–0.0022
Misclosure Length	=	0.0229
Misclosure Bearing	=	95° 24' 15"
Area	=	6,378.4660
Proportional Misclosure	=	1 in 14,537.95

Note that the value presented for the proportional misclosure will be 14,537.95

Traverse Closure, Area Calc., Co-ordinates & Prop.Misc.

Stepping through the Calculation

A. Without Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS, the C?

Key in 0, then press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	72.0000	(distance from start)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	6.5310	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	107.9004	(distance from start)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	72.3939	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	100.1841	(distance from start)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	88.0808	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	102.4027	(distance from start)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.6396	(area thus far)
	103.5423	(azimuth from start in HP notation (D.MMSS))

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	0.0229	(distance from start, also linear misclosure)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6378.4660	(area thus far)
	95.2415	(azimuth from start, also azimuth of misclosure)

Pressing RCL P, then XEQ X ENTER brings the misclosure to the display in rectangular form. The Northing component of the misclosure (ΔN) is in the Y register (line 1) and is –0.0022, while the Easting component of the misclosure (ΔE) is in the X register (line 2) and is 0.0228.

B. Using Co-ordinates

Press XEQ A ENTER

Calculator prompts with USE COORDS briefly, then C?

Key in 1, then press R/S.

Calculator prompts ENTER N0 briefly, then N?

Key in 1000.000, press R/S.

Calculator prompts ENTER E0 briefly, then E?

Key in 500.000, press R/S.

Side 1

Calculator prompts with A? for azimuth of side.

Key in 6.5310, press R/S.

Calculator prompts with D? for distance of side.

Key in 72.00, press R/S.

Display shows:	508.6325	(Easting co-ordinate of current forward point)
	1.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	0.0000	(area thus far)
	1,071.4806	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	6.5310	(Azimuth for current forward point from starting point)
	72.0000	(Distance from starting point to current forward point)

(Note that the proportional misclosure at this time will be meaningless, so we don't need to roll down to it.)

Press R/S.

Side 2

Calculator prompts with A? for azimuth of side.

Key in 112.372, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.23, press R/S.

Display shows:	602.9971	(Easting co-ordinate of current forward point)
	2.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	3,542.3468	(area thus far)
	1,032.1575	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	72.3939	(Azimuth for current forward point from starting point)
	107.9004	(Distance from starting point to current forward point)

(Note that the proportional misclosure at this time will be meaningless, so we don't need to roll down to it.)

(Note that the proportional misclosure at this time will be meaningless, so we don't need to roll down to it.)

Press R/S.

Side 3

Calculator prompts with A? for azimuth of side.

Key in 185.395, press R/S.

Calculator prompts with D? for distance of side.

Key in 29.04, press R/S.

Display shows:	600.1310	(Easting co-ordinate of current forward point)
	3.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	4,984.4807	(area thus far)
	1,003.2593	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	88.0808	(Azimuth for current forward point from starting point)
	100.1841	(Distance from starting point to current forward point)

Press R/S.

Side 4

Calculator prompts with A? for azimuth of side.

Key in 181.3, press R/S.

Calculator prompts with D? for distance of side.

Key in 27.88, press R/S.

Display shows:	599.4012	(Easting co-ordinate of current forward point)
	4.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6,378.6396	(area thus far)
	975.3888	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	103.5423	(Azimuth for current forward point from starting point)
	102.4027	(Distance from starting point to current forward point)

Press R/S.

Side 5

Calculator prompts with A? for azimuth of side.

Key in 283.543, press R/S.

Calculator prompts with D? for distance of side.

Key in 102.38, press R/S.

Display shows:	500.0228	(Easting co-ordinate of current forward point)
	5.0000	(number of sides entered)

Press the R↓ key twice, and the display becomes:

	6,378.4660	(area thus far)
	999.9978	(Northing co-ordinate of current forward point)

Press R/S.

Display shows:	95.2415	(Azimuth for current forward point from starting point)
	0.0229	(Distance from starting point to current forward point)

Traverse Closure, Area Calc., Co-ordinates & Prop.Misc.

Press the R↓ key, and the display becomes:

Display shows:	14,537.9488	(Proportional Misclosure 1 in *X*, where *X* is displayed)
	95.2415	(Azimuth for current forward point from starting point)

Press RCL P, then XEQ X ENTER to bring the misclosure onto the stack in rectangular mode. The misclosure in Northings (latitude, ΔN) will be displayed in the Y register (line 1) as –0.0022. The misclosure in Eastings (departure, ΔE) will be displayed in the X register (line 2) and is 0.0228.

Storage Registers Used

A Used by the V program for the entered azimuth.

B Stores 360 for azimuth correction.

C Test for displaying co-ordinates: 1 = YES; 0 = NO.

D Used by the V program for the entered distance.

E Easting co-ordinates of the starting point.

F Co-ordinates of starting point, as a complex number.

I Used by the V and X programs to address the additional storage registers.

M The number of sides entered.

N The Northing co-ordinate of the starting point.

P Current position of forward point, as a complex number.

Q Current area.

R Last side entered, as a complex number.

S Temporary storage for area calculation.

T Proportional Misclosure value 1 in *X*, where *X* is stored.

Statistical Registers: not used.

Other registers: 10, 11 and 12 used by the V program; 10 used by the X program.

Labels Used

Label **A** Length = 289 Checksum = A19C

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

The program labeled V, which takes an azimuth in degrees, minutes and seconds (in HP notation), and a distance, and converts them to a complex number for processing in the calculator. This routines uses storage location A and D, but copies out and replaces the contents of these storage locations in order to preserve them. It also uses storage register I for indirect addressing, which uses registers 10, 11 and 12.

The program labeled X, which takes a complex number and converts it to the rectangular components, is called for co-ordinate presentation in this program. It uses storage register I for indirect addressing, which uses register 10.

Label **V**	Length = 128	Checksum = 39FE
Label **X**	Length = 53	Checksum = C46D

HP-35s Calculator Program — Curves 1A

Solve the Parameters of a Circular Horizontal Curve, given any two Parameters

Programmer: Dr. Bill Hazelton

Date: April, 2008. **Version**: 1.1 **Mnemonic**: **S** for Curve **S**olution.

Line	Instruction	Display	User Instructions
S001	LBL S		LBL S
S002	CLSTK		CLEAR 5
S003	FS? 10		FLAGS 3 .0
S004	GTO S008		
S005	SF 1		FLAGS 1 1
S006	SF 10		FLAGS 1 .0
S007	GTO S009		
S008	CF 1		FLAGS 2 1
S009	SOLVE HZ CURVE		(Key in using EQN RCL S, RCL O, etc.)
S010	PSE		PSE
S011	CLx		CLEAR 1
S012	STO C		STO C
S013	STO R		STO R
S014	STO Q		STO Q
S015	STO T		STO T
S016	STO A		STO A
S017	CLΣ		CLEAR 4
S018	CHORD LENGTH		(Key in using EQN RCL C, RCL H, etc.)
S019	PSE		PSE
S020	INPUT C	C?	INPUT C
S021	x ≠ 0 ?		x?0 1
S022	Σ+		
S023	RADIUS		(Key in using EQN RCL R, RCL A, etc.)
S024	PSE		PSE
S025	INPUT R	R?	INPUT R
S026	x ≠ 0 ?		x?0 1
S027	Σ+		
S028	DEFLECTION θ		(Key in using EQN RCL D, RCL E, etc.)
S029	PSE		PSE
S030	INPUT Q	Q?	INPUT Q
S031	RCL Q		
S032	HMS→		HMS→
S033	STO Q		STO Q
S034	x ≠ 0 ?		x?0 1
S035	Σ+		
S036	TANGENT LENGTH		(Key in using EQN RCL T, RCL A, etc.)
S037	PSE		PSE
S038	INPUT T	T?	INPUT T
S039	x ≠ 0 ?		x?0 1

Line	Instruction
S040	Σ+
S041	ARC LENGTH
S042	PSE
S043	INPUT A
S044	x ≠ 0 ?
S045	Σ+
S046	2
S047	n *
S048	x < y?
S049	GTO S103
S050	x = y?
S051	GTO S057
S052	ONLY 2 INPUTS
S053	PSE
S054	SET 1 TO 0!
S055	PSE
S056	GTO S011
****	Start of "switch"
S057	RCL C
S058	x = 0?
S059	GTO S073
S060	RCL R
S061	x ≠ 0?
S062	GTO S108
S063	RCL Q
S064	x ≠ 0?
S065	GTO S112
S066	RCL T
S067	x ≠ 0?
S068	GTO S116
S069	RCL A
S070	x ≠ 0?
S071	GTO S140
S072	GTO S103
S073	RCL R
S074	x = 0?
S075	GTO S086
S076	RCL Q
S077	x ≠ 0?
S078	GTO S120
S079	RCL T
S080	x ≠ 0?

Line	Instruction
S081	GTO S124
S082	RCL A
S083	x ≠ 0?
S084	GTO S128
S085	GTO S103
S086	RCL Q
S087	x = 0?
S088	GTO S096
S089	RCL T
S090	x ≠ 0?
S091	GTO S132
S092	RCL A
S093	x ≠ 0?
S094	GTO S136
S095	GTO S103
S096	RCL T
S097	x = 0?
S098	GTO S103
S099	RCL A
S100	x = 0?
S101	GTO S103
S102	GTO S144
****	Error message
S103	NOT ENOUGH
S104	PSE
S105	DATA! RE–ENTER
S106	PSE
S107	GTO S011
****	C, R, known
S108	XEQ S185
S109	XEQ S255
S110	XEQ S215
S111	GTO S147
****	C, Q, known
S112	XEQ S221
S113	XEQ S255
S114	XEQ S215
S115	GTO S147
****	C, T, known
S116	XEQ S206
S117	XEQ S221
S118	XEQ S215

Line	Instruction
S119	GTO S147
****	R, Q, known
S120	XEQ S246
S121	XEQ S255
S122	XEQ S215
S123	GTO S147
****	R, T, known
S124	XEQ S199
S125	XEQ S246
S126	XEQ S215
S127	GTO S147
****	R, A, known
S128	XEQ S194
S129	XEQ S246
S130	XEQ S255
S131	GTO S147
****	Q, T, known
S132	XEQ S237
S133	XEQ S221
S134	XEQ S215
S135	GTO S147
****	Q, A, known
S136	XEQ S231
S137	XEQ S246
S138	XEQ S255
S139	GTO S147
****	C, A, known
S140	XEQ S265
S141	XEQ S221
S142	XEQ S255
S143	GTO S147
****	T, A, known
S144	XEQ S284
S145	XEQ S231
S146	XEQ S246
****	Seg area & show
S147	RCL Q
S148	→RAD
S149	RCL Q
S150	SIN
S151	—
S152	RCL R

* This is the statistical count, retrieved using ↱ SUMS n.
**** These lines are simply comments in the code. You don't key it into the calculator!

Solve the Parameters of a Circular Horizontal Curve

Line	Instruction
S153	x^2
S154	×
S155	2
S156	÷
S157	STO B
S158	SOLUTION
S159	PSE
S160	CHORD
S161	PSE
S162	VIEW C
S163	RADIUS
S164	PSE
S165	VIEW R
S166	TANGENT
S167	PSE
S168	VIEW T
S169	ARC LENGTH
S170	PSE
S171	VIEW A
S172	RCL Q
S173	→HMS
S174	STO Q
S175	DEFLECTION θ
S176	PSE
S177	VIEW Q
S178	RCL Q
S179	HMS→
S180	STO Q
S181	SEGMENT AREA
S182	PSE
S183	VIEW B
S184	GTO S305
****	Calculate Q – 1
S185	RCL C
S186	2
S187	÷
S188	RCL÷ R
S189	ASIN
S190	2
S191	×
S192	STO Q
S193	RTN
****	Calculate Q – 2
S194	RCL A
S195	RCL÷ R

Line	Instruction
S196	→DEG
S197	STO Q
S198	RTN
****	Calculate Q – 3
S199	RCL T
S200	RCL÷ R
S201	ATAN
S202	2
S203	×
S204	STO Q
S205	RTN
****	Calculate Q – 4
S206	RCL C
S207	2
S208	÷
S209	RCL÷ T
S210	ACOS
S211	2
S212	×
S213	STO Q
S214	RTN
****	Calculate A
S215	RCL R
S216	RCL Q
S217	→RAD
S218	×
S219	STO A
S220	RTN
****	Calculate R – 1
S221	RCL C
S222	2
S223	÷
S224	RCL Q
S225	2
S226	÷
S227	SIN
S228	÷
S229	STO R
S230	RTN
****	Calculate R – 2
S231	RCL A
S232	RCL Q
S233	→RAD
S234	÷
S235	STO R

Line	Instruction
S236	RTN
****	Calculate C – 1
S237	2
S238	RCL× T
S239	RCL Q
S240	2
S241	÷
S242	COS
S243	×
S244	STO C
S245	RTN
****	Calculate C – 2
S246	2
S247	RCL× R
S248	RCL Q
S249	2
S250	÷
S251	SIN
S252	×
S253	STO C
S254	RTN
****	Calculate T
S255	RCL C
S256	2
S257	÷
S258	RCL Q
S259	2
S260	÷
S261	COS
S262	÷
S263	STO T
S264	RTN
****	Calculate Q (AC)
S265	0
S266	STO U
S267	RCL A
S268	RCL÷ C
S269	1
S270	—
S271	0.06
S272	÷
S273	√x
S274	→DEG
S275	STO Q
S276	FN= U

Line	Instruction
S277	SOLVE Q
S278	RTN
S279	CANNOT SOLVE
S280	PSE
S281	WITH THESE DATA
S282	PSE
S283	GTO S305
****	Calculate Q (AT)
S284	1
S285	STO U
S286	RCL T
S287	RCL÷ A
S288	0.5
S289	—
S290	7
S291	×
S292	RCL A
S293	RCL÷ T
S294	x^2
S295	÷
S296	→DEG
S297	STO Q
S298	FN= U
S299	SOLVE Q
S300	RTN
S301	CANNOT SOLVE
S301	PSE
S303	WITH THESE DATA
S304	PSE
****	End part
S305	FS? 1
S306	CF 10
S307	STOP
S308	RTN

Line	Instruction
U001	LBL U
U002	RCL U
U003	x = 0?
U004	GTO U017
U005	RCL Q
U006	2
U007	÷
U008	TAN
U009	RCL Q
U010	→RAD
U011	÷
U012	RCL T
U013	RCL÷ A
U014	—
U015	→DEG
U016	RTN
U017	RCL Q
U018	2
U019	÷
U020	SIN
U021	2
U022	×
U023	RCL Q
U024	→RAD
U025	÷
U026	RCL C
U027	RCL÷ A
U028	—
U029	→DEG
U030	RTN

Notes

1. The **** lines are comments and are not to be entered into the calculator. They are there to make it easier to work through entering a long program.

2. Be very careful when entering the line numbers in the various XEQ and GTO statements.

3. Angles are entered and displayed in HP notation (DDD.MMSSss).

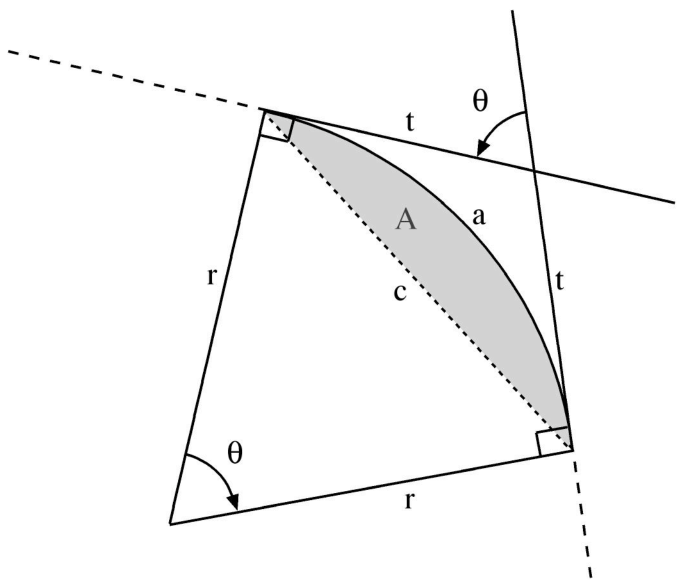
4. The program will not work for parameters that are the result of a deflection angle greater than or equal to 180°. This produces a "division by zero" error. Similarly, "impossible figures" will not produce correct results.

5. Some pairs of parameters have considerable sensitivity to small variations in their values. Therefore, consider doing a little sensitivity analysis (e.g., re-do the calculation with the parameters changed by an amount about equal to the expected error in them) to see what a reasonable precision of the result might be.

6. This program is designed to work with exactly two parameters. If you have more or fewer, the program will demand that you use only two. Choose the two most suitable parameters and ignore the others, using them as a check on the values produced. The program cannot do an adjustment based on redundant data.

Theory

The theory of solving the parameters of a horizontal circular curve is fairly straightforward. Given a curve as shown in the figure below, the various parameters are related through the following equations. Therefore, given any two parameters, it is possible to solve for all the others.



In this situation, θ is the deflection angle, or angle at the center of the arc; c is the chord length; r is the radius; a is the length of the arc of the curve; t is the length of the tangent, from the tangent point to the intersection point; and A is the area of the segment between the arc and the chord (shown with gray shading).

The perpendicular bisector of the chord also bisects the angle at the center of the curve (θ), dividing the quadrilateral into two congruent right triangles, and the isosceles triangle formed by the radii and the chord into two other congruent right triangles. Solving these triangles in various ways allows any two parameters to solve most of the other parameters. The formulae used are as follows:

$$\theta = 2\arcsin\left(\frac{c/2}{r}\right) = 2\arccos\left(\frac{c/2}{t}\right) = 2\arctan\left(\frac{t}{r}\right)$$

$$r = \frac{c/2}{\tan\left(\theta/2\right)} = \frac{a}{\theta}$$

$$c = 2r\sin\left(\frac{\theta}{2}\right) = 2t\cos\left(\frac{\theta}{2}\right)$$

$$t = \frac{c/2}{\cos\left(\theta/2\right)}$$

$$a = r\theta$$

$$A = \frac{1}{2}r^2\left(\theta - \sin\theta\right)$$

When θ is used by itself, it usually denotes its use as a radian value.

In the event that the chord and arc, or the tangent and arc, are the only values known, the solution is not direct. Instead, the following equations are set up (in the subprogram with label U), for each case:

$$\frac{2\sin\left(\frac{\theta}{2}\right)}{\theta} - \frac{c}{a} = 0$$

and

$$\frac{\tan\left(\frac{\theta}{2}\right)}{\theta} - \frac{t}{a} = 0$$

respectively.

These are solved for θ using the HP Solve capability in the calculator, after a starting estimate for θ is calculated.

Running the Program

Key in XEQ S then press the Enter key. The program starts and displays:

SOLVE HZ CURVE

then prompts for the chord length, displaying:

CHORD LENGTH

then stops while displaying:

C?
0.0000

If the length of the chord is known, key it in and press R/S. If it is not known, leave the value at zero and press R/S. The calculator then displays:

RADIUS

then stops while displaying:

R?
0.0000

If the radius is known, key it in, then press R/S. If it is not known, leave the value at zero and press R/S. The calculator then displays:

DEFLECTION θ

then stops while displaying:

Q?
0.0000

If the value of the deflection angle is known, key it in here in DDD.MMSSss format (HP notation), then press R/S. if the deflection angle is not known, leave the value at zero and press R/S. The calculator then displays:

TANGENT LENGTH

then stops while displaying:

T?
0.0000

If the length of the tangent is known, key it in here and press R/S. If it is not known, leave the value at zero and press R/S. The calculator then displays:

ARC LENGTH

then stops while displaying:

A?
0.0000

If the arc length is known, key it in here and press R/S. If it is not known, leave the alue at zero and press R/S.

If you have entered fewer than two parameter values, i.e., there are fewer than two non-zero values, the calculator briefly displays:

NOT ENOUGH then, briefly:
DATA! RE-ENTER

and returns to prompting for the chord length, as above. You then are prompted for all the other possible data values, in turn, as shown above.

If you have entered more than two parameter values, the calculator briefly displays:

ONLY 2 INPUTS then, briefly:
SET 1 to 0!

and returns to prompting for the chord length, as above. You then are prompted for all the other possible data values, in turn, as shown above.

If you have entered exactly two parameter values, the calculator displays:

RUNNING then, briefly:

SOLUTION then, briefly:

CHORD

then stops and shows the chord value, e.g.:

C=
258.8190

Press R/S. The calculator briefly displays:

RADIUS

then stops and shows the radius value, e.g.:

R=
500.0000

Press R/S. The calculator briefly displays:

TANGENT

then stops and shows the tangent length, e.g.:

T=
133.9750

Press R/S/. The calculator briefly displays:

ARC LENGTH

then stops and shows the arc length, e.g.:

A=
261.7990

Press R/S. The calculator briefly displays:

DEFLECTION θ

then stops and shows the deflection angle, q, in HP notation (DDD.MMSSss format), e.g.:

Q=
30.595900

Press R/S. The calculator briefly displays:

SEGMENT AREA

then stops and shows the area of the segment between the chord and the arc, e.g.:

B=
34,199.8470

Press R/S. The program resets flag 10 to its original value, then stops and returns to normal calculator operation.

In the event that the parameters entered were the chord and arc lengths, or the tangent and arc lengths, the solution will take a little longer, and the calculator will display:

SOLVING

for a short time, while the HP Solve process is being done. As this is the first step in both cases, it is followed by the calculator displaying:

RUNNING

before moving to display the solution.

Sample Computations

	1	**2**	**3**
Radius	500.000	500.000	500.000
Deflection Angle	30.000	45.000	60.000
Chord Length	258.819	382.683	500.000
Tangent Length	133.975	207.107	288.675
Arc Length	261.799	392.699	523.599
Segment Area	2,949.847	9,786.423	22,646.518

Entering various combinations of any two values for any one solution should give the other parameter values. However, there may be some sensitivity when various input parameters are used, so that there will be some small variation in the output parameters in some cases, In particular, the area may change by small amounts, and solutions that start with the arc length are sometimes particularly sensitive.

	4	5	6
Radius	500.000	500.000	500.000
Deflection Angle	90.000	120.000	150.000
Chord Length	707.107	866.025	965.926
Tangent Length	500.000	866.025	1,866.025
Arc Length	785.398	1,047.198	1,308.997
Segment Area	71,349.541	153,546.212	264,749.235

	7	8	9
Radius	500.000	500.000	250.000
Deflection Angle	170.000	105.000	109.4522
Chord Length	996.195	793.353	408.965
Tangent Length	5,715.026	651.613	355.425
Arc Length	1,483.530	916.298	478.901
Segment Area	349,176.444	108,333.736	30,452.048

Storage Registers Used

A Arc length (a).

B Segment area (A).

C Chord length (c).

Q Deflection angle (θ).

R Radius (r).

T Tangent length (t).

U Selector for TA or CA solutions in HP Solve.

Statistical Registers: Used to count the number of parameters entered, only the count (n) is used.

Labels Used

Label **S** Length = 1180 Checksum = F62D

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

The program labeled U looks at the value in the storage register U, and uses this to decide whether a TA (tangent and arc lengths) or CA (chord and arc length) solution is needed, then jumps to the part of the code that implements the equation to be solved. The HP Solve package in the calculator uses the code under label U as the basis for solving for the deflection angle (θ, stored in Q the program), when called from the main program (under label S). The subprogram under label U accesses the storage locations A, Q, and C or T, as needed, for the solution.

Label **U** Length = 93 Checksum = 894F

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Special Thanks

The original version of the program had an error in the formula for the area of the segment, which was also in the code, and hence in the examples. There were some other typos in the examples, as well. Asa Ramsay was kind enough to bring these to my attention, allowing me to fix them quickly and post a revised program.

HP-35s Calculator Program Curves 2A

Compute Values for a Parabolic Vertical Curve

Programmer: Dr. Bill Hazelton

Date: March, 2008. **Version**: 1.0 **Mnemonic**: **P** for **P**arabolic Vertical Curve.

Line	Instruction	Display	User Instructions
P001	LBL P		LBL P
P002	CLSTK		CLEAR 5
P003	FS? 10		FLAGS 3 .0
P004	GTO P008		
P005	SF 1		FLAGS 1 1
P006	SF 10		FLAGS 1 .0
P007	GTO P009		
P008	CF 1		FLAGS 2 1
P009	COMP V CURVE		(Key in using EQN RCL C, RCL O, etc.)
P010	PSE		PSE
P011	CLx		CLEAR 1
P012	STO S		STO S
P013	STO R		STO R
P014	STO Q		STO Q
P015	STO P		STO P
P016	STO A		STO A
P017	STO L		STO L
P018	ENTER P I RD		(Key in using EQN RCL E, RCL N, etc.)
P019	PSE		PSE
P020	INPUT R	R?	INPUT R
P021	ENTER P I EL		(Key in using EQN RCL E, RCL N, etc.)
P022	PSE		PSE
P023	INPUT S	S?	INPUT S
P024	START GRADE		(Key in using EQN RCL S, RCL T, etc.)
P025	PSE		PSE
P026	INPUT P	P?	INPUT P
P027	END GRADE		(Key in using EQN RCL E, RCL N, etc.)
P028	PSE		PSE
P029	INPUT Q	Q?	INPUT Q
P030	RCL Q		
P031	RCL— P		
P032	STO A		STO A
P033	ENTER LENGTH		(Key in using EQN RCL E, RCL N, etc.)
P034	PSE		PSE
P035	INPUT L	L?	INPUT L
P036	RCL R		
P037	RCL L		
P038	2		
P039	÷		
P040	—		

Line	Instruction
P041	STO U
P042	RCL S
P043	RCL P
P044	RCL× L
P045	2
P046	÷
P047	—
P048	STO V
P049	RCL A
P050	RCL÷ L
P051	2
P052	÷
P053	STO B
P054	RCL P
P055	RCL÷ B
P056	2
P057	÷
P058	+/—
P059	STO I
****	Max/Min Pt. Data
P060	0
P061	STO Y
P062	MAX–MIN [0–1]
P063	PSE
P064	INPUT Y
P065	RCL Y
P066	x ≤ 0?
P067	GTO P085
P068	RCL I
P069	x^2
P070	RCL× B
P071	RCL I
P072	RCL× P
P073	+
P074	RCL+ V
P075	STO E
P076	MAX—MIN EL
P077	PSE
P078	VIEW E
P079	MAX—MIN RD
P080	PSE
P081	RCL U

Line	Instruction
P082	RCL+ I
P083	STO D
P084	VIEW D
****	End Pts Data
P085	0
P086	STO Y
P087	END PTS [0–1]
P088	PSE
P089	INPUT Y
P090	RCL Y
P091	x ≤ 0?
P092	GTO P116
P093	START PT RD
P094	PSE
P095	VIEW U
P096	START PT EL
P097	PSE
P098	VIEW V
P099	END PT RD
P100	PSE
P101	RCL U
P102	RCL+ L
P103	STO F
P104	VIEW F
P105	RCL L
P106	x^2
P107	RCL× B
P108	RCL L
P109	RCL× P
P110	+
P111	RCL+ V
P112	STO E
P113	END PT EL
P114	PSE
P115	VIEW E
****	Get EL from RD
P116	0
P117	STO Y
P118	STO X
P119	COMP 1 EL [0–1]
P120	PSE
P121	INPUT Y

Line	Instruction
P122	RCL Y
P123	x ≤ 0?
P124	GTO P149
P125	ENTER RD
P126	PSE
P127	INPUT X
P128	RCL X
P129	RCL— U
P130	STO I
P131	x^2
P132	RCL× B
P133	RCL I
P134	RCL× P
P135	+
P136	RCL+ V
P137	STO H
P138	EL OF POINT
P139	PSE
P140	VIEW H
P141	0
P142	STO Y
P143	AGAIN [0–1]
P144	PSE
P145	INPUT Y
P146	RCL Y
P147	x > 0?
P148	GTO P125
****	Get RD from EL
P149	0
P150	STO Y
P151	STO H
P152	STO X
P153	COMP 1 RD [0–1]
P154	PSE
P155	INPUT Y
P156	RCL Y
P157	x ≤ 0?
P158	GTO P224
P159	ENTER EL
P160	PSE
P161	INPUT H
P162	RCL V

**** These lines are simply comments in the code. You don’t key it into the calculator!

Line	Instruction
P163	RCL— H
P164	STO C
P165	RCL P
P166	x^2
P167	RCL B
P168	RCL× C
P169	4
P170	×
P171	—
P172	STO D
P173	x < 0?
P174	GTO P212
P175	x = 0?
P176	GTO P201
P177	RCL D
P178	√x
P179	STO D
P180	RCL— P
P181	RCL÷ B
P182	2
P183	÷
P184	RCL+ U
P185	STO X
P186	FIRST RD
P187	PSE
P188	VIEW X
P189	RCL D
P190	+/—
P191	RCL— P
P192	RCL÷ B
P193	2
P194	÷
P195	RCL+ U
P196	STO X
P197	SECOND RD
P198	PSE
P199	VIEW X
P200	GTO P216
P201	RCL P
P202	+/—
P203	RCL÷ B
P204	2
P205	÷
P206	RCL+ U

Line	Instruction
P207	STO X
P208	SINGLE RD
P209	PSE
P210	VIEW X
P211	GTO P216
P212	THIS ELEV
P213	PSE
P214	NOT ON CURVE
P215	PSE
P216	0
P217	STO Y
P218	AGAIN [0–1]
P219	PSE
P220	INPUT Y
P221	RCL Y
P222	x > 0?
P223	GTO P159
****	Step thru RDs
P224	0
P225	STO Y
P226	STO X
P227	STO H
P228	STO D
P229	STO C
P230	STO I
P231	STEP THRU RD
P232	PSE
P233	NO—YES [0–1]
P234	PSE
P235	INPUT Y
P236	RCL Y
P237	x ≤ 0?
P238	GTO P313
P239	FIRST INCRMNT
P240	PSE
P241	INPUT C
P242	GENRL INCRMNT
P243	PSE
P244	INPUT D
P245	RD VALUE
P246	PSE
P247	VIEW U
P248	EL VALUE
P249	PSE

Line	Instruction
P250	VIEW V
P251	RD VALUE
P252	PSE
P253	RCL C
P254	RCL+ U
P255	STO X
P256	VIEW X
P257	RCL C
P258	x^2
P259	RCL× B
P260	RCL C
P261	RCL× P
P262	+
P263	RCL+ V
P264	STO H
P265	EL VALUE
P266	PSE
P267	VIEW H
P268	RCL C
P269	STO I
P270	RCL D
P271	STO+ I
P272	RCL I
P273	RCL— L
P274	x > 0?
P275	GTO P294
P276	RCL I
P277	x^2
P278	RCL× B
P279	RCL I
P280	RCL× P
P281	+
P282	RCL+ V
P283	STO H
P284	RCL I
P285	RCL+ U
P286	STO X
P287	RD VALUE
P288	PSE
P289	VIEW X
P290	EL VALUE
P291	PSE
P292	VIEW H
P293	GTO P270

Line	Instruction
P294	END POINT
P295	PSE
P296	RCL L
P297	x^2
P298	RCL× B
P299	RCL L
P300	RCL× P
P301	+
P302	RCL+ V
P303	STO H
P304	RCL L
P305	RCL+ U
P306	STO X
P307	RD VALUE
P308	PSE
P309	VIEW X
P310	EL VALUE
P311	PSE
P312	VIEW H
****	End part
P313	PROGRAM END
P314	PSE
P315	FS? 1
P316	CF 10
P317	STOP
P318	RTN

Notes

1. The **** lines are comments and are not to be entered into the calculator. They are there to make it easier to work through entering a long program.

2. The program offers options for the processing to be done at each step, asking the user if a particular set of operations are to be done. The user is prompted for a yes/no answer, with the input variable being Y, which is set to 0 (no) by default. To skip the operation, just press R/S. To do the operation, key in 1, then press R/S. The sense is that 0 = no, 1 = yes. Note that there is a need to pause after the prompt, otherwise the entered value will not make it into the calculator, and the ‘no’ response will be acted upon.

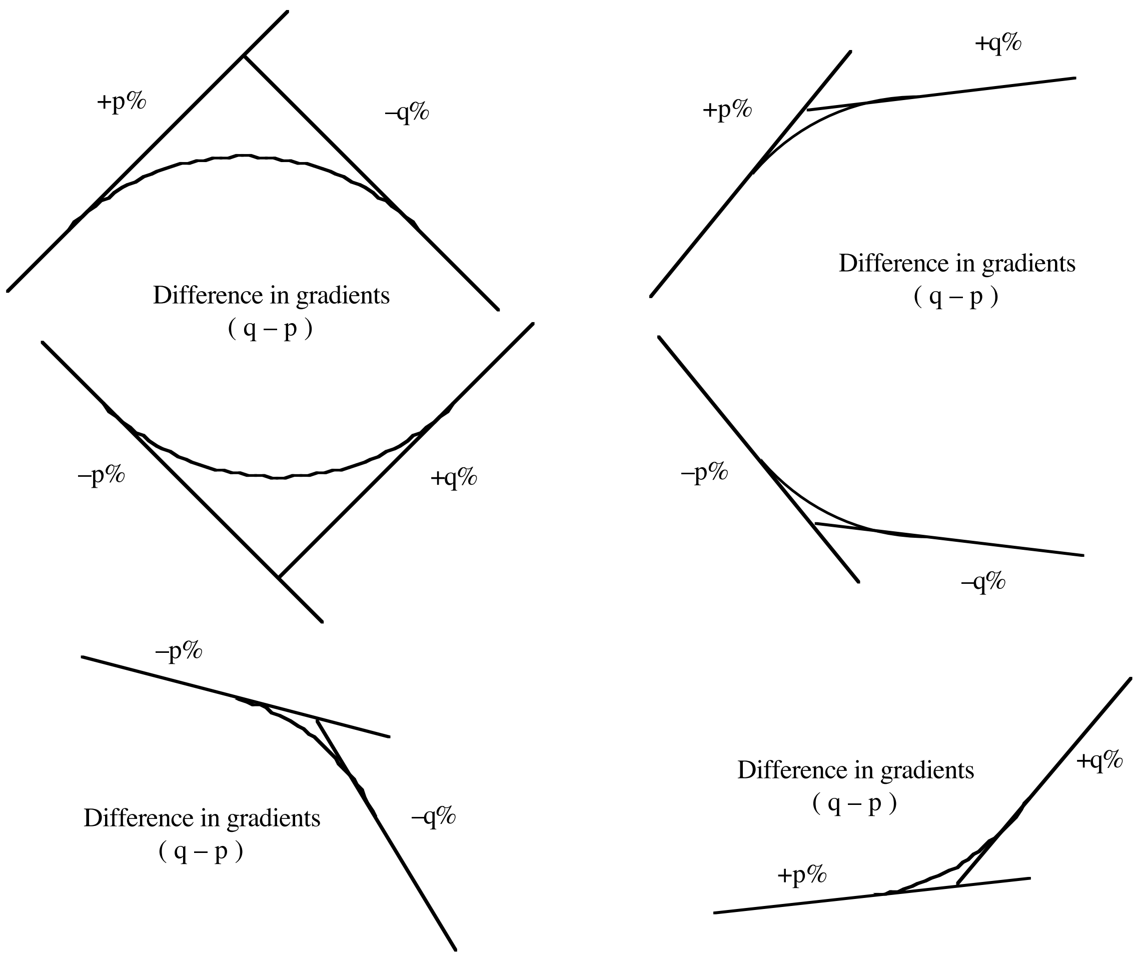
3. Because grades are used, it is important that the units for elevation and running distance are the same. Otherwise the computations will not be correct.

4. Some of the calculations will yield results that are on the parabola being used, but not within the actual segment being used. This is particularly the case with finding maximum and minimum points, which may not lie within the limits of the curve, and finding running distances given elevations. The user should check that the running distances fall within the

start and end points of the curve and ignore results that lie outside the curve. The program does not check these limits.

Theory and Background

The theory of computing the various values for a parabolic 'equal-tangent-length' vertical curve is fairly straightforward. There are six basic possibilities of how the vertical curve can be placed between two grade lines, as shown in the figure below. The curve allows the grade to be changed smoothly from the incoming grade (p) to the outgoing grade (q).



The convention with vertical curves is that they are drawn and computed going from left to right. If the grade is rising from left to right, the gradient is positive. If it is going down from left to right, the gradient is negative. The above diagram shows the signs that the various gradients would take in the circumstances.

If the grades on opposite sides of the curve have the same sign, as in the two examples in the right in the figure above, there is no maximum or minimum value along the curve (other than at its end points). If the grades have opposite signs (as in the two examples on the left side of the above diagram), then there will be a maximum or minimum point of the parabola somewhere along the curve. In this case, the calculation of the elevation and location of such a point is meaningful. If such a calculation is made for the parabolas on the right in the diagram, the turning point for the parabola will be determined, but it will be well outside the part of the parabolic curve actually used. For example, the parabolas shown on the right side of the diagram above have their turning points well to the right of the end of the curve.

For the purposes of vertical curve design, parabolas are the preferred curve. They are simpler to compute than circles or ellipses, but differ from them by amounts that are too small to matter in almost all cases. With a parabola, there is a constant change of grade (or gradient) around the curve, whereas for a circle there is a constant change of angle around the curve.

Another useful characteristic of the parabola is that it can be placed so that the lengths of the tangents are always equal. As the tangents are usually close to level (grades usually being fairly small), the horizontal distances from the point of intersection of the two grade lines to the tangent points are equal, and half the length of the curve. This allows easy placement of the curve with respect to the point of intersection of the grade lines. [Note that if the in and out grades are not equal and of opposite sign, the maximum or minimum point will not fall directly below the intersection point.]

Gradients, Grades, Slopes and Angles

The slopes of the lines into and out of the vertical curve may be expressed in several ways. For this program, gradients should be entered as a decimal value of the gradient.

The gradient is the value of the change in elevation over a horizontal distance divided by that distance, i.e., rise over run, expressed as a decimal number. So, if the slope rises by 2 units for every 100 along, the gradient is $+2 \div 100 = +0.02$. If the slope falls by 5 units for 125 units along, the gradient is $-5 \div 125 = -0.04$.

The gradient can also be expressed as a percentage, which is simply the gradient value (as above) multiplied by 100 to convert it to a percentage. So the above examples would be +2% and –4%, respectively. Percentages are also handy in that they link in well to horizontal distances expressed in stations. As the distance is then in 100 ft units, a 1 foot rise would be a +1% grade, so the rise or fall over one 100 ft 'station' can be converted directly to the percentage gradient.

Gradients can also be expressed as a ratio of the rise to the run, and expressed in the form "1 in so many." To get this "so many" value, simply calculate the reciprocal value of the gradient (as a decimal), so for the two example given above, +0.02 would be $^1/_{0.02} = 50$, and so +1 in 50; –0.04 would be $^1/_{0.04} = 25$, and so –1 in 25.

Gradients can also be represented as the angle of the line from the horizontal, usually given in decimal degrees. The tangent of this angle will be the gradient, and the gradient can be converted to an angle by taking the arctangent of the gradient. So a gradient of +0.02 will give an angle of $\arctan(+0.02) = +1.146°$, while –0.04 will give and angle of $\arctan(-0.04) = -2.291°$. A slope of 1° would give a gradient of $\tan(1°) = 0.017$.

Grade and gradient are used interchangeably, although sometimes they are applied to specific representations. For this program, convert all gradients to the decimal format, e.g., +0.02, –0.04. Be aware that the sign of the gradient is very important and ***must*** be included.

The term 'slope' is also used, but it usually doesn't refer to a specific representation.

Horizontal Distances

Horizontal distances as used in the construction of linear objects are commonly expressed as a distance from a starting point somewhere along the object. How they are expressed depends upon the units being used, the country in which they are being used, and local practice. Similarly, what they are called also varies.

Distances in feet are commonly recorded as ‘station’ values. Here, it is assumed that a station is marked every 100 ft, and that the stations are numbered sequentially from the start, with distances on from the station noted as additional distance. So a distance of 12,546.78 ft would be recorded as 125 + 46.78, meaning 125 stations of 100 feet, plus 46.78 feet past that station. For many construction projects, having points every 100 ft (30.48 m) is very convenient, hence the popularity of this representation. It is easy to convert between the distance representation (12,546.78 ft) and the station representation (125 + 46.78): simply remove the + sign and place the digits together to go to distance, or open the digits two left of the decimal point and put in a +, to convert to stations. (Calculators prefer the distance version.)

With metric units, 100 m is a bit long for station placement, so an equivalent metric representation never really caught on. In metric, it is more usual to use the distance representation (in meters) for all uses. This is also simpler as surveying moves away from the reliance on short lines for set-out (greatly aided by a station every 100 ft), to total station based set-out by co-ordinates across large areas.

The distances are known as ‘stations’ (when working with the 100 ft units), but this is a little odd when using a distance representation. In this case, the distance may be known as the ‘Running Distance’ (abbreviated RD), or the ‘chainage’ in some circumstances. As the station representation is so easily converted to the distance representation, and this program can also be used for metric applications, the horizontal distances in this program will be termed ‘Running Distances’ and often noted as RD.

Calculating The Parabola

The general equation of a parabola is:

$$\mathrm{y} = a\mathrm{x}^2 + b\mathrm{x} + c$$

The magnitude of the term *a* controls the sharpness of the parabola, while the sign of *a* controls the orientation. With *a* positive, the parabola is turned upwards and is bowl-shaped (the apex or turning point is the smallest y value), while with *a* negative, the parabola is hill-shaped, with the apex having the largest y value. So a summit or crest has a negative *a*, while a sag has a positive *a*.

It is convenient to use the starting point of the parabola (i.e., the first or left-most tangent point) as the origin of the co-ordinates. The elevation of this point (A on the diagram overleaf) above the chosen datum is equal to the term *c* in the equation. The slope $\frac{dy}{dx}$ of any tangent is equal to $2a\mathrm{x} + b$. But as at Point A, x = 0, the term *b* in the general equation is the slope or gradient at point A, the tangent gradient p.

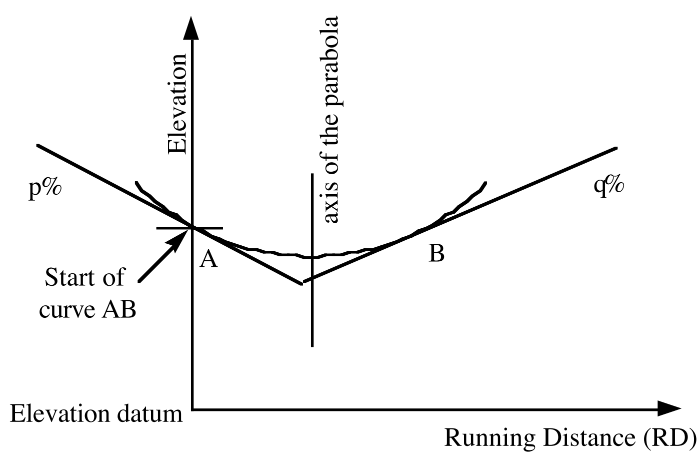
Compute Values for a Parabolic Vertical Curve

The second derivative $\frac{d^2y}{dx^2}$ of the general parabola equation equals $2a$, a constant. This means that a tangent to a vertical-axis parabola changes a constant amount of *grade* for each increment of distance. (In contrast, a tangent to a vertical circular curve changes direction a constant amount of *angle* for equal increments of distance along the arc.) The useful consequence is that the rate of change of grade on a vertical curve is constant and equals $2a$ per 100 units of distance (feet or meters, depending upon the units chosen). On a vertical curve the total change in direction between the profile grades is q – p, termed A.

If this change is accomplished on a curve L units long, the constant rate of change must be:

$$2a = \frac{q - p}{L} = \frac{A}{L}$$

with gradients q, p and A in decimal form, and L in either feet or meters, consistent with the job.



A practical formula for a general parabola for a vertical curve is therefore

$$\text{Elevation} = a\,(\text{RD} - \text{RD}_A)^2 + p\,(\text{RD} - \text{RD}_A) + \text{Elevation}_A$$

where RD is the running distance of any point, RD_A is the running distance of point A, a is the first parameter of the parabola (half the rate of change of grade as a decimal), p is the slope of the entry (or incoming) tangent (in decimal form), Elevation is the elevation of the point at running distance RD, and Elevation_A is the Elevation of point A. Elevation and RD should be in the same units, consistent with the entire job.

The location of the turning point of the parabola (the apex) can be computed by noting that the slope of the tangent at the turning point is 0, and solving $2ax + b = 0$. In this case, the location of the turning point is: $x = \frac{-b}{2a}$. This value can be converted to a running distance by adding the value of RD_A, and then used in the equation to compute the elevation of the maximum or minimum point on the curve.

Given the Elevation of a point on the curve, its location, x, from Point A can be computed by solving the equation:

$$a\,x^2 + p\,x + (\text{Elevation}_A - \text{Elevation}) = 0$$

As this is a quadratic, the standard quadratic solution will produce two solutions (in most cases), and it is up to the user to decide which is the most applicable. The program deals with the various cases (0, 1 and 2 solutions) separately. If there is a single solution, the apex (or nadir) of the curve has been selected. If there are no solutions, then the elevation selected is beyond the turning point elevation of the curve and therefore cannot lie on the curve.

For determining the elevations at set distances along the curve, the user can specific an initial increment, to bring the steps onto an even running distance, then specify a general increment, which will be used for the remainder of the curve. The program calculates the elevation of the first point on the curve (the left-hand tangent point), then moves along the initial increment, then proceeds along the curve using steps of the general increment, until the end of the curve is reached. The final point on the curve (the right-hand tangent point) is calculated and the program ends. For each point, the running distance and elevation are shown.

Note that if the final point calculated by increments happens to also be the end point, it will be calculated and displayed, and then the end point will be calculated again and displayed. This is because the test for coming to the end of the curve is that the increment is beyond the end point.

Running the Program

Key in XEQ P then press the Enter key. The program starts and displays:

COMP V CURVE

then prompts for the running distance of the intersection point of the two grade lines, displaying:

ENTER P I RD

then stops while displaying:

R?
0.0000

Key in the RD of the intersection point and press R/S. The calculator then prompts for the elevation of the intersection point of the two grade lines, displaying:

ENTER P I EL

then stops while displaying:

S?
0.0000

Key in the elevation of the intersection point, then press R/S. The calculator then prompts for the starting grade (p), displaying:

START GRADE

then stops while displaying:

P?
0.0000

The value of the slope coming in to the vertical curve should be entered as a decimal value, e.g., 0.025 (an upward 2.5% gradient), then press R/S. The calculator then prompts for the grade coming out of the vertical curve, displaying:

END GRADE

then stops while displaying:

Q?
0.0000

Key in the value of the slope of the grade coming out of the curve, as a decimal grade, e.g., –0.04 (a downward 4% gradient), then press R/S. The calculator then prompts for the length of the curve, briefly displaying:

ENTER LENGTH

then stops while displaying:

L?
0.0000

Key in the length of the curve desired and press R/S.

From this point on, the program asks if the user wants to do optional computations. These options are presented in the order that follows, and any or all of them may be skipped or run.

A. Maximum / Minimum Point

The calculator display RUNNING briefly, then prompts for whether or not to calculate the point on the (whole) parabola where the maximum or minimum elevation occurs. If the vertical curve is a crest or sag, the maximum or minimum point will occur on the actual curve segment. If the in and out grades have the same sign, the maximum or minimum point will not be on the actual curve segment, so this option can be skipped. The calculator briefly displays:

MAX–MIN [0–1]

then stops and displays the prompt:

Y?
0.0000

The 0 means ‘no’ to the question of whether to run this option, and is the default response. If you don’t want to run this option, press R/S and the program will advance to the next option. If you do want to run this option, key in 1 (or any number greater than 0), which signifies 'yes,' and press R/S. The calculator briefly displays:

MAX–MIN EL

then stops and displays the Elevation of the maximum or minimum point, looking as follows:

E=
114.5611

Press R/S to continue. The calculator then briefly displays:

MAX–MIN RD

then stops and displays the Running Distance of the maximum or minimum point, looking as follows:

D=
24,589.2222

Press R/S to continue, and this option concludes.

B. Compute the End Points

This option computes the running distance and elevation of the end points of the curve, where the parabolic curve joins the tangent gradients. The calculator briefly displays:

END PTS [0–1]

then stops and displays the prompt:

Y?
0.0000

If you don't want to run this option, press R/S and the program will advance to the next option. If you do want to run this option, key in 1 (or any number greater than 0), which signifies 'yes,' and press R/S. The calculator briefly displays:

START PT RD

then stops and display the running distance of the start point of the curve, such as follows:

U=
24,367.0000

Press R/S to continue. The calculator briefly displays:

START PT EL

then stops and displays the elevation of the start point of the curve, such as follows:

V=

103.4500

Press R/S to continue. The calculator briefly displays:

END PT RD

then stops and display the running distance of the end point of the curve, such as follows:

F=
24,767.0000

Press R/S to continue. The calculator briefly displays:

END PT EL

then stops and displays the elevation of the end point of the curve, such as follows:

E=
107.4500

Press R/S to continue, and the option ends.

C. Compute Elevation at a Specified Running Distance

With this option, the user can enter any running distance and the calculator will compute the elevation on the curve at that point. The calculator does not check if the running distance is on the curve segment actually being used, so the user must check this. The calculator briefly displays:

COMP 1 EL [0–1]

although the right-hand bracket will be off screen. The calculator then stops and displays:

Y?
0.0000

If you don't want to run this option, press R/S and the program will advance to the next option. If you do want to run this option, key in 1 (or any number greater than 0), which signifies 'yes,' and press R/S. The calculator briefly displays:

ENTER RD

then stops and prompts for the RD to be entered, displaying:

X?
0.0000

Key in the running distance of the point of interest and press R/S. The calculator then briefly displays:

EL OF POINT

then stops and displays the elevation value of the selected point, such as follows:

H=
114.3690

Press R/S to continue, and the calculator prompts to see if you want to compute another elevation, briefly displaying:

AGAIN [0–1]

then stopping and displaying:

Y?
0.0000

If you want to do another point, key in 1 and press R/S. The program then prompts for the running distance (as above) and loops through the option until you decide not to do it again. If you don't want to do this option, press R/S, and the option ends.

D. Compute the Running Distances at a Specified Elevation

This option allows the user to enter an elevation and compute the running distance(s) at which it occurs. Since the curve is a parabola, there will be zero, one or two solutions. If the elevation cannot occur on the curve, i.e., the elevation is beyond the elevation of the turning point of the parabola, there will be zero answers.

If the elevation chosen is that of the turning point, there will be just one answer, which will be provided. Any other elevation on the curve will have two solutions, and each will be given.

The calculator briefly displays:

COMP 1 RD [0–1]

with the right-hand bracket off the screen. The calculator then stops and displays the prompt:

Y?
0.0000

If you want to skip this option, just press R/S and the calculator moves on to the next option. If you do want to run this option, key in 1 and press R/S. The calculator briefly displays:

ENTER EL

then stops and displays:

H?
0.0000

Key in the elevation of interest, and press R/S.

If the elevation is not on the curve, the calculator briefly displays:

THIS ELEV

then

NOT ON CURVE

and then prompts to see if you want to run the option again.

If the elevation is at the maximum or minimum point, the calculator briefly displays:

SINGLE RD

then stops and displays the running distance of the point, such as follows:

X=
24,392.9180

Press R/S/ to continue and the calculator prompts to see if you want to run the option again.

If the elevation is elsewhere on the curve, the calculator briefly displays:

FIRST RD

then stops and displays the running distance of the first solution point, such as follows:

X=
24,383.0819

Press R/S and the calculator briefly displays:

SECOND RD

then stops and displays the running distance of the second solution point, such as follows:

X=
24,795.3625

It is up to the user to decide if the points fall within the end points of the curve, and chose points that are useful.

Press R/S to continue, and the calculator then briefly displays:

AGAIN [0–1]

then stops and displays:

Y?
0.0000

To run the option again, key in and 1 and press R/S. The calculator will then prompt for the elevation to be entered, as above, and run through the option again. If you don't want to run the option again, press R/S and the option ends.

E. Step Through a Series of Running Distances to get Elevations at each

The final option allows the user to step through a series of equally-spaced points along the curve. As the start point is often at an odd running distance, this option allows the user to select a first increment, to allow the running distances to be brought to even values (e.g., exactly onto 100 ft stations), and then select a general increment to be applied successively until the end point is reached. The end point is calculated as the last point along the curve.

The option begins by displaying briefly:

STEP THRU RD

then displaying briefly:

NO–YES [0–1]

then stopping and displaying:

Y?
0.0000

If you want to run this option, key in 1 and press R/S. If not, just press R/S and the program ends. If you are running the option, the calculator briefly displays:

FIRST INCRMNT

then stops and displays:

C=
0.0000

Key in the first increment, then press R/S. The calculator briefly displays:

GENRL INCRMNT

then stops and displays:

D=
0.0000

Key in the increment to be used for all the other distances, then press R/S. The calculator then briefly displays:

RD VALUE

then stops and displays the running distance of the start point, such as follows:

U=
24,367.0000

Press R/S to continue. The calculator briefly displays:

EL VALUE

then stops and displays the elevation at the start point, such as follows:

V=
103.4500

Press R/S to continue. The calculator then loops through the following sequence, briefly displaying:

RD VALUE

then stopping and displaying the next running distance value, such as follows:

X=
24,467.0000

Press R/S to continue. The calculator briefly displays:

EL VALUE

then stops and displays the elevation value at the running distance just given, such as follows:

H=
111.2000

Press R/S to continue through this loop until the increments extend past the last point. At this stage, the last point is displayed. The calculator briefly displays:

END POINT

then briefly displays:

RD VALUE

then stops and displays the running distance of the end point, such as follows:

X=
24,767.0000

Press R/S to continue. The calculator briefly displays:

EL VALUE

then stops and displays the elevation of the end point, such as follows:

H=
107.45

Press R/S to continue. The calculator briefly displays

PROGRAM END

and then stops at the end of the program, having reset Flag 10 to its value at the time the program was started.. If R/S is pressed again, the program returns to the point whence it was called. This is not strictly necessary, unless the program was called by another program.

Sample Computations

The sample computations are based on the following general data:

Running Distance of the Point of Intersection = 24,567.0000

Elevation of the Point of Intersection = 123.4500

Length of the Curve = 400.0000

The running distance of the start point is therefore 24,367 and of the end point is 24,767.

The following tabulations show the results for different in and out gradients. The increments used for the running distance values are both 50 (both the first increment and the general increment).

Case	In Grade	Out Grade	Start EL	End EL	Max/Min RD	Max/Min EL
1	0.0200	-0.0200	119.45	119.45	24567.00	121.45
2	0.0400	-0.0600	115.45	111.45	24527.00	118.65
3	0.0600	-0.0400	111.45	115.45	24607.00	118.65
4	0.0800	-0.1000	107.45	103.45	24544.78	114.56
5	0.1000	-0.0800	103.45	107.45	24589.22	114.56
6	-0.0200	0.0400	127.45	131.45	24500.33	126.12
7	-0.0400	0.0200	131.45	127.45	24633.67	126.12
8	-0.0600	0.0800	135.45	139.45	24538.43	130.31
9	-0.0800	0.1000	139.45	143.45	24544.78	132.34
10	-0.1000	0.0600	143.45	135.45	24617.00	130.95
11	0.0200	0.0400	119.45	131.45	23967.00	115.45
12	0.0400	0.0200	115.45	127.45	25167.00	131.45
13	0.0600	0.0800	111.45	139.45	23167.00	75.45
14	0.0800	0.1000	107.45	143.45	22767.00	43.45
15	0.1000	0.0600	103.45	135.45	25367.00	153.45
16	-0.0200	-0.0400	127.45	115.45	23967.00	131.45
17	-0.0400	-0.0600	131.45	111.45	23567.00	147.45
18	-0.0600	-0.0400	135.45	115.45	25567.00	99.45
19	-0.0800	-0.1000	139.45	103.45	22767.00	203.45
20	-0.1000	-0.0800	143.45	107.45	26367.00	43.45

	Running Distance								
Case	**24367**	**24417**	**24467**	**24517**	**24567**	**24617**	**24667**	**24717**	**24767**
1	119.45	120.33	120.95	121.33	121.45	121.33	120.95	120.33	119.45
2	115.45	117.14	118.20	118.64	118.45	117.64	116.20	114.14	111.45
3	111.45	114.14	116.20	117.64	118.45	118.64	118.20	117.14	115.45
4	107.45	110.89	113.20	114.39	114.45	113.39	111.20	107.89	103.45
5	103.45	107.89	111.20	113.39	114.45	114.39	113.20	110.89	107.45
6	127.45	126.64	126.20	126.14	126.45	127.14	128.20	129.64	131.45
7	131.45	129.64	128.20	127.14	126.45	126.14	126.20	126.64	127.45
8	135.45	132.89	131.20	130.39	130.45	131.39	133.20	135.89	139.45
9	139.45	136.01	133.70	132.51	132.45	133.51	135.70	139.01	143.45
10	143.45	138.95	135.45	132.95	131.45	130.95	131.45	132.95	135.45
11	119.45	120.51	121.70	123.01	124.45	126.01	127.70	129.51	131.45
12	115.45	117.39	119.20	120.89	122.45	123.89	125.20	126.39	127.45
13	111.45	114.51	117.70	121.01	124.45	128.01	131.70	135.51	139.45
14	107.45	111.51	115.70	120.01	124.45	129.01	133.70	138.51	143.45
15	103.45	108.33	112.95	117.33	121.45	125.33	128.95	132.33	135.45
16	127.45	126.39	125.20	123.89	122.45	120.89	119.20	117.39	115.45
17	131.45	129.39	127.20	124.89	122.45	119.89	117.20	114.39	111.45
18	135.45	132.51	129.70	127.01	124.45	122.01	119.70	117.51	115.45
19	139.45	135.39	131.20	126.89	122.45	117.89	113.20	108.39	103.45
20	143.45	138.51	133.70	129.01	124.45	120.01	115.70	111.51	107.45

These values should allow the program to be tested to make sure it is working properly. The above tabulated values were calculated by spreadsheet, rather than the calculator, but the calculator results were checked against the these tabulations.

Storage Registers Used

A Difference between the incoming and outgoing gradients.

B Parameter *a* in the parabola equation.

C Elevation difference. First increment value for ‘stepping’ option.

D Running Distance of a computed point. General increment for ‘stepping’ option.

E Elevation of a computed point.

F Running Distance of a computed point.

H Elevation of a computed point, and entered elevation to have RD calculated.

I Distance along curve from start point.

L Length of the curve to be computed.

P Gradient of the incoming tangent (start grade)

Q Gradient of the outgoing tangent (end grade).

R Running Distance of the Point of Intersection.

S Elevation of the Point of Intersection.

U Running Distance of the start point.

V Elevation of the start point.

X Running Distance entered to compute elevation at that point.

Y Yes/No variable for option choices.

Statistical Registers: Not used.

Labels Used

Label **P** Length = 1383 Checksum = 6E0F

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

HP-35s Calculator Program Lambert 1

Convert Latitude and Longitude to Lambert Conformal Conic Projection Co-ordinates (SPCS)

Programmer: Dr. Bill Hazelton

Date: July, 2010. **Version**: 1.2

Line	Instruction	Display	User Instructions
N001	LBL N		LBL N
N002	CLSTK		CLEAR 5
N003	FS? 10		FLAGS 3 .0
N004	GTO N008		
N005	SF 1		FLAGS 1 1
N006	SF 10		FLAGS 1 .0
N007	GTO N009		
N008	CF 1		FLAGS 2 1
N009	LAT–LONG 2 LCC		(Key in using EQN RCL L, RCL A, etc.)
N010	PSE		PSE
N011	CL x		CLEAR 1
N012	STO F		STO F
N013	STO L		STO L
N014	STO P		STO P
N015	STO Q		STO Q
N016	STO C		STO C
N017	STO D		STO D
N018	STO G		STO G
N019	STO H		STO H
N020	6378137		a value for ellipsoid (WGS84/NAD83)
N021	STO A		STO A
N022	0.00669438		e^2 value for ellipsoid (WGS84/NAD83)
N023	STO E		STO E
N024	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
N025	PSE		PSE
N026	INPUT A		INPUT A
N027	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
N028	PSE		PSE
N029	INPUT E		INPUT E
N030	RCL E		
N031	√x		
N032	STO O		STO O
N033	CHK—NTR LAT 0		(Key in using EQN RCL C, RCL H, etc.)
N034	PSE		PSE
N035	INPUT P		INPUT P
N036	CHK—NTR LONG 0		(Key in using EQN RCL C, RCL H, etc.)
N037	PSE		PSE
N038	INPUT Q		INPUT Q

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Line	Instruction
N039	STD PARALLEL 1
N040	PSE
N041	INPUT C
N042	STD PARALLEL 2
N043	PSE
N044	INPUT D
N045	CHK—NTR E 0
N046	PSE
N047	INPUT G
N048	CHK—NTR N 0
N049	PSE
N050	INPUT H
N051	RCL P
N052	HMS→
N053	STO P
N054	RCL Q
N055	HMS→
N056	STO Q
N057	RCL C
N058	HMS→
N059	STO C
N060	RCL D
N061	HMS→
N062	STO D
N063	ENTER PT LAT
N064	PSE
N065	INPUT F
N066	ENTER PT LONG
N067	PSE
N068	INPUT L
N069	RCL L
N070	HMS→
N071	STO L
N072	RCL F
N073	HMS→
N074	STO F
****	Calculate m_2
N075	RCL D
N076	STO Z
N077	XEQ N209
N078	RCL Z
N079	STO S
****	Calculate m_1
N080	RCL C
N081	STO Z

Line	Instruction
N082	XEQ N209
N083	RCL Z
N084	STO R
****	Calculate t_1
N085	RCL C
N086	STO Z
N087	XEQ N222
N088	RCL Z
N089	STO U
****	Calculate t_2
N090	RCL D
N091	STO Z
N092	XEQ N222
N093	RCL Z
N094	STO V
****	Calculate t_0
N095	RCL P
N096	STO Z
N097	XEQ N222
N098	RCL Z
N099	STO T
****	Calculate n
N100	RCL R
N101	LN
N102	RCL S
N103	LN
N104	—
N105	RCL U
N106	LN
N107	RCL V
N108	LN
N109	—
N110	÷
N111	STO N
****	Calculate F
N112	RCL R
N113	RCL÷ N
N114	RCL U
N115	RCL N
N116	y^x
N117	÷
N118	STO J
****	Calculate r_0
N119	RCL A
N120	RCL× J

Line	Instruction
N121	RCL T
N122	RCL N
N123	y^x
N124	×
N125	STO M
****	Calculate θ
N126	RCL L
N127	RCL— Q
N128	RCL× N
N129	STO K
****	Calculate r
N130	RCL F
N131	STO Z
N132	XEQ N222
N133	RCL Z
N134	RCL N
N135	y^x
N136	RCL× J
N137	RCL× A
N138	STO B
****	Calculate X and Y
N139	RCL K
N140	SIN
N141	RCL× B
N142	STO X
N143	RCL M
N144	RCL K
N145	COS
N146	RCL× B
N147	—
N148	STO Y
N149	RCL G
N150	STO+ X
N151	RCL H
N152	STO+ Y
****	Calculate scale factor
N153	1
N154	RCL F
N155	SIN
N156	x^2
N157	RCL× E
N158	—
N159	$\sqrt{x}$
N160	RCL A
N161	x < > y

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Line	Instruction
N162	÷
N163	RCL F
N164	COS
N165	×
N166	RCL÷ N
N167	RCL÷ B
N168	1/x
N169	STO W
****	Calculate γ
N170	RCL K
N171	→HMS
N172	STO K
****	Show results
N173	RESULTS
N174	PSE
N175	EASTING
N176	PSE
N177	VIEW X
N178	NORTHING
N179	PSE
N180	VIEW Y
N181	GRID CONV
N182	PSE
N183	VIEW K
N184	PT SCALE FACT
N185	PSE
N186	VIEW W
****	Check for next pt.
N187	0
N188	STO Z
N189	NEXT PT [0—1]
N190	PSE
N191	INPUT Z

Line	Instruction
N192	RCL Z
N193	x = 0 ?
N194	GTO N204
N195	NEW ZONE [0—1]
N196	PSE
N197	0
N198	STO Z
N199	INPUT Z
N200	RCL Z
N201	x = 0 ?
N202	GTO N063
N203	GTO N024
****	End of program
N204	PROGRAM END
N205	PSE
N206	FS? 1
N207	CF 10
N208	RTN
****	Subroutines
****	Compute m
N209	RCL Z
N210	COS
N211	RCL Z
N212	SIN
N213	x^2
N214	RCL× E
N215	1
N216	x < > y
N217	—
N218	√x
N219	÷
N220	STO Z
N221	RTN

Line	Instruction
****	Compute t
N222	1
N223	RCL Z
N224	SIN
N225	RCL× O
N226	—
N227	RCL Z
N228	SIN
N229	RCL× O
N230	1
N231	+
N232	÷
N233	RCL O
N234	2
N235	÷
N236	y^x
N237	45
N238	RCL Z
N239	2
N240	÷
N241	—
N242	TAN
N243	x < > y
N244	÷
N245	STO Z
N246	RTN

Notes

(1) The program should be run in RPN mode, as results in ALG mode are unknown.

(2) Latitudes and longitudes should be entered in HP notation, i.e., DDD.MMSS. The grid convergence is displayed in HP notation.

(3) The program may be used for any Lambert Conformal Conic projection, if the appropriate parameters are known. Similarly, any ellipsoid may be used, if its a and e^2 parameters are known. Parameters for a wide range of ellipsoids and all SPCS Lambert zones are included at the end of this document.

(4) Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in North America. It is critical to enter the correct sign in calculator when entering values.

(5) Lines with **** are comments only, and should not be entered into the calculator. They are there to make program entry a little easier.

(6) This program is long and often appears to be a stream of meaningless commands. This means that it may be more prone to errors when being entered. It is suggested that the program be entered using the given constants, tested (and the checksum checked), and when it is satisfactory, the values for the zone that are set to zero at the start of the program can be changed to those most suitable for the bulk of the expected work. See the **Localization** section at the end of the document.

(7) When working in SPCS 1927, there are some small differences in Northings between this program and the NGS conversion program. This may be caused by a different method of computing the distance from the pole in days gone by (see the discussion in the **Theory** section). The differences are small (less than 0.25 inch in all places tested thus far) and appear to be larger the further the point is from the pole. The conversions in SPCS 1983 agree to 0.001 m, which is the finest value the NGS program provides.

(8) The scale factor is exactly 1.0000 when the point is on the standard parallels. It is less than 1.0000 between the standard parallels, and greater than 1.0000 outside the standard parallels.

Theory

Converting from geographical co-ordinates (latitude and longitude) to cartesian co-ordinates on a Lambert Conformal Conic projection is a straightforward transformation, if somewhat long-winded. The following is derived from Snyder's *Map Projections – A Working Manual*.

If we are working on the ellipsoid, which we really have to do for most cases and certainly for the SPCS, we need the following information in order to convert geographical co-ordinates into grid co-ordinates: a, e^2, ϕ_1, ϕ_2, ϕ_0, λ_0, ϕ and λ. We know a and e^2 for the particular ellipsoid being used, and the fixed latitude and longitude values are already determined for every SPCS zone. The details are provided for all SPCS Lambert Zones at the end of this discussion.

(Note that the SPCS 27 has co-ordinates in US Survey feet, and uses the Clarke 1866 ellipsoid. The SPCS 83 has co-ordinates in meters, and uses the GRS80 spheroid, which effectively is the same as WGS-84. Some states have either the US Survey foot or the International foot as alternative distance units; check which one is in use in the state you are working in at any particular time. Note that there is a datum shift between the two systems (1927 and 1983) as well, and that you cannot really do a direct linear shift between them.

With the basic information, we can compute the co-ordinates as follows:

$$x = r \sin \theta$$

$$y = r_0 - r \cos \theta$$

then use

$$E = E_0 + x$$

$$N = N_0 + y$$

where the following formulae are used. Note that subscripts 0, 1 or 2 imply the use of specific values of ϕ in the formulae for r, m and t at the appropriate places.

The example of r_0 is given below. The terms t, m and F are used only to simplify computation and do not imply any real (or readily apparent) physical quantity or value.

$$r = a\,F\,t^n$$

$$\theta = n\,(\lambda - \lambda_0)$$

$$r_0 = a\,F\,t_0^{\,n}$$

$$n = \frac{\ln m_1 - \ln m_2}{\ln t_1 - \ln t_2} \qquad \text{(constant of the projection or cone)}$$

$$m = \frac{\cos\phi}{\sqrt{1 - e^2 \sin^2\phi}}$$

$$t = \frac{\tan\left(\frac{\pi}{4} - \frac{\phi}{2}\right)}{\left(\dfrac{1 - e\sin\phi}{1 + e\sin\phi}\right)^{\frac{e}{2}}}$$

or

$$t = \sqrt{\left(\left(\frac{1 - \sin\phi}{1 + \sin\phi}\right)\left(\frac{1 + e\sin\phi}{1 - e\sin\phi}\right)^{e}\right)}$$

$$F = \frac{m_1}{n\,t_1^{\,n}}$$

$$k = \frac{n\ r}{\nu\cos\phi} \qquad \text{(scale factor at the latitude } \phi\text{)}$$

where ν = the radius of the ellipsoid at the parallel of latitude ϕ

$$\gamma = \theta \qquad \text{(grid convergence at the point)}$$

Note that n, F and r_0 are constants for a particular map or SPCS zone and only need to be computed once.

When computing with these values, you will get results slightly different from those in the published tables for SPCS 27 (although many of these are out of print). The discrepancy is of the order of 20-30 meters, which is relatively small considering the r distances coming from the pole, and the relative error in a SPCS zone is quite small. The reason for this is that for calculating convenience 70 years ago, when the tables were developed, the tables were, in effect, calculated using the following variant of t, i.e.

$$t = \tan\left(\frac{\pi}{2} - \frac{\phi_g}{2}\right)$$

where ϕ_g is the geocentric latitude, also able to be expressed by

$$\phi_g = \arctan((1 - e^2) \tan \phi)$$

t is actually the cotangent of half the co-latitude of the conformal latitude, χ, which is derived by assuming that a sphere was used as a kind of substitute for the ellipsoid in some calculations (see Snyder for greater details of this, if you are interested). As it happens, the expansions for χ and ϕ_g are numerically very nearly the same. However, the small differences still exist in the SPCS 27 tables. There are other smaller differences caused by the slightly lower precision of older desk-top calculating machines, compared to modern equipment, and the adaptation of formulae to suit these machines. However, these discrepancies are pretty small. This program does not take account of these differences, and so there are small differences between the results here (in the Northings only) and the results from the NGS conversion program. These appear to be less than 0.25 inch anywhere that has been tested thus far.

As you can see from the tables below, most of the SPCS Lambert zones adopt ϕ_0 as having a Y or N value of zero. It is chosen so as to be well south of the limits of the zone. For most of the Lambert zones, the central meridian gets a value of 2,000,000 feet (SPCS 27), and a range of meter values for SPCS 83. See the tables below for exact data.

Sample Computations

Example 1

Using the SPCS 1983 (a = 6,378,137 m, e^2 = 0.006 694 3800), the following results are obtained.

Ohio North Zone, 3401: True Origin: ϕ_0 = 39° 40', λ_0 = –82° 30';
Standard Parallels: ϕ_1 = 40° 26', ϕ_2 = 41° 42';
False Origin: E_0 = 600,000.000 m, N_0 = 0.000 m.

Latitude = 40° 05' 30" Longitude = –83° 10' 20"

Easting (E) = 542,668.995 m Northing (N) = 47,416.966 m

Grid Convergence (γ) = –0° 26' 29.82" Point Scale Factor (k) = 1.000 082 97

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Example 2

Using the SPCS 1927 (a = 20,925,832.2 ft, e^2 = 0.006 768 66), the following results are obtained.

Ohio North Zone, 3401: True Origin: ϕ_0 = 39° 40', λ_0 = –82° 30';
Standard Parallels: ϕ_1 = 40° 26', ϕ_2 = 41° 42';
False Origin: E_0 = 2,000,000.000 ft, N_0 = 0.000 ft.

Latitude = 40° 05' 30" Longitude = –83° 10' 20"

Easting (E) = 1,811,901.577 ft Northing (N) = 155,564.399 ft

Grid Convergence (γ) = –0° 26' 29.82" Point Scale Factor (k) = 1.000 082 97

Note: the NGS conversion program gave the same results, except for the Northing, which it gave as 155,564.393, a difference of 0.006 ft (about 0.07 inches). Testing other points in this zone indicate a consistent difference of about this amount. This may be because of the different method of computing the distances from the pole (r and r_0) in earlier computations of the zones.

Example 3

Using the SPCS 1927 (a = 20,925,832.2 ft, e^2 = 0.006 768 66), the following results are obtained.

California III Zone (0403), SPCS 1927, ϕ_0 = 36° 30', λ_0 = –120° 30';
ϕ_1 = 37° 04', ϕ_2 = 38° 26';
E_0 = 2,000,000.000 ft, N_0 = 0.000 ft.

Latitude = 37° 25' 40" Longitude = –119° 45' 20"

Easting (E) = 2,216,169.136 ft Northing (N) = 338,664.251 ft

Grid Convergence (γ) = 0° 27' 20.8" Point Scale Factor (k) = 0.999 945 01

Note: The NGS conversion program give the same results, except for the Northing, which it gives as 338,664.238 ft, a difference of 0.013 ft (about 0.16 inches). Testing other points in this zone indicate a consistent difference of about this amount. This may be because of the different method of computing the distances from the pole (r and r_0) in earlier computations of the zones.

Example 4

Using the SPCS 1983 (a = 6,378,137 m, e^2 = 0.006 694 3800), the following results are obtained.

California III Zone (0403), SPCS 1983, ϕ_0 = 36° 30', λ_0 = –120° 30';
ϕ_1 = 37° 04', ϕ_2 = 38° 26';
E_0 = 2,000,000.000 m, N_0 = 500,000.000 m.

Latitude = 37° 25' 40" Longitude = –119° 45' 20"

Easting (E) = 2,065,886.861 m Northing (N) = 603,227.485 m

Grid Convergence (γ) = 0° 27' 20.8" Point Scale Factor (k) = 0.999 945 01

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Running the Program

Press XEQ N, then press ENTER to start the program. The calculator briefly displays LAT—LONG 2 LCC, then briefly shows CHECK—ENTER A. The program then stops and displays the prompt for entering the semi-major axis value, while displaying the current default value:

A?
6,378,137.0000 (This is for GRS80/WGS84/NAD83)

If you are happy with this value for the semi-major axis of the ellipsoid, press R/S to continue. Otherwise. Key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHECK—ENTER E. The program then stops and displays the prompt for entering the eccentricity of the ellipsoid, e:

E?
0.00669438 (This is for GRS80/WGS84/NAD83)

If this value for the eccentricity is correct, press R/S to continue. Otherwise, key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHK—NTR LAT 0. The program then stops and displays the prompt for entering the origin latitude for the co-ordinate, ϕ_0:

P?
0.0000

Key in the correct latitude in HP notation (DDD.MMSS), and press R/S to continue. In this case, key in 39.40 for Ohio North.

The calculator briefly displays CHK—NTR LONG 0. The program then stops and displays the prompt for entering the longitude of the central meridian of the projection, λ_0. Note that in the western hemisphere, this will be a negative value, and should be in HP notation (DDD.MMSS).

Q?
0.0000

Key in the correct longitude, in HP notation and remembering the sign, then press R/S to continue. In this case, key in –82.30 for Ohio North

The calculator briefly displays STD PARALLEL 1. The program then stops and displays the prompt for entering the latitude of one of the standard parallels for the projection, ϕ_1. The value should be entered in HP notation.

C?
0.000000

Key in the correct value and press R/S to continue. In this case, key in 40.26 and press R/S to continue.

The calculator briefly displays STD PARALLEL 2. The program then stops and displays the prompt for entering the latitude of the other standard parallel for the projection, ϕ_2. The value should be entered in HP notation.

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

D?
0.000000

Key in the correct value and press R/S to continue. In this case, key in 41.42 and press R/S to continue.

The calculator briefly displays CHK—NTR E 0. The program then stops and prompts for the false easting value, or the easting offset. This is the value of the easting at the central meridian (λ_0), denoted E_0.

G?
0.0000

Key in the correct value, and press R/S to continue. In this case, key in 600000.0 and press R/S.

The calculator briefly displays CHK—NTR N 0. The program then stops and prompts for entry of the false northing value, or the northing offset. This is the value of the northing co-ordinate at ϕ_0, λ_0.

H?
0.0000 (This is for UTM)

If this is the correct value (for some zones, it is zero), press R/S to continue. If a different value is desired, key in the value and press R/S. In this case, just press R/S. This is the N_0 value for Ohio North.

The calculator briefly displays ENTER PT LAT. The program stops and displays the prompt for entering the latitude of the point to be converted. This should be entered in HP notation.

F?
0.0000

Key in the latitude of the point in HP notation and press R/S to continue. In this case, key in 40.0530 and press R/S.

The calculator briefly displays ENTER PT LONG. The program then stops and displays the prompt for entering the longitude of the point to be converted. This should be entered in HP notation.

L?
0.0000

Key in the longitude of the point in HP notation and press R/S to continue. In this case, key in —83.1020 and press R/S/

The program displays RUNNING for a short while, then displays RESULTS briefly, followed by EASTING briefly. The program then stops and displays the easting value of the point. In this case, the calculator displays:

X=
542,662.995

This is the easting of the point, in this case in meters. Press R/S to continue. The calculator briefly displays NORTHING, then stops and displays the northing value of the point. In this case, the calculator displays:

Y=
47,416.966

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

This is the northing of the point, in this case in meters. Press R/S to continue. The calculator briefly displays GRID CONV, then stops and displays the grid convergence value in HP notation. In this case, the calculator displays:

K=
—0.262981976

This is the grid convergence in HP notation, and is —0° 26' 29".82 in more conventional notation. Press R/S to continue. The calculator briefly displays PT SCALE FACT, then stops and displays the point scale factor of the point on the Transverse Mercator projection. In this case, the calculator displays:

W=
1.00008297

This is the point scale factor. Press R/S to continue.

You now have the choice of running one or more additional points. The calculator briefly displays NEXT PT [0—1], then stops and displays the prompt for answering questions:

Z?
0.0000

If you want to quit the program, just press R/S. If you want to enter more points, key in 1 and press R/S. In this case, the calculator then prompts to see if you want to use the same parameters. The calculator briefly displays NEW ZONE [0—1], then stops at the question prompt:

Z?
0.0000

If you want to go to a new zone, key in 1 and press R/S, and the calculator will take you to the point where you can change any of the values (Point A above), starting with the ellipsoid parameters. If you want to work in the same zone already entered, just press R/S, and the program will take you to "Point B" and prompt for the latitude of the point to be converted, and continue from there. You can go around the program as many times as necessary.

When you choose to end the program, the calculator briefly displays PROGRAM END and then comes to an end, returning to the point from which it was called, or to normal operations.

Storage Registers Used

A Semi-major axis of the ellipsoid being used, a

B r, distance from the pole to the point

C ϕ_1, one of the two standard parallels of the projection

D ϕ_2, one of the two standard parallels of the projection

E Eccentricity of the ellipsoid, e^2

F ϕ, latitude of the point to be converted

G E_0, the false easting or easting offset

H N_0, the false northing, or northing offset, at ϕ_0, λ_0

J F, an internal computed value

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

K θ, the angle between the line from the pole to the point, and the central meridian, also the grid convergence

L λ, longitude of the point to be converted

M r_0, distance from the pole to ϕ_0

N n, the constant of the projection or cone

O e, the square root of the eccentricity of the ellipsoid.

P ϕ_0, the latitude of the co-ordinate origin on the projection

Q λ_0, the central meridian of the projection

R m_1, an internal computed value

S m_2, an internal computed value

T t_0, an internal computed value

U t_1, an internal computed value

V t_2, an internal computed value

W point scale factor, k

X Easting co-ordinate of converted point

Y Northing co-ordinate of converted point

Z Response variable for checking if another point

Statistical Registers: not used

Labels Used

Label **N** Length = 989 Checksum = F78C

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Parameters for the Computations

State Plane Co-ordinate System (SPCS) 1983

Several US states use the Lambert Conformal Conic projection for SPCS 1983. The various parameters for each zone in the 1983 system are given in the table below. Use these parameters with the program, together with the GRS80/WGS84/NAD83 ellipsoid parameters, in meters.

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Alaska						
Zone 10	51° 50'	53° 50'	176° 00'	51° 00'	1,000,000.00	0.00
Arkansas						
North	34° 56'	36° 14'	92° 00'	34° 20'	400,000.00	0.00
South	33° 18'	34° 46'	92° 00'	32° 40'	400,000.00	400,000.00
California						
I	40° 00'	41° 40'	122° 00'	39° 20'	2,000,000.00	500,000.00
II	38° 20'	39° 50'	122° 00'	37° 40'	2,000,000.00	500,000.00
III	37° 04'	38° 26'	120° 30'	36° 30'	2,000,000.00	500,000.00
IV	36° 00'	37° 15'	119° 00'	35° 20'	2,000,000.00	500,000.00
V	34° 02'	35° 28'	118° 00'	33° 30'	2,000,000.00	500,000.00
VI	32° 47'	33° 53'	116° 15'	32° 10'	2,000,000.00	500,000.00
Colorado						
North	39° 43'	40° 47'	105° 30'	39° 20'	914,401.83	304,800.61
Central	38° 27'	39° 45'	105° 30'	37° 50'	914,401.83	304,800.61
South	37° 14'	38° 26'	105° 30'	36° 40'	914,401.83	304,800.61
Connecticut	41° 12'	41° 52'	72° 45'	40° 50'	304800.61	152400.30
Florida						
North	29° 35'	30° 45'	84° 30'	29° 00'	600000.00	0.00
Iowa						
North	42° 04'	43° 16'	93° 30'	41° 30'	1500000.00	1000000.00
South	40° 37'	41° 47'	93° 30'	40° 00'	500000.00	0.00
Kansas						
North	38° 43'	39° 47'	98° 00'	38° 20'	400000.00	0.00
South	37° 16'	38° 34'	98° 30'	36° 40'	400000.00	400000.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Kentucky						
North	37° 58'	38° 58'	84° 15'	37° 30'	500,000.00	0.00
South	36° 44'	37° 56'	85° 45'	36° 20'	500,000.00	500,000.00
Louisiana						
North	31° 10'	32° 40'	92° 30'	30° 30'	1,00,0000.00	0.00
South	29° 18'	30° 42'	91° 20'	28° 30'	1,000,000.00	0.00
Offshore	26° 10'	27° 50'	91° 20'	25° 30'	1,000,000.00	0.00
Maryland	38° 18'	39° 27'	77° 00'	37° 40'	400,000.00	0.00
Massachusetts						
Mainland	41° 43'	42° 41'	71° 30'	41° 00'	200,000.00	750,000.00
Island	41° 17'	41° 29'	70° 30'	41° 00'	500,000.00	0.00
Michigan						
North	45° 29'	47° 05'	87° 00'	44° 47'	8,000,000.00	0.00
Central	44° 11'	45° 42'	84° 22'	43° 19'	6,000,000.00	0.00
South	42° 06'	43° 40'	84° 22'	41° 30'	4,000,000.00	0.00
Minnesota						
North	47° 02'	48° 38'	93° 06'	46° 30'	800,000.00	100000.00
Central	45° 37'	47° 03'	94° 15'	45° 00'	800,000.00	100,000.00
South	43° 47'	45° 13'	94° 00'	43° 00'	800,000.00	100,000.00
Montana	45° 00'	49° 00'	109° 30'	44° 15'	600,000.00	0.00
Nebraska	40° 00'	43° 00'	100° 00'	39° 50'	500,000.00	0.00
New York						
Long Island	40° 40'	41° 02'	74° 00'	40° 10'	300,000.00	0.00
North Carolina	34° 20'	36° 10'	79° 00'	33° 45'	609,601.22	0.00
North Dakota						
North	47° 26'	48° 44'	100° 30'	47° 00'	600,000.00	0.00
South	46° 11'	47° 29'	100° 30'	45° 40'	600,000.00	0.00
Ohio						
North	40° 26'	41° 42'	82° 30'	39° 40'	600,000.00	0.00
South	38° 44'	40° 02'	82° 30'	38° 00'	600,000.00	0.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Oklahoma						
North	35° 34'	36° 46'	98° 00'	35° 00'	600,000.00	0.00
South	33° 56'	35° 14'	98° 00'	33° 20'	600,000.00	0.00
Oregon						
North	44° 20'	46° 00'	120° 30'	43° 40'	2,500,000.00	0.00
South	42° 20'	44° 00'	120° 30'	41° 40'	1,500,000.00	0.00
Pennsylvania						
North	40° 53'	41° 57'	77° 45'	40° 10'	600,000.00	0.00
South	39° 56'	40° 58'	77° 45'	39° 20'	600,000.00	0.00
Puerto Rico and Virgin Islands						
1	18° 02'	18° 26'	66° 26'	17° 50'	200,000.00	200,000.00
2 (St. Croix)	18° 02'	18° 26'	66° 26'	17° 50'		
Samoa	–14° 16'	–14° 16'	170° 00'			
South Carolina	32° 30'	34° 50'	81° 00'	31° 50'	609,600.00	0.00
South Dakota						
North	44° 25'	45° 41'	100° 00'	43° 50'	600,000.00	0.00
South	42° 50'	44° 24'	100° 20'	42° 20'	600,000.00	0.00
Tennessee	35° 15'	36° 25'	86° 00'	34° 20'	600,000.00	0.00
Texas						
North	34° 39'	36° 11'	101° 30'	34° 00'	200,000.00	1,000,000.00
North central	32° 08'	33° 58'	98° 30'	31° 40'	600,000.00	2,000,000.00
Central	30° 07'	31° 53'	100° 20'	29° 40'	700,000.00	3,000,000.00
South central	28° 23'	30° 17'	99° 00'	27° 50'	600,000.00	4,000,000.00
South	26° 10'	27° 50'	98° 30'	25° 40'	300,000.00	5,000,000.00
Utah						
North	40° 43'	41° 47'	111° 30'	40° 20'	500,000.00	1,000,000.00
Central	39° 01'	40° 39'	111° 30'	38° 20'	500,000.00	2,000,000.00
South	37° 13'	38° 21'	111° 30'	36° 40'	500,000.00	3,000,000.00
Virginia						
North	38° 02'	39° 12'	78° 30'	37° 40'	3,500,000.00	2,000,000.00
South	36° 46'	37° 58'	78° 3'0	36° 20'	3,500,000.00	1,000,000.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Washington						
North	47° 30'	48° 44'	120° 50'	47° 00'	500,000.00	0.00
South	45° 50'	47° 20'	120° 30'	45° 20'	500,000.00	0.00
West Virginia						
North	39° 00'	40° 15'	79° 30'	38° 30'	600,000.00	0.00
South	37° 29'	38° 53'	81° 00'	37° 00'	600,000.00	0.00
Wisconsin						
North	45° 34'	46° 46'	90° 00'	45° 10'	600,000.00	0.00
Central	44° 15'	45° 30'	90° 00'	43° 50'	600,000.00	0.00
South	42° 44'	44° 04'	90° 00'	42° 00'	600,000.00	0.00

State Plane Co-ordinate System (SPCS) 1927

Several US states used the Lambert Conformal Conic projection for SPCS 1927. The various parameters for each zone in the 1927 system are given in the table below. Use these parameters with the program, together with the Clarke 1866 ellipsoid in feet.

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Alaska						
Zone 10	51° 50'	53° 50'	176° 00'	51° 00'	3000000.00	0.00
Arkansas						
North	34° 56'	36° 14'	92° 00'	34° 20'	2000000.00	0.00
South	33° 18'	34° 46'	92° 00'	32° 40'	2000000.00	0.00
California						
I	40° 00'	41° 40'	122° 00'	39° 20'	2000000.00	0.00
II	38° 20'	39° 50'	122° 00'	37° 40'	2000000.00	0.00
III	37° 04'	38° 26'	120° 30'	36° 30'	2000000.00	0.00
IV	36° 00'	37° 15'	119° 00'	35° 20'	2000000.00	0.00
V	34° 02'	35° 28'	118° 00'	33° 30'	2000000.00	0.00
VI	32° 47'	33° 53'	116° 15'	32° 10'	2000000.00	0.00
VII	33° 52'	34° 25"	118° 20'	34° 08'	4186692.58	4160926.74

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Colorado						
North	39° 43'	40° 47'	105° 30'	39° 20'	2000000.00	0.00
Central	38° 27'	39° 45'	105° 30'	37° 50'	2000000.00	0.00
South	37° 14'	38° 26'	105° 30'	36° 40'	2000000.00	0.00
Connecticut	41° 12'	41° 52'	72° 45'	40° 50'	600000.00	0.00
Florida						
North	29° 35'	30° 45'	84° 30'	29° 00'	2000000.00	0.00
Iowa						
North	42° 04'	43° 16'	93° 30'	41° 30'	2000000.00	0.00
South	40° 37'	41° 47'	93° 30'	40° 00'	2000000.00	0.00
Kansas						
North	38° 43'	39° 47'	98° 00'	38° 20'	2000000.00	0.00
South	37° 16'	38° 34'	98° 30'	36° 40'	2000000.00	0.00
Kentucky						
North	37° 58'	38° 58'	84° 15'	37° 30'	2000000.00	0.00
South	36° 44'	37° 56'	85° 45'	36° 20'	2000000.00	0.00
Louisiana						
North	31° 10'	32° 40'	92° 30'	30° 40'	2000000.00	0.00
South	29° 18'	30° 42'	91° 20'	28° 40'	2000000.00	0.00
Offshore	26° 10'	27° 50'	91° 20'	25° 40'	2000000.00	0.00
Maryland	38° 18'	39° 27'	77° 00'	37° 50'	800000.00	0.00
Massachusetts						
Mainland	41° 43'	42° 41'	71° 30'	41° 00'	600000.00	0.00
Island	41° 17'	41° 29'	70° 30'	41° 00'	200000.00	0.00
Michigan (current)						
North	45° 29'	47° 05'	87° 00'	44° 47'	2000000.00	0.00
Central	44° 11'	45° 42'	84° 20'	43° 19'	2000000.00	0.00
South	42° 06'	43° 40'	84° 20'	41° 30'	2000000.00	0.00
Minnesota						
North	47° 02'	48° 38'	93° 06'	46° 30'	2000000.00	0.00
Central	45° 37'	47° 03'	94° 15'	45° 00'	2000000.00	0.00
South	43° 47'	45° 13'	94° 00'	43° 00'	2000000.00	0.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Montana						
North	47° 51'	48° 43'	109° 30'	47° 00'	2000000.00	0.00
Central	46° 27'	47° 53'	109° 30'	45° 50'	2000000.00	0.00
South	44° 52'	46° 24'	109° 30'	44° 00'	2000000.00	0.00
Nebraska						
North	41° 51'	42° 49'	100° 00'	41° 20'	2000000.00	0.00
South	40° 17'	41° 43'	99° 30'	39° 40'	2000000.00	0.00
New York						
Long Island	40° 40'	41° 02'	74° 00'	40° 30'	2000000.00	100000.00
North Carolina	34° 20'	36° 10'	79° 00'	33° 45'	2000000.00	0.00
North Dakota						
North	47° 26'	48° 44'	100° 30'	47° 00'	2000000.00	0.00
South	46° 11'	47° 29'	100° 30'	45° 40'	2000000.00	0.00
Ohio						
North	40° 26'	41° 42'	82° 30'	39° 40'	2000000.00	0.00
South	38° 44'	40° 02'	82° 30'	38° 00'	2000000.00	0.00
Oklahoma						
North	35° 34'	36° 46'	98° 00'	35° 00'	2000000.00	0.00
South	33° 56'	35° 14'	98° 00'	33° 20'	2000000.00	0.00
Oregon						
North	44° 20'	46° 00'	120° 30'	43° 40'	2000000.00	0.00
South	42° 20'	44° 00'	120° 30'	41° 40'	2000000.00	0.00
Pennsylvania						
North	40° 53'	41° 57'	77° 45'	40° 10'	2000000.00	0.00
South	39° 56'	40° 58'	77° 45'	39° 20'	2000000.00	0.00
Puerto Rico and Virgin Islands						
1	18° 02'	18° 26'	66° 26'	17° 50'	500000.00	0.00
2 (St. Croix)	18° 02'	18° 26'	66° 26'	17° 50'	500000.00	100000.00
Samoa	-14° 16'	-14° 16'	170° 00'		500000.00	0.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
South Carolina						
North	33° 46'	34° 58'	81° 00'	33° 00'	2000000.00	0.00
South	32° 20'	33° 40'	81° 00'	31° 50'	2000000.00	0.00
South Dakota						
North	44° 25'	45° 41'	100° 00'	43° 50'	2000000.00	0.00
South	42° 50'	44° 24'	100° 20'	42° 20'	2000000.00	0.00
Tennessee	35° 15'	36° 25'	86° 00'	34° 40'	2000000.00	100000.00
Texas						
North	34° 39'	36° 11'	101° 30'	34° 00'	2000000.00	0.00
North central	32° 08'	33° 58'	97° 30'	31° 40'	2000000.00	0.00
Central	30° 07'	31° 53'	100° 20'	29° 40'	2000000.00	0.00
South central	28° 23'	30° 17'	99° 00'	27° 50'	2000000.00	0.00
South	26° 10'	27° 50'	98° 30'	25° 40'	2000000.00	0.00
Utah						
North	40° 43'	41° 47'	111° 30'	40° 20'	2000000.00	0.00
Central	39° 01'	40° 39'	111° 30'	38° 20'	2000000.00	0.00
South	37° 13'	38° 21	111° 30'	36° 40'	2000000.00	0.00
Virginia						
North	38° 02'	39° 12'	78° 30'	37° 40'	2000000.00	0.00
South	36° 46'	37° 58'	78° 30'	36° 20'	2000000.00	0.00
Washington						
North	47° 30'	48° 44'	120° 50'	47° 00'	2000000.00	0.00
South	45° 50'	47° 20'	120° 30'	45° 20'	2000000.00	0.00
West Virginia						
North	39° 00'	40° 15'	79° 30'	38° 30'	2000000.00	0.00
South	37° 29'	38° 53'	81° 00'	37° 00'	2000000.00	0.00
Wisconsin						
North	45° 34'	46° 46'	90° 00'	45° 10'	2000000.00	0.00
Central	44° 15'	45° 30'	90° 00'	43° 50'	2000000.00	0.00
South	42° 44'	44° 04'	90° 00'	42° 00'	2000000.00	0.00

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

Ellipsoids

There are a range of ellipsoids in common or former use. The table below has the a and e^2 values for a number of common (and less common) ellipsoids.

Ellipsoid	a Semi-major Axis	e^2 Eccentricity
GRS80–WGS94–NAD83	6378137 m	0.006 694 38
Clarke 1866 (NAD27)	6378206.4 m	0.006 768 66
Clarke 1866 (NAD27)	20925832.2 ft	0.006 768 66
ANS (Australian)	6378160 m	0.006 694 541 855
Airy 1830	6377563.4 m	0.006 670 54
Bessel 1841	6377397.16 m	0.006 674 372
Clarke 1880	6378249.15 m	0.006 803 511
Everest 1830	6377276.35 m	0.006 637 847
Fischer 1960 (Mercury)	6378166 m	0.006 693 422
Fischer 1968	6378150 m	0.006 693 422
Hough 1956	6378270 m	0.006 722 67
International	6378388 m	0.006 722 67
Krassovsky 1940	6378245 m	0.006 693 422
South American 1960	6378160 m	0.006 694 542
GRS 1967	6378160 m	0.006 694 605
GRS 1975	6378140 m	0.006 694 385
WGS 60	6378165 m	0.006 693 422
WGS 66	6378145 m	0.006 694 542
WGS 72	6378135 m	0.006 694 317 778
WGS 84	6378137 m	0.006 694 38

Localization

If it is intended to do most conversions in the one SPCS zone, then the parameters for that zone can be coded into the program. When such a program is run, the program will prompt for the values (which allows the user to work in a different zone, as needed), but will display and store the regular values for the chosen zone. These can also be changed by changing the program, if a series of points on a different zone are to be converted.

The code required to 'hardwire' zone-specific values into the program is given below, based on a specific zone. If we were going to use California Zone III in SPCS 1983, its parameters are:

ϕ_0 = 36° 30' λ_0 = –120° 30' a = 6378137 m e^2 = 0.006 694 38

ϕ_1 = 37° 04' ϕ_2 = 38° 26' E_0 = 2,000,000.000 m N_0 = 500,000.000 m

Latitude/Longitude to Lambert Conformal Conic Co-ordinates

The resulting code would be as follows, with the rest of the code left out. Note that the angular values are entered in HP notation (DDD.MMSS), as the program converts everything for internal use later.

Line	Instruction	Display	User Instructions
N001	LBL N		
………	……………		
N011	CL x		
N012	STO F		
N013	STO L		
N014	36.3		ϕ_0 value of zone
N015	STO P		
N016	—120.30		λ_0 value of zone
N017	STO Q		
N018	37.04		ϕ_1 value of zone
N019	STO C		
N020	38.26		ϕ_2 value of zone
N021	STO D		
N022	2000000.0		E_0 value of zone
N023	STO G		
N024	500000.0		N_0 value of zone
N025	STO H		
N026	6378137		a value for ellipsoid (WGS84/NAD83)
N027	STO A		
N028	0.00669438		e^2 value for ellipsoid (WGS84/NAD83)
N029	STO E		
N030	CHECK–ENTER A		
N031	PSE		
N032	INPUT A		
	…………………		
	…………………		

This will change subsequent line numbers (they will be 6 greater than before), as well as the program length and checksum, but the program should otherwise be unaffected and should run correctly.

Use the values for your preferred zone, and everything should be fine.

Reference

SNYDER, J.P., 1987. *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper 1395. Washington: US Government Printing Office.

Revisions

November, 2009. Line N081 corrected.

July, 2010. Line N220 corrected.

October, 2011. Standard parallel value for Alaska corrected.

Convert Lambert Conformal Conic Projection Co-ordinates (SPCS) to Latitude and Longitude

Programmer: Dr. Bill Hazelton

Date: September, 2010. **Version**: 1.1 **Mnemonic**: **L** for **L**ambert to **L**atitude and **L**ongitude

Line	Instruction	Display	User Instructions
L001	LBL L		LBL L
L002	CLSTK		CLEAR 5
L003	FS? 10		FLAGS 3 .0
L004	GTO L008		
L005	SF 1		FLAGS 1 1
L006	SF 10		FLAGS 1 .0
L007	GTO L009		
L008	CF 1		FLAGS 2 1
L009	LCC 2 LAT–LONG		(Key in using EQN RCL L, RCL C, etc.)
L010	PSE		PSE
L011	CL x		CLEAR 1
L012	STO X		STO X
L013	STO Y		STO Y
L014	STO P		STO P
L015	STO Q		STO Q
L016	STO C		STO C
L017	STO D		STO D
L018	STO G		STO G
L019	STO H		STO H
L020	6378137		a value for ellipsoid (WGS84/NAD83)
L021	STO A		STO A
L022	0.00669438		e^2 value for ellipsoid (WGS84/NAD83)
L023	STO E		STO E
L024	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
L025	PSE		PSE
L026	INPUT A		INPUT A
L027	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
L028	PSE		PSE
L029	INPUT E		INPUT E
L030	RCL E		
L031	√x		
L032	STO O		STO O
L033	CHK—NTR LAT 0		(Key in using EQN RCL C, RCL H, etc.)
L034	PSE		PSE
L035	INPUT P		INPUT P
L036	CHK—NTR LONG 0		(Key in using EQN RCL C, RCL H, etc.)
L037	PSE		PSE
L038	INPUT Q		INPUT Q

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Line	Instruction
L039	STD PARALLEL 1
L040	PSE
L041	INPUT C
L042	STD PARALLEL 2
L043	PSE
L044	INPUT D
L045	CHK—NTR E 0
L046	PSE
L047	INPUT G
L048	CHK—NTR N 0
L049	PSE
L050	INPUT H
L051	RCL P
L052	HMS→
L053	STO P
L054	RCL Q
L055	HMS→
L056	STO Q
L057	RCL C
L058	HMS→
L059	STO C
L060	RCL D
L061	HMS→
L062	STO D
L063	ENTER EASTING
L064	PSE
L065	INPUT X
L066	ENTER NORTHING
L067	PSE
L068	INPUT Y
L069	RCL G
L070	STO— X
L071	RCL H
L072	STO— Y
****	Calculate m_1
L073	RCL C
L074	STO Z
L075	XEQ L224
L076	RCL Z
L077	STO R
****	Calculate m_2
L078	RCL D
L079	STO Z
L080	XEQ L224
L081	RCL Z

Line	Instruction
L082	STO S
****	Calculate t_1
L083	RCL C
L084	STO Z
L085	XEQ L237
L086	RCL Z
L087	STO U
****	Calculate t_2
L088	RCL D
L089	STO Z
L090	XEQ L237
L091	RCL Z
L092	STO V
****	Calculate t_0
L093	RCL P
L094	STO Z
L095	XEQ L237
L096	RCL Z
L097	STO T
****	Calculate n
L098	RCL R
L099	LN
L100	RCL S
L101	LN
L102	—
L103	RCL U
L104	LN
L105	RCL V
L106	LN
L107	—
L108	÷
L109	STO N
****	Calculate F
L110	RCL R
L111	RCL÷ N
L112	RCL U
L113	RCL N
L114	y^x
L115	÷
L116	STO J
****	Calculate r_0
L117	RCL A
L118	RCL× J
L119	RCL T
L120	RCL N

Line	Instruction
L121	y^x
L122	×
L123	STO M
****	Calculate θ
L124	RCL X
L125	RCL M
L126	RCL— Y
L127	÷
L128	ATAN
L129	STO W
****	Calculate r
L130	RCL X
L131	x^2
L132	RCL M
L133	RCL— Y
L134	x^2
L135	+
L136	√x
L137	STO B
****	Calculate λ
L138	RCL W
L139	RCL÷ N
L140	RCL+ Q
L141	→HMS
L142	STO L
****	Calculate ϕ
L143	90
L144	RCL B
L145	RCL÷ A
L146	RCL÷ J
L147	RCL N
L148	1/x
L149	y^x
L150	STO I
L151	ATAN
L152	2
L153	×
L154	—
L155	STO F
L156	FN= Z
L157	SOLVE F
L158	GTO L164
L159	CANNOT SOLVE
L160	PSE
L161	FOR LATITUDE

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Line	Instruction
L162	PSE
L163	GTO L219
****	Calculate scale factor
L164	1
L165	RCL F
L166	SIN
L167	x^2
L168	RCL× E
L169	—
L170	$\sqrt{x}$
L171	RCL A
L172	x < > y
L173	÷
L174	RCL F
L175	COS
L176	×
L177	RCL÷ N
L178	RCL÷ B
L179	1/x
L180	STO K
****	Calculate γ
L181	RCL W
L182	→HMS
L183	STO W
****	Show results
L184	SF 10
L185	RESULTS
L186	PSE
L187	LATITUDE
L188	PSE
L189	RCL F
L190	→HMS
L191	STO F
L192	VIEW F
L193	LONGITUDE
L194	PSE
L195	VIEW L
L196	GRID CONV
L197	PSE
L198	VIEW W
L199	PT SCALE FACT
L200	PSE
L201	VIEW K
****	Check for next pt.
L202	0
L203	STO Z

Line	Instruction
L204	NEXT PT [0—1]
L205	PSE
L206	INPUT Z
L207	RCL Z
L208	x = 0 ?
L209	GTO L219
L210	NEW ZONE [0—1]
L211	PSE
L212	0
L213	STO Z
L214	INPUT Z
L215	RCL Z
L216	x = 0 ?
L217	GTO L063
L218	GTO L024
****	End of program
L219	PROGRAM END
L220	PSE
L221	FS? 1
L222	CF 10
L223	RTN
****	Subroutines
****	Compute m
L224	RCL Z
L225	COS
L226	RCL Z
L227	SIN
L228	x^2
L229	RCL× E
L230	1
L231	x < > y
L232	—
L233	$\sqrt{x}$
L234	÷
L235	STO Z
L236	RTN
****	Compute t
L237	1
L238	RCL Z
L239	SIN
L240	RCL× O
L241	—
L242	RCL Z
L243	SIN
L244	RCL× O
L245	1

Line	Instruction
L246	+
L247	÷
L248	RCL O
L249	2
L250	÷
L251	y^x
L252	45
L253	RCL Z
L254	2
L255	÷
L256	—
L257	TAN
L258	x < > y
L259	÷
L260	STO Z
L261	RTN
****	*********
****	Calculate ϕ
Z001	LBL Z
Z002	1
Z003	RCL F
Z004	STO Z
Z005	SIN
Z006	RCL× O
Z007	—
Z008	RCL F
Z009	SIN
Z010	RCL× O
Z011	1
Z012	+
Z013	÷
Z014	RCL O
Z015	2
Z016	÷
Z017	y^x
Z018	RCL× I
Z019	ATAN
Z020	2
Z021	×
Z022	90
Z023	x < > y
Z024	—
Z025	STO F
Z026	RCL— Z
Z027	RTN

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Notes

(1) The program should be run in RPN mode, as results in ALG mode are unknown.

(2) Latitudes and longitudes should be entered in HP notation, i.e., DDD.MMSS. The grid convergence is displayed in HP notation.

(3) The program may be used for any Lambert Conformal Conic projection, if the appropriate parameters are known. Similarly, any ellipsoid may be used, if its a and e^2 parameters are known. Parameters for a wide range of ellipsoids and all SPCS Lambert zones are included at the end of this document.

(4) Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in North America. It is critical to enter the correct sign in calculator when entering values.

(5) Lines with **** are comments only, and should not be entered into the calculator. They are there to make program entry a little easier.

(6) This program is long and often appears to be a stream of meaningless commands. This means that it may be more prone to errors when being entered. It is suggested that the program be entered using the given constants, tested (and the checksum checked), and when it is satisfactory, the values for the zone that are set to zero at the start of the program can be changed to those most suitable for the bulk of the expected work. See the **Localization** section at the end of the document.

(7) When working in SPCS 1927, there are some small differences in latitudes between this program and the NGS conversion program. This may be caused by a different method of computing the distance from the pole in days gone by (see the discussion in the **Theory** section). The differences are small (less than 0.25 inch in all places tested thus far) and appear to be larger the further the point is from the pole. The conversions in SPCS 1983 agree to 0.001 m, which is the finest value the NGS program provides.

(8) The scale factor is exactly 1.0000 when the point is on the standard parallels. It is less than 1.0000 between the standard parallels, and greater than 1.0000 outside the standard parallels.

Theory

Unlike the Transverse Mercator projection, where the forward and reverse co-ordinate conversions are pretty straight-forward, if long-winded, the solution to the conversion of grid co-ordinates to geographical co-ordinates on the Lambert Conformal Conic is a rather more complex affair, requiring an iterated solution of the latitude of the point.

Given a, e^2, ϕ_1, ϕ_2, ϕ_0, λ_0, E_0, N_0, and the x and y co-ordinates of the point to be converted (E and N or X and Y, with false origin removed), we can calculate n, F and r_0 from the USGS formulae given below. You substitute ϕ_0, ϕ_1, or ϕ_2 into the formulae for m and t to get the appropriate values (denoted by subscripts) required. The value of e in the formula for t is $\sqrt{e^2}$.

$$r_0 = a F t_0^n$$

$$n = \frac{\ln m_1 - \ln m_2}{\ln t_1 - \ln t_2} \qquad \text{(constant of the projection or cone)}$$

$$F = \frac{m_1}{n\, t_1^{\,n}}$$

$$m = \frac{\cos\phi}{\sqrt{1 - e^2 \sin^2\phi}}$$

$$t = \frac{\tan\left(\frac{\pi}{4} - \frac{\phi}{2}\right)}{\left(\frac{1 - e\ \sin\phi}{1 + e\ \sin\phi}\right)^{\frac{e}{2}}}$$

The following values may be computed using the co-ordinates (less the false origin values), as follows:

$$x = E - E_0$$

$$y = N - N_0$$

$$r = \sqrt{x^2 + (r_0 - y)^2}$$

$$\theta = \arctan\left(\frac{x}{r_0 - y}\right)$$

$$\lambda = \frac{\theta}{n} + \lambda_0$$

Then we have to undertake an iterative solution to the following equation, solving for ϕ with a rough estimate for ϕ, substituting that value back into the equation and solving for a better value of ϕ. When there is no significant change in ϕ, we can stop the process.

A good starting estimate for ϕ is $\phi = \frac{\pi}{2} - 2 \arctan t$.

The formula that is being iterated is:

$$\phi = \frac{\pi}{2} - 2 \arctan\left(t \left(\frac{1 - e \sin\phi}{1 + e \sin\phi}\right)^{e/2}\right)$$

where

$$t = \left(\frac{r}{a\,F}\right)^{1/n}$$

Using the following formulae, the other various required quantities can be calculated

$\gamma = \theta$ (grid convergence at the point)

$k = \dfrac{n\ r}{\nu \cos\phi}$ (scale factor at the latitude ϕ)

where ν = the radius of the ellipsoid at the parallel of latitude ϕ

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Note that n, F and r_0 are constants for a particular map or SPCS zone and only need to be computed once.

The iterative solution lends itself to the SOLVE capability in the calculator, and allows this part to be programmed economically and fairly simply, albeit at the cost of an additional label. When it was attempted to use SOLVE on an equation, the calculator ran out of memory, as using equations in SOLVE is memory hungry.

To avoid iterations, you can use the following series expansion (note that Snyder gives some clues for faster implementation of this expansion on page 19 of his book). I have not tried this out, so cannot comment on its effectiveness, but it may be better suited to being run in the HP-33S calculator (no loops required!).

$$\begin{aligned}\phi = \chi &+ \left(\frac{e^2}{2} + \frac{5e^4}{24} + \frac{e^6}{12} + \frac{13e^8}{360} + \dots\right)\sin 2\chi \\ &+ \left(\frac{7e^4}{48} + \frac{29e^6}{240} + \frac{811e^8}{11520} + \dots\right)\sin 4\chi \\ &+ \left(\frac{7e^6}{120} + \frac{81e^8}{1120} + \dots\right)\sin 6\chi + \left(\frac{4279e^8}{161280} + \dots\right)\sin 8\chi\end{aligned}$$

where

$$\chi = \frac{\pi}{2} - 2\arctan t$$

If you are using the tables for SPCS 27 and these were derived from geocentric latitudes, you can use the same formulae, except that t for calculations leading to n, F and r_0 (formulae given above) is calculated using:

$$t = \tan\left(\frac{\pi}{2} - \frac{\phi_g}{2}\right)$$

where

$$\phi_g = \frac{\pi}{2} - 2\arctan t$$

and the t in this equation is derived from $t = \left(\frac{r}{a\,F}\right)^{1/n}$

The 1927 solutions, based on slightly different calculation of ϕ and hence t, lead to slightly different results in the calculation of the final latitude from the co-ordinates. The differences are apparent in the example calculations given. You may want to check the differences across the region being converted and apply an average correction to the values computed by the calculator, or do the 1927 conversions on-line.

(Note that the SPCS 27 has co-ordinates in US Survey feet, and uses the Clarke 1866 ellipsoid. The SPCS 83 has co-ordinates in meters, and uses the GRS80 spheroid, which effectively is the same as WGS-84. Some states have either the US Survey foot or the International foot as alternative distance units; check which one is in use in the state you are working in at any particular time. Note that there is a datum shift between the two systems (1927 and 1983) as well, and that you cannot really do a direct linear shift between them.

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

As you can see from the tables below, most of the SPCS Lambert zones adopt ϕ_0 as having a Y or N value of zero. It is chosen so as to be well south of the limits of the zone. For most of the Lambert zones, the central meridian gets a value of 2,000,000 feet (SPCS 27), and a range of meter values for SPCS 83. See the tables below for exact data.

With the two standard parallels for the zone, it doesn't matter which is used for ϕ_1 and which for ϕ_2. Provided that the values for t_1, t_2, m_1 and m_2 are calculated and applied consistently, n, F and r_0 turn out the same either way. For convenience in the northern hemisphere, the southern parallel is used as ϕ_1, but this is not necessary for proper operation of the program.

Sample Computations

Example 1

Using the SPCS 1983 (a = 6,378,137 m, e^2 = 0.006 694 3800), the following results are obtained.

Ohio North Zone, 3401: True Origin: ϕ_0 = 39° 40', λ_0 = –82° 30';
Standard Parallels: ϕ_1 = 40° 26', ϕ_2 = 41° 42';
False Origin: E_0 = 600,000.000 m, N_0 = 0.000 m.

Easting (E) = 542,668.995 m Northing (N) = 47,416.966 m

Latitude = 40° 05' 30" Longitude = –83° 10' 20"

Grid Convergence (γ) = –0° 26' 29.82" Point Scale Factor (k) = 1.000 082 97

Example 2

Using the SPCS 1927 (a = 20,925,832.2 ft, e^2 = 0.006 768 66), the following results are obtained.

Ohio North Zone, 3401: True Origin: ϕ_0 = 39° 40', λ_0 = –82° 30';
Standard Parallels: ϕ_1 = 40° 26', ϕ_2 = 41° 42';
False Origin: E_0 = 2,000,000.000 ft, N_0 = 0.000 ft.

Easting (E) = 1,811,901.577 ft Northing (N) = 155,564.399 ft

Latitude = 40° 05' 30".000 0 Longitude = –83° 10' 20".000 0

Grid Convergence (γ) = –0° 26' 29.82" Point Scale Factor (k) = 1.000 082 97

Note: the NGS conversion program gave the same results, except for the latitude, which it gave as 40° 05' 30".000 06, a difference of 0.006 ft (about 0.07 inches or 1.8 mm), and –83° 10' 20".000 01 for the longitude, a difference of 0.000 75 ft (about 0.009 inches or 0.3 mm). Testing other points in this zone indicate a consistent difference of about this amount. This may be because of the different method of computing the distances from the pole (r and r_0) in earlier computations of the zones. Note that the NGS conversion from State Plane to geographical co-ordinates does not provide the grid convergence and scale factor. To get this, you will need to use a different converter.

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Example 3

Using the SPCS 1927 ($a = 20{,}925{,}832.2$ ft, $e^2 = 0.006\ 768\ 66$), the following results are obtained.

California III Zone (0403), SPCS 1927, $\phi_0 = 36° 30'$, $\lambda_0 = -120° 30'$;
$\phi_1 = 37° 04'$, $\phi_2 = 38° 26'$;
$E_0 = 2{,}000{,}000.000$ ft, $N_0 = 0.000$ ft.

Easting (E) = 2,216,169.136 ft   Northing (N) = 338,664.251 ft

Latitude = 37° 25' 40".000 0   Longitude = –119° 45' 20".000 0

Grid Convergence (γ) = 0° 27' 20.8"   Point Scale Factor (k) = 0.999 945 01

Note: the NGS conversion program gave the same results, except for the latitude, which it gave as 37° 25' 40".000 12, a difference of 0.012 ft (about 0.14 inches or 3.6 mm). Testing other points in this zone indicate a consistent difference of about this amount. This may be because of the different method of computing the distances from the pole (r and r_0) in earlier computations of the zones. Note that the NGS conversion from State Plane to geographical co-ordinates does not provide the grid convergence and scale factor. To get this, you will need to use a different converter.

Example 4

Using the SPCS 1983 ($a = 6{,}378{,}137$ m, $e^2 = 0.006\ 694\ 3800$), the following results are obtained.

California III Zone (0403), SPCS 1983, $\phi_0 = 36° 30'$, $\lambda_0 = -120° 30'$;
$\phi_1 = 37° 04'$, $\phi_2 = 38° 26'$;
$E_0 = 2{,}000{,}000.000$ m, $N_0 = 500{,}000.000$ m.

Easting (E) = 2,065,886.861 m   Northing (N) = 603,227.485 m

Latitude = 37° 25' 40"   Longitude = –119° 45' 20"

Grid Convergence (γ) = 0° 27' 20.8"   Point Scale Factor (k) = 0.999 945 01

Running the Program

Press XEQ L, then press ENTER to start the program. The calculator briefly displays LCC 2 LAT-LONG, then briefly shows CHECK—ENTER A. The program then stops and displays the prompt for entering the semi-major axis value, while displaying the current default value:

A?
6,378,137.0000   (This is for GRS80/WGS84/NAD83)

If you are happy with this value for the semi-major axis of the ellipsoid, press R/S to continue. Otherwise. Key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHECK—ENTER E. The program then stops and displays the prompt for entering the eccentricity of the ellipsoid, e:

E?
0.00669438   (This is for GRS80/WGS84/NAD83)

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

If this value for the eccentricity is correct, press R/S to continue. Otherwise, key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHK—NTR LAT 0. The program then stops and displays the prompt for entering the origin latitude for the co-ordinate, ϕ_0:

P?
0.0000

Key in the correct latitude in HP notation (DDD.MMSS), and press R/S to continue. In this case, key in 39.40 for Ohio North.

The calculator briefly displays CHK—NTR LONG 0. The program then stops and displays the prompt for entering the longitude of the central meridian of the projection, λ_0. Note that in the western hemisphere, this will be a negative value, and should be in HP notation (DDD.MMSS).

Q?
0.0000

Key in the correct longitude, in HP notation and remembering the sign, then press R/S to continue. In this case, key in –82.30 for Ohio North

The calculator briefly displays STD PARALLEL 1. The program then stops and displays the prompt for entering the latitude of one of the standard parallels for the projection, ϕ_1. The value should be entered in HP notation.

C?
0.000000

Key in the correct value and press R/S to continue. In this case, key in 40.26 and press R/S to continue.

The calculator briefly displays STD PARALLEL 2. The program then stops and displays the prompt for entering the latitude of the other standard parallel for the projection, ϕ_2. The value should be entered in HP notation.

D?
0.000000

Key in the correct value and press R/S to continue. In this case, key in 41.42 and press R/S to continue.

The calculator briefly displays CHK—NTR E 0. The program then stops and prompts for the false easting value, or the easting offset. This is the value of the easting at the central meridian (λ_0), denoted E_0.

G?
0.0000

Key in the correct value, and press R/S to continue. In this case, key in 600000.0 and press R/S.

The calculator briefly displays CHK—NTR N 0. The program then stops and prompts for entry of the false northing value, or the northing offset. This is the value of the northing co-ordinate at ϕ_0, λ_0.

H?
0.0000

If this is the correct value (for some zones, it is zero), press R/S to continue. If a different value is desired, key in the value and press R/S. In this case, just press R/S. This is the N_0 value for Ohio North.

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

The calculator briefly displays ENTER EASTING. The program stops and displays the prompt for entering the easting of the point to be converted. This should be entered in HP notation.

X?
0.0000

Key in the easting of the point and press R/S to continue. In this case, key in 542668.995 and press R/S.

The calculator briefly displays ENTER NORTHING. The program then stops and displays the prompt for entering the northing of the point to be converted. This should be entered in HP notation.

Y?
0.0000

Key in the longitude of the point in HP notation and press R/S to continue. In this case, key in 47416.966 and press R/S/

The program displays RUNNING for a short while, then SOLVING for a while, then displays RESULTS briefly, followed by LATITUDE briefly. The program then stops and displays the latitude value of the point. In this case, the calculator displays:

F=
40.05300000

This is the latitude of the point, in HP notation (DDD.MMSSsss, i.e., 40° 05' 30".00 N). Press R/S to continue. The calculator briefly displays LONGITUDE, then stops and displays the northing value of the point. In this case, the calculator displays:

L=
—83.10200000

This is the longitude of the point, in HP notation (DDD.MMSSsss, i.e., 83° 10' 20".00 W). Press R/S to continue. The calculator briefly displays GRID CONV, then stops and displays the grid convergence value in HP notation. In this case, the calculator displays:

K=
—0.26298198

This is the grid convergence in HP notation, and is —0° 26' 29".82 in more conventional notation. Press R/S to continue. The calculator briefly displays PT SCALE FACT, then stops and displays the point scale factor of the point on the Lambert Conformal Conic projection. In this case, the calculator displays:

W=
1.00008297

This is the point scale factor. Press R/S to continue.

You now have the choice of running one or more additional points. The calculator briefly displays NEXT PT [0—1], then stops and displays the prompt for answering questions:

Z?
0.0000

If you want to quit the program, just press R/S, the calculator briefly displays PROGRAM END and then comes to an end, returning to the point whence it was called, or to normal operations. If you want to enter more points, key in 1 and press R/S. In this case, the calculator then prompts to see if you want to use the same parameters. The calculator briefly displays NEW ZONE [0—1], then stops at the question prompt:

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Z?
0.0000

If you want to go to a new zone, key in 1 and press R/S, and the calculator will take you to the point where you can change any of the values (Point A above), starting with the ellipsoid parameters. If you want to work in the same zone already entered, just press R/S, and the program will take you to "Point B" and prompt for the latitude of the point to be converted, and continue from there. You can go around the program as many times as necessary.

Reference

SNYDER, J.P., 1987. *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper 1395. Washington: US Government Printing Office.

Storage Registers Used

A	Semi-major axis of the ellipsoid being used, a
B	r, distance from the pole to the point
C	ϕ_1, one of the two standard parallels of the projection
D	ϕ_2, one of the two standard parallels of the projection
E	Eccentricity of the ellipsoid, e^2
F	ϕ, latitude of the point being converted
G	E_0, the false easting or easting offset
H	N_0, the false northing, or northing offset, at ϕ_0, λ_0
I	Temporary variable t in the latitude solution.
J	F, an internal computed value
K	point scale factor, k
L	λ, longitude of the point being converted
M	r_0, distance from the pole to ϕ_0
N	n, the constant of the projection or cone
O	e, the square root of the eccentricity of the ellipsoid.
P	ϕ_0, the latitude of the co-ordinate origin on the projection
Q	λ_0, the central meridian of the projection
R	m_1, an internal computed value
S	m_2, an internal computed value
T	t_0, an internal computed value
U	t_1, an internal computed value
V	t_2, an internal computed value
W	θ, the angle between the line from the pole to the point, and the central meridian, also the grid convergence
X	Easting co-ordinate of converted point
Y	Northing co-ordinate of converted point
Z	Response variable for checking if another point, as well as passing values to subroutines.

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Statistical Registers: not used

Labels Used

Label **L**	Length = 1065	Checksum = 7096
Label **Z**	Length = 87	Checksum = 1E33

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Parameters for the Computations

Ellipsoids

There are a range of ellipsoids in common or former use. The table below has the a and e^2 values for a number of common (and less common) ellipsoids.

Ellipsoid	a Semi-major Axis	e^2 Eccentricity
GRS80–WGS94–NAD83	6378137 m	0.006 694 38
Clarke 1866 (NAD27)	6378206.4 m	0.006 768 66
Clarke 1866 (NAD27)	20925832.2 ft	0.006 768 66
ANS (Australian)	6378160 m	0.006 694 541 855
Airy 1830	6377563.4 m	0.006 670 54
Bessel 1841	6377397.16 m	0.006 674 372
Clarke 1880	6378249.15 m	0.006 803 511
Everest 1830	6377276.35 m	0.006 637 847
Fischer 1960 (Mercury)	6378166 m	0.006 693 422
Fischer 1968	6378150 m	0.006 693 422
Hough 1956	6378270 m	0.006 722 67
International	6378388 m	0.006 722 67
Krassovsky 1940	6378245 m	0.006 693 422
South American 1960	6378160 m	0.006 694 542
GRS 1967	6378160 m	0.006 694 605
GRS 1975	6378140 m	0.006 694 385
WGS 60	6378165 m	0.006 693 422
WGS 66	6378145 m	0.006 694 542
WGS 72	6378135 m	0.006 694 317 778
WGS 84	6378137 m	0.006 694 38

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

State Plane Co-ordinate System (SPCS) 1983

Several US states use the Lambert Conformal Conic projection for SPCS 1983. The various parameters for each zone in the 1983 system are given in the table below. Use these parameters with the program, together with the GRS80/WGS84/NAD83 ellipsoid parameters, in meters.

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Alaska						
Zone 10	51° 50'	53° 50'	176° 00'	51° 00'	1,000,000.00	0.00
Arkansas						
North	34° 56'	36° 14'	92° 00'	34° 20'	400,000.00	0.00
South	33° 18'	34° 46'	92° 00'	32° 40'	400,000.00	400,000.00
California						
I	40° 00'	41° 40'	122° 00'	39° 20'	2,000,000.00	500,000.00
II	38° 20'	39° 50'	122° 00'	37° 40'	2,000,000.00	500,000.00
III	37° 04'	38° 26'	120° 30'	36° 30'	2,000,000.00	500,000.00
IV	36° 00'	37° 15'	119° 00'	35° 20'	2,000,000.00	500,000.00
V	34° 02'	35° 28'	118° 00'	33° 30'	2,000,000.00	500,000.00
VI	32° 47'	33° 53'	116° 15'	32° 10'	2,000,000.00	500,000.00
Colorado						
North	39° 43'	40° 47'	105° 30'	39° 20'	914,401.83	304,800.61
Central	38° 27'	39° 45'	105° 30'	37° 50'	914,401.83	304,800.61
South	37° 14'	38° 26'	105° 30'	36° 40'	914,401.83	304,800.61
Connecticut	41° 12'	41° 52'	72° 45'	40° 50'	304800.61	152400.30
Florida						
North	29° 35'	30° 45'	84° 30'	29° 00'	600000.00	0.00
Iowa						
North	42° 04'	43° 16'	93° 30'	41° 30'	1500000.00	1000000.00
South	40° 37'	41° 47'	93° 30'	40° 00'	500000.00	0.00
Kansas						
North	38° 43'	39° 47'	98° 00'	38° 20'	400000.00	0.00
South	37° 16'	38° 34'	98° 30'	36° 40'	400000.00	400000.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Kentucky						
North	37° 58'	38° 58'	84° 15'	37° 30'	500,000.00	0.00
South	36° 44'	37° 56'	85° 45'	36° 20'	500,000.00	500,000.00
Louisiana						
North	31° 10'	32° 40'	92° 30'	30° 30'	1,00,0000.00	0.00
South	29° 18'	30° 42'	91° 20'	28° 30'	1,000,000.00	0.00
Offshore	26° 10'	27° 50'	91° 20'	25° 30'	1,000,000.00	0.00
Maryland	38° 18'	39° 27'	77° 00'	37° 40'	400,000.00	0.00
Massachusetts						
Mainland	41° 43'	42° 41'	71° 30'	41° 00'	200,000.00	750,000.00
Island	41° 17'	41° 29'	70° 30'	41° 00'	500,000.00	0.00
Michigan						
North	45° 29'	47° 05'	87° 00'	44° 47'	8,000,000.00	0.00
Central	44° 11'	45° 42'	84° 22'	43° 19'	6,000,000.00	0.00
South	42° 06'	43° 40'	84° 22'	41° 30'	4,000,000.00	0.00
Minnesota						
North	47° 02'	48° 38'	93° 06'	46° 30'	800,000.00	100000.00
Central	45° 37'	47° 03'	94° 15'	45° 00'	800,000.00	100,000.00
South	43° 47'	45° 13'	94° 00'	43° 00'	800,000.00	100,000.00
Montana	45° 00'	49° 00'	109° 30'	44° 15'	600,000.00	0.00
Nebraska	40° 00'	43° 00'	100° 00'	39° 50'	500,000.00	0.00
New York						
Long Island	40° 40'	41° 02'	74° 00'	40° 10'	300,000.00	0.00
North Carolina	34° 20'	36° 10'	79° 00'	33° 45'	609,601.22	0.00
North Dakota						
North	47° 26'	48° 44'	100° 30'	47° 00'	600,000.00	0.00
South	46° 11'	47° 29'	100° 30'	45° 40'	600,000.00	0.00
Ohio						
North	40° 26'	41° 42'	82° 30'	39° 40'	600,000.00	0.00
South	38° 44'	40° 02'	82° 30'	38° 00'	600,000.00	0.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Oklahoma						
North	35° 34'	36° 46'	98° 00'	35° 00'	600,000.00	0.00
South	33° 56'	35° 14'	98° 00'	33° 20'	600,000.00	0.00
Oregon						
North	44° 20'	46° 00'	120° 30'	43° 40'	2,500,000.00	0.00
South	42° 20'	44° 00'	120° 30'	41° 40'	1,500,000.00	0.00
Pennsylvania						
North	40° 53'	41° 57'	77° 45'	40° 10'	600,000.00	0.00
South	39° 56'	40° 58'	77° 45'	39° 20'	600,000.00	0.00
Puerto Rico and Virgin Islands						
1	18° 02'	18° 26'	66° 26'	17° 50'	200,000.00	200,000.00
2 (St. Croix)	18° 02'	18° 26'	66° 26'	17° 50'		
Samoa	–14° 16'	–14° 16'	170° 00'			
South Carolina	32° 30'	34° 50'	81° 00'	31° 50'	609,600.00	0.00
South Dakota						
North	44° 25'	45° 41'	100° 00'	43° 50'	600,000.00	0.00
South	42° 50'	44° 24'	100° 20'	42° 20'	600,000.00	0.00
Tennessee	35° 15'	36° 25'	86° 00'	34° 20'	600,000.00	0.00
Texas						
North	34° 39'	36° 11'	101° 30'	34° 00'	200,000.00	1,000,000.00
North central	32° 08'	33° 58'	98° 30'	31° 40'	600,000.00	2,000,000.00
Central	30° 07'	31° 53'	100° 20'	29° 40'	700,000.00	3,000,000.00
South central	28° 23'	30° 17'	99° 00'	27° 50'	600,000.00	4,000,000.00
South	26° 10'	27° 50'	98° 30'	25° 40'	300,000.00	5,000,000.00
Utah						
North	40° 43'	41° 47'	111° 30'	40° 20'	500,000.00	1,000,000.00
Central	39° 01'	40° 39'	111° 30'	38° 20'	500,000.00	2,000,000.00
South	37° 13'	38° 21'	111° 30'	36° 40'	500,000.00	3,000,000.00
Virginia						
North	38° 02'	39° 12'	78° 30'	37° 40'	3,500,000.00	2,000,000.00
South	36° 46'	37° 58'	78° 3'0	36° 20'	3,500,000.00	1,000,000.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False East	False North
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (m)	N_0 (m)
Washington						
North	47° 30'	48° 44'	120° 50'	47° 00'	500,000.00	0.00
South	45° 50'	47° 20'	120° 30'	45° 20'	500,000.00	0.00
West Virginia						
North	39° 00'	40° 15'	79° 30'	38° 30'	600,000.00	0.00
South	37° 29'	38° 53'	81° 00'	37° 00'	600,000.00	0.00
Wisconsin						
North	45° 34'	46° 46'	90° 00'	45° 10'	600,000.00	0.00
Central	44° 15'	45° 30'	90° 00'	43° 50'	600,000.00	0.00
South	42° 44'	44° 04'	90° 00'	42° 00'	600,000.00	0.00

State Plane Co-ordinate System (SPCS) 1927

Several US states used the Lambert Conformal Conic projection for SPCS 1927. The various parameters for each zone in the 1927 system are given in the table below. Use these parameters with the program, together with the Clarke 1866 ellipsoid in feet.

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Alaska						
Zone 10	51° 50'	53° 50'	176° 00'	51° 00'	3000000.00	0.00
Arkansas						
North	34° 56'	36° 14'	92° 00'	34° 20'	2000000.00	0.00
South	33° 18'	34° 46'	92° 00'	32° 40'	2000000.00	0.00
California						
I	40° 00'	41° 40'	122° 00'	39° 20'	2000000.00	0.00
II	38° 20'	39° 50'	122° 00'	37° 40'	2000000.00	0.00
III	37° 04'	38° 26'	120° 30'	36° 30'	2000000.00	0.00
IV	36° 00'	37° 15'	119° 00'	35° 20'	2000000.00	0.00
V	34° 02'	35° 28'	118° 00'	33° 30'	2000000.00	0.00
VI	32° 47'	33° 53'	116° 15'	32° 10'	2000000.00	0.00
VII	33° 52'	34° 25"	118° 20'	34° 08'	4186692.58	4160926.74

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Colorado						
North	39° 43'	40° 47'	105° 30'	39° 20'	2000000.00	0.00
Central	38° 27'	39° 45'	105° 30'	37° 50'	2000000.00	0.00
South	37° 14'	38° 26'	105° 30'	36° 40'	2000000.00	0.00
Connecticut	41° 12'	41° 52'	72° 45'	40° 50'	600000.00	0.00
Florida						
North	29° 35'	30° 45'	84° 30'	29° 00'	2000000.00	0.00
Iowa						
North	42° 04'	43° 16'	93° 30'	41° 30'	2000000.00	0.00
South	40° 37'	41° 47'	93° 30'	40° 00'	2000000.00	0.00
Kansas						
North	38° 43'	39° 47'	98° 00'	38° 20'	2000000.00	0.00
South	37° 16'	38° 34'	98° 30'	36° 40'	2000000.00	0.00
Kentucky						
North	37° 58'	38° 58'	84° 15'	37° 30'	2000000.00	0.00
South	36° 44'	37° 56'	85° 45'	36° 20'	2000000.00	0.00
Louisiana						
North	31° 10'	32° 40'	92° 30'	30° 40'	2000000.00	0.00
South	29° 18'	30° 42'	91° 20'	28° 40'	2000000.00	0.00
Offshore	26° 10'	27° 50'	91° 20'	25° 40'	2000000.00	0.00
Maryland	38° 18'	39° 27'	77° 00'	37° 50'	800000.00	0.00
Massachusetts						
Mainland	41° 43'	42° 41'	71° 30'	41° 00'	600000.00	0.00
Island	41° 17'	41° 29'	70° 30'	41° 00'	200000.00	0.00
Michigan (current)						
North	45° 29'	47° 05'	87° 00'	44° 47'	2000000.00	0.00
Central	44° 11'	45° 42'	84° 20'	43° 19'	2000000.00	0.00
South	42° 06'	43° 40'	84° 20'	41° 30'	2000000.00	0.00
Minnesota						
North	47° 02'	48° 38'	93° 06'	46° 30'	2000000.00	0.00
Central	45° 37'	47° 03'	94° 15'	45° 00'	2000000.00	0.00
South	43° 47'	45° 13'	94° 00'	43° 00'	2000000.00	0.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
Montana						
North	47° 51'	48° 43'	109° 30'	47° 00'	2000000.00	0.00
Central	46° 27'	47° 53'	109° 30'	45° 50'	2000000.00	0.00
South	44° 52'	46° 24'	109° 30'	44° 00'	2000000.00	0.00
Nebraska						
North	41° 51'	42° 49'	100° 00'	41° 20'	2000000.00	0.00
South	40° 17'	41° 43'	99° 30'	39° 40'	2000000.00	0.00
New York						
Long Island	40° 40'	41° 02'	74° 00'	40° 30'	2000000.00	100000.00
North Carolina	34° 20'	36° 10'	79° 00'	33° 45'	2000000.00	0.00
North Dakota						
North	47° 26'	48° 44'	100° 30'	47° 00'	2000000.00	0.00
South	46° 11'	47° 29'	100° 30'	45° 40'	2000000.00	0.00
Ohio						
North	40° 26'	41° 42'	82° 30'	39° 40'	2000000.00	0.00
South	38° 44'	40° 02'	82° 30'	38° 00'	2000000.00	0.00
Oklahoma						
North	35° 34'	36° 46'	98° 00'	35° 00'	2000000.00	0.00
South	33° 56'	35° 14'	98° 00'	33° 20'	2000000.00	0.00
Oregon						
North	44° 20'	46° 00'	120° 30'	43° 40'	2000000.00	0.00
South	42° 20'	44° 00'	120° 30'	41° 40'	2000000.00	0.00
Pennsylvania						
North	40° 53'	41° 57'	77° 45'	40° 10'	2000000.00	0.00
South	39° 56'	40° 58'	77° 45'	39° 20'	2000000.00	0.00
Puerto Rico and Virgin Islands						
1	18° 02'	18° 26'	66° 26'	17° 50'	500000.00	0.00
2 (St. Croix)	18° 02'	18° 26'	66° 26'	17° 50'	500000.00	100000.00
Samoa	-14° 16'	-14° 16'	170° 00'		500000.00	0.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

	Standard Parallels		Origin Longitude	Origin Latitude	False Easting	False Northing
	ϕ_1 South	ϕ_2 North	λ_0 West	ϕ_0 North	E_0 (ft.)	N_0 (ft.)
South Carolina						
North	33° 46'	34° 58'	81° 00'	33° 00'	2000000.00	0.00
South	32° 20'	33° 40'	81° 00'	31° 50'	2000000.00	0.00
South Dakota						
North	44° 25'	45° 41'	100° 00'	43° 50'	2000000.00	0.00
South	42° 50'	44° 24'	100° 20'	42° 20'	2000000.00	0.00
Tennessee	35° 15'	36° 25'	86° 00'	34° 40'	2000000.00	100000.00
Texas						
North	34° 39'	36° 11'	101° 30'	34° 00'	2000000.00	0.00
North central	32° 08'	33° 58'	97° 30'	31° 40'	2000000.00	0.00
Central	30° 07'	31° 53'	100° 20'	29° 40'	2000000.00	0.00
South central	28° 23'	30° 17'	99° 00'	27° 50'	2000000.00	0.00
South	26° 10'	27° 50'	98° 30'	25° 40'	2000000.00	0.00
Utah						
North	40° 43'	41° 47'	111° 30'	40° 20'	2000000.00	0.00
Central	39° 01'	40° 39'	111° 30'	38° 20'	2000000.00	0.00
South	37° 13'	38° 21	111° 30'	36° 40'	2000000.00	0.00
Virginia						
North	38° 02'	39° 12'	78° 30'	37° 40'	2000000.00	0.00
South	36° 46'	37° 58'	78° 30'	36° 20'	2000000.00	0.00
Washington						
North	47° 30'	48° 44'	120° 50'	47° 00'	2000000.00	0.00
South	45° 50'	47° 20'	120° 30'	45° 20'	2000000.00	0.00
West Virginia						
North	39° 00'	40° 15'	79° 30'	38° 30'	2000000.00	0.00
South	37° 29'	38° 53'	81° 00'	37° 00'	2000000.00	0.00
Wisconsin						
North	45° 34'	46° 46'	90° 00'	45° 10'	2000000.00	0.00
Central	44° 15'	45° 30'	90° 00'	43° 50'	2000000.00	0.00
South	42° 44'	44° 04'	90° 00'	42° 00'	2000000.00	0.00

Lambert Conformal Conic Co-ordinates to Latitude/Longitude

Localization

If it is intended to do most conversions in the one SPCS zone, then the parameters for that zone can be coded into the program. When such a program is run, the program will prompt for the values (which allows the user to work in a different zone, as needed), but will display and store the regular values for the chosen zone. These can also be changed by changing the program, if a series of points on a different zone are to be converted.

The code required to ‘hardwire’ zone-specific values into the program is given below, based on a specific zone. If we were going to use California Zone III in SPCS 1983, its parameters are:

ϕ_0 = 36° 30'	λ_0 = –120° 30'	a = 6378137 m	e^2 = 0.006 694 38
ϕ_1 = 37° 04'	ϕ_2 = 38° 26'	E_0 = 2,000,000.000 m	N_0 = 500,000.000 m

The resulting code would be as follows, with the rest of the code left out. Note that the angular values are entered in HP notation (DDD.MMSS), as the program converts everything for internal use later.

Line	Instruction	Display	User Instructions
L001	LBL L		
………	……………		
L011	CL x		
L012	STO X		
L013	STO Y		
L014	36.3		ϕ_0 value of zone
L015	STO P		
L016	—120.30		λ_0 value of zone
L017	STO Q		
L018	37.04		ϕ_1 value of zone
L019	STO C		
L020	38.26		ϕ_2 value of zone
L021	STO D		
L022	2000000.0		E_0 value of zone
L023	STO G		
L024	500000.0		N_0 value of zone
L025	STO H		
L026	6378137		a value for ellipsoid (WGS84/NAD83)
L027	STO A		
L028	0.00669438		e^2 value for ellipsoid (WGS84/NAD83)
L029	STO E		
L030	CHECK–ENTER A		
L031	PSE		
L032	INPUT A		
	…………………		

This will change subsequent line numbers (they will be 6 greater than before), as well as the program length and checksum, but the program should otherwise be unaffected and should run correctly. Use the values for your preferred zone, and everything should be fine.

Corrections Line L079 changed to STO Z. Alaska standard parallel value corrected.

Roots of a Quadratic Equation

Programmer: Dr. Bill Hazelton

Date: March, 2008. Version: 1.0 Mnemonic: **Q**uadratic Roots

Line	Instruction	User Instructions
Q001	LBL Q	LBL Q
Q002	CLSTK	CLEAR 5
Q003	FS? 10	FLAGS 3 .0
Q004	GTO Q008	
Q005	SF 1	FLAGS 1 1
Q006	SF 10	FLAGS 1 .0
Q007	GTO Q009	
Q008	CF 1	FLAGS 2 1
Q009	QUADRATIC ROOT	(Key in using EQN RCL Q RCL U, etc.,)
Q010	PSE	PSE
Q011	ENTER A	(Key in using EQN RCL E RCL N, etc.,)
Q012	PSE	PSE
Q013	INPUT A	INPUT A
Q014	ENTER B	(Key in using EQN RCL E RCL N, etc.,)
Q015	PSE	PSE
Q016	INPUT B	INPUT B
Q017	ENTER C	(Key in using EQN RCL E RCL N, etc.)
Q018	PSE	PSE
Q019	INPUT C	INPUT C
Q020	RCL B	
Q021	x^2	x^2
Q022	RCL A	
Q023	RCL× C	
Q024	4	
Q025	×	
Q026	–	
Q027	STO D	STO D
Q028	$x < 0?$	x?0 3 (<)
Q029	GTO Q054	[to 2 complex roots]
Q030	x = 0?	x?0 6 (=)
Q031	GTO Q082	[to 1 real root]
Q032	RCL D	[2 real roots]
Q033	√x	
Q034	RCL– B	
Q035	RCL÷ A	
Q036	2	
Q037	÷	
Q038	STO P	STO P
Q039	1ST REAL ROOT	(Key in using EQN 1 RCL S RCL T SPACE, etc.)

Line	Instruction	User Instructions
Q040	PSE	PSE
Q041	VIEW P	VIEW P
Q042	RCL D	
Q043	√x	
Q044	RCL+ B	
Q045	RCL÷ A	
Q046	2	
Q047	÷	
Q048	+/–	
Q049	STO Q	STO Q
Q050	2ND REAL ROOT	(Key in using EQN 2 RCL N RCL D SPACE, etc.)
Q051	PSE	PSE
Q052	VIEW Q	VIEW Q
Q053	GTO Q091	[to end part]
Q054	RCL D	[2 complex roots]
Q055	ABS	ABS
Q056	√x	
Q057	**i**	
Q058	×	
Q059	RCL—B	
Q060	RCL÷ A	
Q061	2	
Q062	÷	
Q063	STO P	STO P
Q064	1ST CMPLX ROOT	(Key in using EQN 1 RCL S RCL T SPACE, etc.)
Q065	PSE	PSE
Q066	VIEW P	VIEW P
Q067	RCL D	
Q068	ABS	ABS
Q069	√x	
Q070	+/—	
Q071	**i**	
Q072	×	
Q073	RCL—B	
Q074	RCL÷ A	
Q075	2	
Q076	÷	
Q077	STO Q	STO Q
Q078	2ND CMPLX ROOT	(Key in using EQN 2 RCL N RCL D SPACE, etc.)
Q079	PSE	PSE
Q080	VIEW Q	VIEW Q
Q081	GTO Q091	[to end part]
Q082	RCL B	[1 real root]
Q083	+/—	
Q084	RCL÷ A	
Q085	2	

Roots of a Quadratic Equation

Line	Instruction	User Instructions
Q086	÷	
Q087	STO P	STO P
Q088	SINGLE ROOT	(Key in using EQN RCL S RCL I RCL N, etc.)
Q089	PSE	PSE
Q090	VIEW P	VIEW P
O091	FS? 1	FLAGS 3 1
Q092	CF 10	FLAGS 2 .0
Q093	STOP	R/S
Q094	RTN	RTN

Notes

(1) A simple program to calculate the roots of a quadratic equation in standard form: $ax^2 + bx + c = 0$.

(2) If there is a single real root for the equation, the value in storage location D is equal to zero, so after showing SINGLE ROOT, then the value in storage location P (the solution), the program ends.

(3) If there are two real roots, the program displays 1ST REAL ROOT, displays the first solution (in storage location P), then displays 2ND REAL ROOT, then shows the second solution (in storage location Q).

(4) If there are two complex roots, the program displays 1ST COMPLX ROOT, displays the first solution as a complex number (using the complex number display format selected), then displays 2ND COMPLX ROOT, and then the second solution as a complex number.

(5) This program sets Flag 10, so that equations can be displayed as prompts. The program completes by returning Flag 10 to its setting when the program began.

(6) This program may be called by another program. The roots are left on the stack after the program finishes, regardless of the approach used, as well as in storage locations P and Q. To simplify returning to the calling routine, omit the STOP at line Q093. It is there to allow the stand-alone use of the program to avoid automatically returning to some other program point.

Theory

The solution to a quadratic equation in the form $ax^2 + bx + c = 0$ is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where $b^2 - 4ac$ is known as the discriminant. The discriminant, commonly termed Δ, is stored in storage location D.

If Δ is positive, the equation has two real roots. If Δ = 0, the equation has a single, real root, i.e., it is a perfect square. If Δ is negative, the equation has two complex roots. The calculator solves for all of these situations.

Roots of a Quadratic Equation

Running the Program

Press XEQ Q, then press ENTER.

Program displays QUADRATIC ROOT briefly, then ENTER A briefly, then prompts:
A?
Key in the value of *a* from the equation. Press R/S.

Program displays ENTER B briefly, then prompts:
B?
Key in the value of *b* from the equation. Press R/S.

Program displays ENTER C briefly, then prompts:
C?
Key in the value of *c* from the equation. Press R/S.

A. If there are two real roots, the program briefly displays 1ST REAL ROOT, then displays:
P=
and the value of one of the roots of the equation.

Press R/S and the calculator briefly displays 2ND REAL ROOT, then displays:
Q=
and the value of the other root of the equation.

Press R/S and the calculator resets the various flags and the program ends.

B. If there are two complex roots, the calculator briefly displays 1ST CMPLX ROOT, then displays:
P=
and the value of one of the roots, as a complex number. The value is displayed in the current calculator mode for displaying complex numbers. This may be set, and is explained below.

Press R/S and the calculator briefly displays 2ND CMPLX ROOT, then displays:
Q=
and the value of the other root, as a complex number.

Press R/S and the calculator resets the various flags and the program ends.

C. If there is a single root to the equation, i.e., the equation is a perfect square, the calculator briefly displays SINGLE ROOT, then displays:
P=
and the value of the single root.

Press R/S and the calculator resets the various flags and the program ends.

Complex Number Display

Complex numbers in the HP-35s can be displayed in two ways in RPN mode. These are the rectangular mode and polar mode. Which display mode is set depends upon the display setting. To see complex numbers in rectangular form, press ↰ DISPLAY, then 9. The *x i y* mode shows numbers as: 25.3456 i –54.3210

The first part is the real part of the number, and the i separates this from the imaginary part, which is the second part. So the above number would be read as 25.3456 – i54.3210.

The other display mode, polar, is selected by pressing ↰ DISPLAY, then selecting option 10 (scroll up or down to find it). This is shown as $r\ \theta\ a$, and shows first the distance, then the angle of this direction as an angle in decimal degrees counter-clockwise from the $\Re$ (real) axis (horizontal axis to the right).

The same number as above would be shown as: 59.943060 θ –64.986744

where the 59.943060 is the distance, and the –60.986744 is the angle in decimal degrees.

Complex numbers may be entered in either mode, but are displayed in the selected mode. The mode can be set or changed at any time, to suit the user's needs.

Sample Computations

1. Solve $4x^2 + x + 8 = 0$

   There are two complex roots for this equation.

   Solutions: 1st Complxx Root P = -0.125 + i1.408678
   2nd Complex Root Q = –0.125 – i1.408678

2. Solve $4x^2 + 8x - 6 = 0$

   There are two real roots for this equation.

   Solutions: 1st Root P = 0.581139
   2nd Root Q = –2.581139

3. Solve $x^2 + 2x + 1 = 0$

   There is just one root for this equation.

   Solution: Single Root P = –1.00000

Storage Registers Used

A Value of a in equation

B Value of b in equation

C Value of c in equation

D Discriminant, Δ

P Solution of the first root

Q Solution of the second root

Statistical Registers: not used

Labels Used

Label **Q** Length = 390 Checksum = 4A3E

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Roots of Complex Numbers

Programmer: Jackson Reid (with additions by Bill Hazelton)

Date: May, 2008. Version: 1.0 Mnemonic: **R** for Complex **R**oots

Line	Instruction	User Instructions
R001	LBL R	↱ LBL R
R002	CLSTK	↱ CLEAR 5
R003	FS? 10	↰ FLAGS 3 .0
R004	GTO R008	
R005	SF 1	↰ FLAGS 1 1
R006	SF 10	↰ FLAGS 1 .0
R007	GTO R009	
R008	CF 1	↰ FLAGS 2 1
R009	COMPLEX ROOT	(Key in using EQN RCL C RCL O, etc.,)
R010	PSE	↱ PSE
R011	ENTER Z	(Key in using EQN RCL E RCL N, etc.,)
R012	PSE	↱ PSE
R013	INPUT Z	↰ INPUT Z
R014	ENTER N	(Key in using EQN RCL E RCL N, etc.,)
R015	PSE	↱ PSE
R016	RAD	MODE 2
R017	INPUT N	↰ INPUT N
R018	1	
R019	—	
R020	STO L	↱ STO L
R021	RCL Z	
R022	ABS	↱ ABS
R023	STO R	↱ STO R
R024	RCL Z	
R025	ARG	↰ ARG
R026	STO A	↱ STO A
R027	RCL R	
R028	RCL N	
R029	$\sqrt[x]{y}$	↰ $\sqrt[x]{y}$
R030	STO B	↱ STO B
R031	RCL A	
R032	RCL÷ N	
R033	COS	
R034	RCL A	
R035	RCL÷ N	
R036	SIN	
R037	**i**	[The **i** key, next to the cursor keys]
R038	×	
R039	+	

Roots of Complex Numbers

Line	Instruction	User Instructions
R040	RCL× B	
R041	STO C	STO C
R042	FIRST ROOT	(Key in using EQN RCL F RCL I, etc.,)
R043	PSE	PSE
R044	VIEW C	VIEW C
R045	RCL L	
R046	1	
R047	STO– L	STO– L
R048	π	π
R049	2	
R050	×	
R051	STO+ A	STO+ A
R052	RCL A	
R053	RCL÷ N	
R054	COS	
R055	RCL A	
R056	RCL÷ N	
R057	SIN	
R058	**i**	[The **i** key, next to the cursor keys]
R059	×	
R060	+	
R061	RCL× B	
R062	STO D	STO D
R063	NEXT ROOT	(Key in using EQN RCL N RCL E, etc.)
R064	PSE	PSE
R065	VIEW D	VIEW D
R066	RCL L	
R067	x > 0?	x?0 4
R068	GTO R045	
R069	PROGRAM END	(Key in using EQN RCL P RCL R, etc.)
R070	PSE	PSE
R071	FS? 1	FLAGS 3 1
R072	CF 10	FLAGS 2 .0
R073	DEG	MODE 1
R074	RTN	RTN

Notes

(1) A simple program to calculate the roots of a complex number using De Moivre's formula.

(2) The user provides the complex number using the standard calculator entry methods, i.e., $\Re$ **i** $\Im$, or r θ a. The 'number' of the roots to be calculated is then entered.

(3) The results (the various roots) are shown as complex numbers in the current complex display mode of the calculator. If the display is in polar complex display mode, the displayed angles (the value after the θ) will be in radians.

Roots of Complex Numbers

(4) The program works largely in radians mode, but returns to degrees mode at the end of the program. If the program is stopped before coming to the end, the calculator may remain in radians mode.

(5) The program sets Flag 10, so that equations can be displayed as prompts. The program completes by returning Flag 10 to its setting when the program began.

Theory

De Moivre's formula can be written as:

$$(\cos x + i\sin x)^n = \cos(nx) + i\sin(nx)$$

The formula can be used to find the *n*th roots of a complex number, as follows. If z is a complex number, written in polar form as:

$$z = r(\cos x + i\sin x)$$

then:

$$z^{\frac{1}{n}} = \left[r(\cos x + i\sin x)\right]^{\frac{1}{n}} = r^{\frac{1}{n}}\left[\cos\left(\frac{x+2k\pi}{n}\right) + i\sin\left(\frac{x+2k\pi}{n}\right)\right]$$

where k is an integer, and to get the n different roots of z one only needs to consider values of k from 0 to n–1.

Running the Program

Press XEQ R, then press ENTER.

Program displays COMPLEX ROOTS briefly, then ENTER Z briefly, then prompts:

Z?

Key in the complex number, or if using a real number, key it in as a complex number, then press R/S. For example, to find the roots of the real number +1, key in 1 **i** 0, then press R/S.

Program displays ENTER N briefly, then prompts:

N?

Key in the value of n, the integer number of the roots to be obtained (i.e., finding the *n*th root). Press R/S. For example, key in 8 and press R/S/

The calculator briefly displays RUNNING, then briefly displays FIRST ROOT, then stops and shows the first root calculated as a complex number. In this case, if the calculator is in rectangular complex number display mode, the display will show:

C=

1.0000 **i** 0.0000

The first root of +1 is, simply, 1. Press R/S to continue.

Roots of Complex Numbers

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
0.7071 **i** 0.7071

This is equivalent to $\frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}$. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
–5.1034E–12 **i** 1.0000

This is equivalent to i. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
–0.7071 **i** 0.7071

This is equivalent to $-\frac{\sqrt{2}}{2}+i\frac{\sqrt{2}}{2}$. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
–5.1034E–12 **i** 1.0000

This is equivalent to i. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
–1.000 **i** –1.0207E–11

This is equivalent to –1. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=
–0.7071 **i** –0.7071

Roots of Complex Numbers

This is equivalent to $-\frac{\sqrt{2}}{2}-i\frac{\sqrt{2}}{2}$. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=

1.5310E–11 **i** –1.0000

This is equivalent to –i. Press R/S to continue.

The calculator briefly displays RUNNING, then briefly displays NEXT ROOT, then stops and shows the next root calculated as a complex number. The display will show (if in rectangular mode):

C=

0.7071 **i** –0.7071

This is equivalent to $\frac{\sqrt{2}}{2}-i\frac{\sqrt{2}}{2}$. Press R/S to continue.

The calculator displays PROGRAM END and finishes, resetting the Flag 10 and the angular mode.

Complex Number Display

Complex numbers in the HP-35s can be displayed in two ways in RPN mode. These are the rectangular mode and polar mode. Which display mode is set depends upon the display setting. To see complex numbers in rectangular form, press ↰ DISPLAY, then 9. The *x i y* mode shows numbers as: 25.3456 i –54.3210

The first part is the real part of the number, and the *i* separates this from the imaginary part, which is the second part. So the above number would be read as 25.3456 – *i*54.3210.

The other display mode, polar, is selected by pressing ↰ DISPLAY, then selecting option 10 (scroll up or down to find it). This is shown as *r* θ *a*, and shows first the distance, then the angle of this direction as an angle in decimal degrees counter-clockwise from the $\Re$ (real) axis (horizontal axis to the right). If the calculator is in radians mode, the angle is displayed in radians (similarly in grads mode).

The same number as above would be shown as: 59.943060 θ –64.986744

where the 59.943060 is the distance, and the –60.986744 is the angle in decimal degrees.

Complex numbers may be entered in either mode, but are displayed in the selected mode. The mode can be set or changed at any time, to suit the user's needs.

Roots of Complex Numbers

Sample Computations

1. Solve for the 8th root of 1. See the solutions in the Running the Program Section.
2. Solve for the 4th roots of 1.

Solutions:	1st Root	1 (1 **i** 0)
	2nd Root	i (0 **i** 1)
	3rd Root	–1 (–1 **i** 0)
	4th Root	–i (0 **i** –1)

3. Solve for the 3rd roots of 3.5 + i 2.6

Solutions:	1st Root	1.5968 **i** 0.3453
	2nd Root	–1.0974 **i** 1.2102
	3rd Root	–0.4993 **i** –1.5555

4. Solve for the 3rd roots of 3.5 θ 55 (equivalent to 2.0075 **i** 2.8670)

Solutions:	1st Root	1.4412 **i** 0.4776	or	1.5183 θ 0.3200
	2nd Root	–1.1342 **i** 1.0094	or	1.5183 θ 2.4144
	3rd Root	–0.3070 **i** –1.4869	or	1.5183 θ –1.7744

In the first case, the calculator is set to rectangular complex display, and in the second case, it is set to polar complex display. Note that the angles (the value after the θ) are in radians.

Storage Registers Used

A Angular value in the complex number

B Value of $\sqrt[n]{r}$ (i.e., the *n*th root of *r*

C The first root of the complex number, *z*, in complex number format

D Subsequent roots of the complex number, *z*, in complex number format

L Loop counter, decremented each time through the loop

N The value *n*, that being the number of roots to be taken

R The absolute value (or length) of the complex number (or vector), *z*, i.e., $|z|$

Z The complex number, *z*, whose *n* roots are required

Statistical Registers: not used

Labels Used

Label **R** Length = 284 Checksum = 0467

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Complex Numbers, Vectors and Co-ordinates: Getting the Best from the HP-35s.

Introduction

In the third quarter of 2007, Hewlett Packard introduced the HP-35s pocket calculator. The model number (HP-35) commemorates 35 years since the introduction of their first pocket calculator, the HP-35A, thirty-five years before (1972). The HP-35A revolutionized computing and set an exceptionally high standard of quality and performance.

The new HP-35s may not have the same general impact as its famous forebear, but some of its features and approaches to work methodologies will cause some concern for users in the surveying community. This is because the method of dealing with co-ordinates and their connection with azimuths and distances is very different in the HP-35s. The Rectangular to Polar (and *vice versa*) functions no longer appear on the calculator, for example, and complex numbers are almost completely integrated with regular real-number calculator operations.

Figure 1 The HP-35s calculator.

Comparisons with Other HP Calculators

The HP-33S calculator, released in 2004, followed a long line of mid-range pocket calculators with broad functionality, starting with the HP-25, released in the late 1970s[1]. These calculators had all the computing power of the upper-range machines, but lacked the expansion, alphabetical, graphing and some high-end calculation capabilities of the more expensive calculators (e.g., HP-41, HP-48). For general number crunching, however, they were excellent. This line of calculators was a very popular choice for surveyors, especially for personal and field calculators.

With these calculators, the process of working with a traverse, for example, was consistent across the machines and the years. The azimuth and distance of each line in the traverse were entered in the appropriate format and arrangement, they were converted to latitude and departure components (using Polar to Rectangular conversion), and these were added in the statistical registers. The resulting sums of the latitudes and departures could be converted back to a closing azimuth and distance using the Rectangular to Polar conversion).

The HP-50g calculator is currently HP's top-end offering, with formidable computational power. It is a machine that essentially runs the HP-48GX's operating system and software, with extensions and developments, on a new hardware platform. While the operating system has changed dramatically from the first HP high-end pocket calculator, the HP-50g can trace its roots back to that machine, the venerable HP-65.

With the HP-48 onwards (the HP-48, HP-49 and HP-50 models), the operating system is completely object-oriented, which is noticeable in how the calculator works. The method of working with a traverse with these calculators is to enter the distance and azimuth as a vector. Subsequent vectors can be added to this vector. The vector can be viewed in rectangular mode, showing co-ordinates, or in polar mode, showing the azimuth and distance of the vector. The calculator includes tools for building, decomposing and manipulating vectors, which make handing these relatively simple.

The HP-35s is part-way between the approaches of the HP-33S and the HP-50g. It allows the use of vectors, but only in a very limited way. Co-ordinates and vectors in 2-D are better represented as complex numbers, and the calculator has limited tools to work with these in ways that are useful for surveyors. There is no simple way to work traverse measurements in the same manner as with the HP-33S, but there are insufficient tools to allow work to be done in the same manner as in the HP-50g.

In summary, in order to work with traverse sides and co-ordinates in the HP-35s, you need to know about complex numbers, vectors and how these all relate to each other, at least in the context of how the HP-35s works. This is not to say that the approach taken in the HP-35s is wrong, merely different from how things were done in the past.

This document is intended to inform surveyors and surveying students about things that are handy to know about complex numbers, vectors and co-ordinates, as they apply to how things are done with the HP-35s. Once the users can get the ideas into their way of thinking, it should be much easier to get the calculator to do what you need it to do.

---

[1] It could be argued that the line really begins with the HP-45, released around 1973, but the HP-45 lacked programmability. The HP-55 of the same era had some programming capacity, but was not as popular as the other calculators in the HP model line at that time.

Real and Other Numbers and the Argand Plane

All the numbers we tend to run across in everyday life are real numbers. Real numbers include the integers, fractions, decimal numbers, irrational numbers such as $\sqrt{2}$, and transcendental numbers, such as π. Real numbers can be represented on a number line. However, there are some algebraic equations that cannot be solved in terms of real numbers, e.g., $x^2 + 4x + 6 = 0$, which has the solutions $x = -2 + \sqrt{(-2)}$ and $-2 - \sqrt{(-2)}$. Square roots of negative numbers are not defined among the real numbers to allow these solutions.

A solution to this difficulty was the creation of imaginary numbers, all of which are able to be expressed as $a\ i$, where a is any real number and i is the square root of –1. Since imaginary numbers are simply the value i scaled by a real number, the imaginary numbers can be represented on an imaginary number line.

If we have to deal with a single number such as one of those above, e.g., $x = -2 + \sqrt{(-2)}$, this is often expressed as $x = -2 + i\sqrt{2}$, which is made up of two parts, one real (–2) and one imaginary $i\sqrt{2}$. This type of number is termed a *complex number*, as it is made up of more than one number, and cannot be reduced to anything simpler than this two-part number.

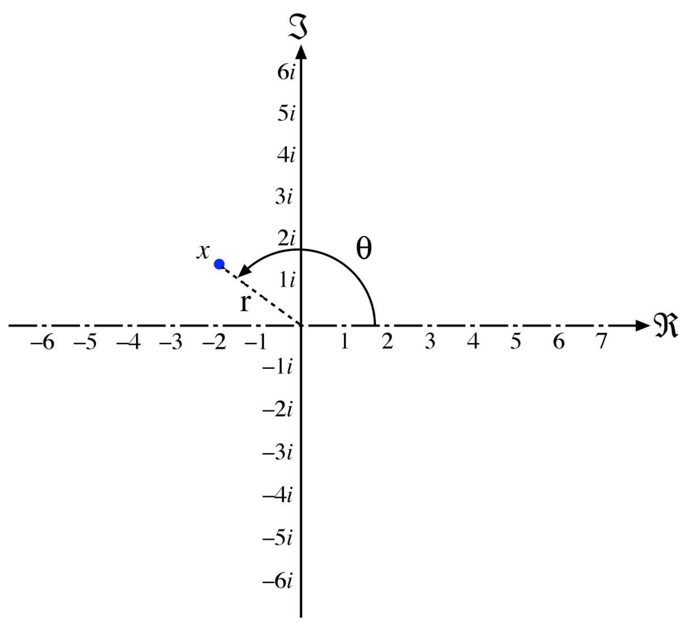


Figure 2 The Argand Plane, used for locating complex numbers.

If the real part of the number can be represented on the real number line, while the imaginary part can be represented on the imaginary number line, it is possible to represent the complex number on a 2-D plane having co-ordinates on each number line, with the two number lines at right angles. This plane is known as the *Argand Plane*, and the representation of the complex number $x = -2 + i\sqrt{2}$ is shown in Figure 2.

The –2 value is located on the real number axis (designated with the letter $\Re$ for 'real'), and the $i\sqrt{2}$ value is located on the imaginary number axis (designated with the letter $\Im$ for 'imaginary'). The point produced from these two co-ordinates is the location of the complex number x in the Argand Plane.

The complex number x can also be located using polar co-ordinates, rather than rectangular co-ordinates. In this case, the distance of the point x from the origin of the Argand Plane (r) and the rotation (θ) of the line between the origin and the point x, taken counter-clockwise from the positive real axis, define the same line. In the case of the complex number x, the value of r is 2.4495 and the angle is 144°.7356, so x can be expressed as $-2 + i\sqrt{2}$ or 2.4495 θ 144°.7356. Both representations are equivalent and the choice between them comes down to convenience at any given time. In many cases in mathematics, the angle is recorded in radians, rather than degrees, but this is a straightforward difference, and converting decimal degrees to radians and *vice versa* are built-in functions..

The Argand plane follows the conventions of mathematical co-ordinate representations, in that all rotations are measured counter-clockwise from the horizontal axis extending to the right. In the surveying and mapping disciplines, a different convention applies, so we have to think about a 'Surveying Argand Plane' to work with co-ordinates in the surveying sense. This is shown in Figure 3.

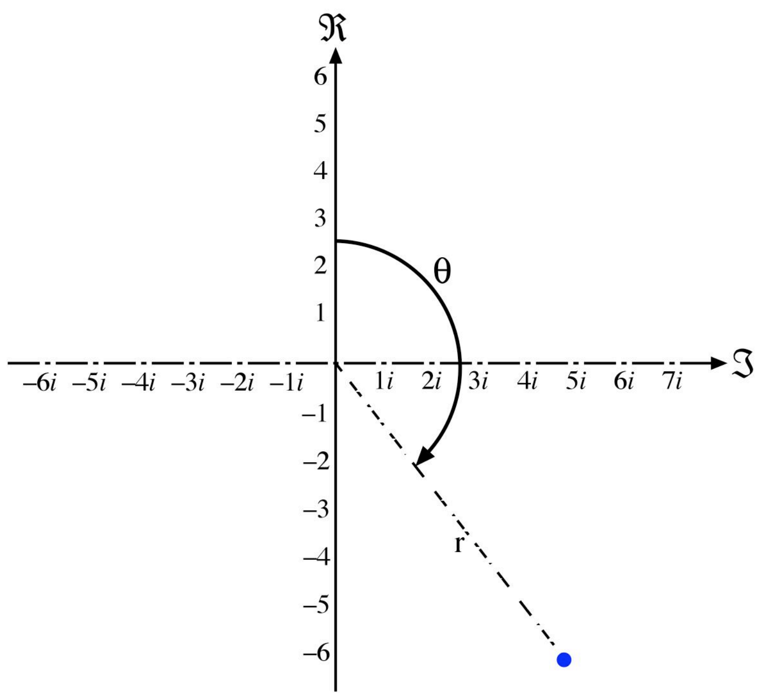


Figure 3 The 'Surveying Argand Plane,' showing a point with both rectangular and polar co-ordinates.

In this case, the real axis is up the page, and is equivalent to the usual North or Y axis, while the imaginary is to the right, equivalent to the usual East or X axis. A point with the co-ordinates East = 5, North = –6 has been shown. The equivalent polar co-ordinates are r = 7.8105 and θ = 140°.1944.

Of course, the same point could be designated as $z = -6 + 5i$, or z = 7.8102 θ 144°.1944, or z = 7.8102 θ 2.4469 radians.

So there are many ways to represent a complex number, and a complex number can be used to represent a point in a 2-D space.

In the same way that a complex number can be represented in both rectangular and polar forms, a vector can be represented in both forms. In polar form, it is the familiar azimuth and distance of a line. In rectangular form, the vector appears as orthogonal components, usually the latitude and departure of the line. In fact, in the HP-35s, it is more useful to represent 2-D vectors as complex numbers than as actual vectors.

Working with the HP-35s

The HP-35s allows complex numbers to be carried through many calculations, registers and stack locations in the same manner as conventional real numbers. The calculator allocates sufficient memory for the full 12-digit precision of both components of the complex number to be maintained. If the display is too small to show the full precision, the number can be scrolled to the left or right to allow the user to see the full precision of the complex number's components.

With this complex number capability, combined with the lack of built-in ways of working with vectors solely by components (which was the way it was done on the HP-33s back to the HP-45), it is sensible to work completely with the complex number representations, rather than trying to deal with the components manually. It also allows some of the capabilities of the HP-50g to be used for work on the HP-35s.

Displaying Complex Numbers

The HP-35s can display complex numbers in two forms when in RPN mode. These are the rectangular and polar forms. The DISPLAY menu controls which is used for display (note that the internal representation remains the same; only the display is changed). Pressing the gold (left) key, followed by the < cursor key (DISPLAY) will show the DISPLAY menu. Scrolling down will show options 9 and 10. These can be selected by scrolling to the appropriate value and pressing enter, or by pressing the number on the keyboard (use .0 for 10). Option 9 sets the calculator to display complex numbers in $x\ i\ y$ mode (rectangular), while option 10 sets the calculator to display complex numbers in r θ α mode (polar).

Regardless of the complex number was entered, the display mode will change how all complex numbers are displayed. Complex numbers may be entered in either mode, regardless of the current display more.

Note that when complex numbers are displayed in r θ α mode, there is no space between the characters, so the real component ends right next to the displayed θ, which is right next to the imaginary part. It is very easy to read the θ as the digit 8, e.g., 123.4567θ987.6543.

Entering Co-ordinates as a Complex Number

In many cases, co-ordinates are used in conjunction with vector calculations, and so they are most useful when already in the calculator's complex number form. To enter a co-ordinate pair as a complex number, key in the Northing or Y value first, press the **i** key (fourth from the left on the second row from the top), then key in the Easting or X value. If you press the Enter key, the complex number will be duplicated into the Y stack register, and will appear twice, once in each line of the display.

If a second co-ordinate pair is entered, it is possible to subtract one from the other, which will produce a complex number whose components are the differences in Northings and Eastings, if the display setting is for rectangular mode. Changing the display mode to viewing complex numbers in polar form, the complex number appears as a vector, with distance and orientation displayed.

When entering co-ordinates, remember the order: **North**, then **i**, then **East**.

To enter the co-ordinate as part of a program, move the Northing value to the Y register (line 1) and the Easting value to the X register (line 2). This can be done with two RCL statements. Then enter 0 **i** 1 on the next line of the program, then ×, then +. This multiplies the Easting value by **i**, forming a complex number with the value 0 **i** Easting. Adding the Northing as a real number makes the complex number Northing **i** Easting, which is now ready for further processing. This approach can be seen in several of the programs, and as it takes three program lines beyond recalling the co-ordinates, it isn't worth having a separate program for this operation.

Entering Vectors as a Complex Number

Vectors can be entered as a distance and azimuth, providing the polar form of a complex number. Because an angle is involved, it is necessary to enter the angular part in the same form as the current calculator setting. For most surveying work, this will be degrees, but it is also possible to use radians. To enter the vector, key in the distance, then press the blue (right) arrow key, followed by the θ key (on the lower face of the **i** key), then the azimuth in decimal degrees.

Once the Enter key is pressed, the complex number will be pushed up the stack, and will be displayed in the current mode.

If two vectors have been entered, the vector sum can be obtained by having the two vectors on the stack and pressing the + key. If the calculator is on polar display mode, this is the same as keying in the azimuth and distance of each line, converting it to rectangular components, adding the components (usually in the statistical registers), retrieving the sums and converting them back to polar representation; but in this case it can be done with a single keystroke.

One complication is the issue of entering an azimuth in degrees, minutes and seconds. The azimuth must be converted to decimal degrees, but this cannot be done within the complex number, nor within the complex number entry mode. (This can be done on the HP-50g, as vectors may be built directly from elements on the stack.)

The solution is a small program that allows the user to enter azimuths in degrees, minutes and seconds, followed by the distance, and convert them to a complex number. The program (Utility 3) will accomplish this. The azimuth is entered in HP notation (DDD.MMSSsss).

Extracting the Azimuth and Distance from a Complex Number

First make sure the calculator is in DEGREES mode. If the complex number is in the X register, press the blue right-arrow key, then ABS. This is the distance. Before doing anything else, press the blue right arrow key and then LAST*x*, to bring the complex number back to the X register. Press the gold left arrow key, then ARG. The value in line 2 (the X register) is the azimuth in decimal degrees. If this value is negative, enter 360, then press +. Press the blue right arrow key and →HMS to convert the azimuth to degrees, minutes and seconds (in HP notation).

The calculator now has the distance in the Y register (line 1) and the azimuth in HP notation in the X register (line 2).

Extracting Co-ordinates from a Complex Number

In the HP-35s, there is no built-in function to extract the real or imaginary parts from a complex number. This means that if the co-ordinates of a point, or the latitude and departure of a line, are required for separate processing, they have to be extracted from the complex number by several steps. A simple program to do this is available (Utility 4).

Other Capabilities

The HP-35s can do a number of other operations with complex numbers. as well as addition and subtraction, two complex numbers can be multiplied together, or one may be divided into the other. Complex numbers may be raised to powers, and these powers may also be complex numbers. Complex numbers can be used as the exponent of e, and natural logarithms of complex numbers may also be calculated. Trigonometric functions can also use complex numbers as arguments. As already covered, the absolute value (distance or length) and argument (orientation or angle) of a complex number may be calculated.

Multiplication, division and exponentiation as complex numbers are not very relevant to everyday vector usage, and so may be ignored by most surveying people, together with logarithms, anti-logarithms and trigonometric functions involving complex numbers.

Two Other Programs

Two other programs are supplied to cover two small areas of deficiency in the HP-35s calculator. These two deficiencies are the absence of functions that can directly add and subtract angles in degrees, minutes and seconds. These two functions are extremely handy when working with traverses and angle observations, and while present in the HP-50g, are not present in the HP-35s, nor in the HP-33s.

The HMS+ program is Utility 1, while HMS– is Utility 2. These programs are assigned to the C and D keys as they are extremely useful for many surveying calculations.

Note that all these programs can be called as sub-routines from inside other programs, to accomplish tasks required by those programs. While the programs will affect values on the stack, they are designed to avoid causing problems with values in the storage registers that are designated by letters, i.e., A to Z.

Dr. Bill Hazelton
17th October, 2007.

HP-35s Calculator Program TM 1

Convert Latitude and Longitude to Transverse Mercator Co-ordinates (UTM, SPCS, etc.)

Programmer: Dr. Bill Hazelton

Date: April, 2008. **Version**: 1.0 **Mnemonic**: **T** for **T**ransverse Mercator

Line	Instruction	Display	User Instructions
T001	LBL T		↱ LBL T
T002	CLSTK		↱ CLEAR 5
T003	FS? 10		↰ FLAGS 3 .0
T004	GTO T008		
T005	SF 1		↰ FLAGS 1 1
T006	SF 10		↰ FLAGS 1 .0
T007	GTO T009		
T008	CF 1		↰ FLAGS 2 1
T009	LAT–LONG 2 TM		(Key in using EQN RCL L, RCL A, etc.)
T010	PSE		↱ PSE
T011	CL x		↱ CLEAR 1
T012	STO F		↱ STO F
T013	STO L		↱ STO L
T014	6378137		a value for ellipsoid (WGS84)
T015	STO A		↱ STO A
T016	0.00669438		e^2 value for ellipsoid (WGS84)
T017	STO E		↱ STO E
T018	0.9996		k_0 value for zone (UTM 17)
T019	STO K		↱ STO K
T020	–81		λ_0 for zone (UTM 17)
T021	STO B		↱ STO B
T022	0		ϕ_0 for zone (UTM 17)
T023	STO C		↱ STO C
T024	500000		E_0 for zone (UTM 17)
T025	STO J		↱ STO J
T026	0		N_0 for zone (UTM 17)
T027	STO I		↱ STO I
T028	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
T029	PSE		↱ PSE
T030	INPUT A		↰ INPUT A
T031	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
T032	PSE		↱ PSE
T033	INPUT E		↰ INPUT E
T034	CHECK–ENTER K		(Key in using EQN RCL C, RCL H, etc.)
T035	PSE		↱ PSE
T036	INPUT K		↰ INPUT K
T037	CHK–NTR LONG 0		(Key in using EQN RCL C, RCL H, etc.)
T038	PSE		↱ PSE

Latitude/Longitude to Transverse Mercator Co-ordinates

Line	Instruction
T039	INPUT B
T040	CHK–NTR LAT 0
T041	PSE
T042	INPUT C
T043	CHK–NTR N 0
T044	PSE
T045	INPUT I
T046	CHK–NTR E 0
T047	PSE
T048	INPUT J
T049	RCL B
T050	HMS→
T051	STO B
T052	RCL C
T053	HMS→
T054	STO C
T055	ENTER PT LAT
T056	PSE
T057	INPUT F
T058	ENTER PT LONG
T059	PSE
T060	INPUT L
T061	RCL F
T062	HMS→
T063	STO F
T064	RCL L
T065	HMS→
T066	STO L
****	Compute ν
T067	1
T068	RCL F
T069	SIN
T070	x^2
T071	RCL× E
T072	—
T073	√x
T074	RCL A
T075	x < > y
T076	÷
T077	STO N
****	Compute ρ
T078	1
T079	RCL— E
T080	RCL× A
T081	RCL F

Line	Instruction
T082	SIN
T083	x^2
T084	RCL× E
T085	1
T086	x < > y
T087	—
T088	1.5
T089	y^x
T090	÷
T091	STO R
****	Compute ψ
T092	RCL N
T093	RCL÷ R
T094	STO P
****	Compute t
T095	RCL F
T096	TAN
T097	STO T
****	Compute ω
T098	RCL L
T099	RCL— B
T100	→RAD
T101	STO W
****	Compute m and m_0
T102	1
T103	RCL E
T104	4
T105	÷
T106	—
T107	RCL E
T108	x^2
T109	0.046875
T110	×
T111	—
T112	RCL E
T113	3
T114	y^x
T115	0.01953125
T116	×
T117	—
T118	STO D
T119	RCL F
T120	→RAD
T121	×
T122	STO M

Line	Instruction
T123	RCL C
T124	→RAD
T125	RCL× D
T126	STO H
T127	RCL E
T128	RCL E
T129	x^2
T130	4
T131	÷
T132	+
T133	RCL E
T134	3
T135	y^x
T136	0.1171875
T137	×
T138	+
T139	0.375
T140	×
T141	STO D
T142	RCL F
T143	2
T144	×
T145	SIN
T146	×
T147	STO— M
T148	RCL C
T149	2
T150	×
T151	SIN
T152	RCL× D
T153	STO— H
T154	RCL E
T155	x^2
T156	RCL E
T157	3
T158	y^x
T159	0.75
T160	×
T161	+
T162	0.05859375
T163	×
T164	STO D
T165	RCL F
T166	4
T167	×

Latitude/Longitude to Transverse Mercator Co-ordinates

Line	Instruction
T168	SIN
T169	×
T170	STO+ M
T171	RCL C
T172	4
T173	×
T174	SIN
T175	RCL× D
T176	STO+ H
T177	RCL E
T178	3
T179	y^x
T180	35
T181	×
T182	3072
T183	÷
T184	STO D
T185	RCL F
T186	6
T187	×
T188	SIN
T189	×
T190	STO— M
T191	RCL A
T192	STO× M
T193	RCL C
T194	6
T195	×
T196	SIN
T197	RCL× D
T198	STO— H
T199	RCL A
T200	STO× H
****	Compute E
T201	RCL N
T202	RCL× W
T203	RCL F
T204	COS
T205	×
T206	STO X
T207	RCL F
T208	COS
T209	RCL× W
T210	3
T211	y^x

Line	Instruction
T212	RCL× N
T213	6
T214	÷
T215	RCL P
T216	RCL T
T217	x^2
T218	—
T219	×
T220	STO+ X
T221	1
T222	RCL T
T223	x^2
T224	6
T225	×
T226	—
T227	4
T228	×
T229	RCL P
T230	3
T231	y^x
T232	×
T233	RCL T
T234	x^2
T235	8
T236	×
T237	1
T238	+
T239	RCL P
T240	x^2
T241	×
T242	+
T243	RCL T
T244	x^2
T245	2
T246	×
T247	RCL× P
T248	—
T249	RCL T
T250	4
T251	y^x
T252	+
T253	RCL× N
T254	RCL F
T255	COS
T256	RCL× W

Line	Instruction
T257	5
T258	y^x
T259	×
T260	120
T261	÷
T262	STO+ X
T263	61
T264	RCL T
T265	x^2
T266	479
T267	×
T268	—
T269	RCL T
T270	4
T271	y^x
T272	179
T273	×
T274	+
T275	RCL T
T276	6
T277	y^x
T278	—
T279	RCL× N
T280	RCL F
T281	COS
T282	RCL× W
T283	7
T284	y^x
T285	×
T286	5040
T287	÷
T288	STO+ X
T289	RCL K
T290	STO× X
T291	RCL J
T292	STO+ X
****	Compute N
T293	RCL F
T294	COS
T295	RCL× W
T296	RCL× W
T297	2
T298	÷
T299	STO Y
T300	RCL P

Latitude/Longitude to Transverse Mercator Co-ordinates

Line	Instruction
T301	x^2
T302	4
T303	×
T304	RCL+ P
T305	RCL T
T306	x^2
T307	—
T308	RCL F
T309	COS
T310	3
T311	y^x
T312	×
T313	RCL W
T314	4
T315	y^x
T316	×
T317	24
T318	÷
T319	STO+ Y
T320	11
T321	RCL T
T322	x^2
T323	24
T324	×
T325	—
T326	8
T327	×
T328	RCL P
T329	4
T330	y^x
T331	×
T332	1
T333	RCL T
T334	x^2
T335	6
T336	×
T337	—
T338	28
T339	×
T340	RCL P
T341	3
T342	y^x
T343	×
T344	—
T345	1

Line	Instruction
T346	RCL T
T347	x^2
T348	32
T349	×
T350	—
T351	RCL× P
T352	RCL× P
T353	+
T354	RCL T
T355	x^2
T356	RCL× P
T357	2
T358	×
T359	—
T360	RCL T
T361	4
T362	y^x
T363	+
T364	RCL F
T365	COS
T366	5
T367	y^x
T368	×
T369	RCL W
T370	6
T371	y^x
T372	×
T373	720
T374	÷
T375	STO+ Y
T376	1385
T377	RCL T
T378	x^2
T379	3111
T380	×
T381	—
T382	RCL T
T383	4
T384	y^x
T385	543
T386	×
T387	+
T388	RCL T
T389	6
T390	y^x

Line	Instruction
T391	—
T392	RCL F
T393	COS
T394	7
T395	y^x
T396	×
T397	RCL W
T398	8
T399	y^x
T400	×
T401	40320
T402	÷
T403	STO+ Y
T404	RCL F
T405	SIN
T406	RCL× N
T407	STO× Y
T408	RCL M
T409	STO+ Y
T410	RCL K
T411	STO× Y
T412	RCL H
T413	RCL× K
T414	STO— Y
T415	RCL I
T416	STO+ Y
****	Compute γ
T417	0
T418	STO G
T419	RCL F
T420	SIN
T421	RCL× W
T422	STO— G
T423	RCL P
T424	x^2
T425	2
T426	×
T427	RCL— P
T428	RCL F
T429	COS
T430	x^2
T431	×
T432	RCL W
T433	3
T434	y^x

Latitude/Longitude to Transverse Mercator Co-ordinates

Line	Instruction
T435	×
T436	3
T437	÷
T438	RCL F
T439	SIN
T440	×
T441	STO— G
T442	11
T443	RCL T
T444	x^2
T445	24
T446	×
T447	—
T448	RCL P
T449	4
T450	y^x
T451	×
T452	11
T453	RCL T
T454	x^2
T455	36
T456	×
T457	—
T458	RCL P
T459	3
T460	y^x
T461	×
T462	—
T463	1
T464	RCL T
T465	x^2
T466	7
T467	×
T468	—
T469	2
T470	×
T471	RCL× P
T472	RCL× P
T473	+
T474	RCL T
T475	x^2
T476	RCL× P
T477	+
T478	RCL F
T479	COS

Line	Instruction
T480	4
T481	y^x
T482	×
T483	RCL W
T484	5
T485	y^x
T486	×
T487	15
T488	÷
T489	RCL F
T490	SIN
T491	×
T492	STO— G
T493	17
T494	RCL T
T495	x^2
T496	26
T497	×
T498	—
T499	RCL T
T500	4
T501	y^x
T502	2
T503	×
T504	+
T505	RCL F
T506	COS
T507	6
T508	y^x
T509	×
T510	RCL W
T511	7
T512	y^x
T513	×
T514	315
T515	÷
T516	RCL F
T517	SIN
T518	×
T519	STO— G
T520	RCL G
T521	→DEG
T522	→HMS
T523	STO G
****	Compute k

Line	Instruction
T524	1
T525	STO S
T526	RCL F
T527	COS
T528	RCL× W
T529	x^2
T530	2
T531	÷
T532	RCL× P
T533	STO+ S
T534	1
T535	RCL T
T536	x^2
T537	6
T538	×
T539	—
T540	RCL P
T541	3
T542	y^x
T543	×
T544	4
T545	×
T546	1
T547	RCL T
T548	x^2
T549	24
T550	×
T551	+
T552	RCL× P
T553	RCL× P
T554	+
T555	RCL T
T556	x^2
T557	RCL× P
T558	4
T559	×
T560	—
T561	RCL F
T562	COS
T563	RCL× W
T564	4
T565	y^x
T566	×
T567	24
T568	÷

Latitude/Longitude to Transverse Mercator Co-ordinates

Line	Instruction
T569	STO+ S
T570	61
T571	RCL T
T572	x^2
T573	148
T574	×
T575	—
T576	RCL T
T577	4
T578	y^x
T579	16
T580	×
T581	+
T582	RCL F
T583	COS
T584	RCL× W
T585	6
T586	y^x
T587	×
T588	720
T589	÷

Line	Instruction
T590	STO+ S
T591	RCL K
T592	STO× S
****	Show results
T593	RESULTS
T594	PSE
T595	EASTING
T596	PSE
T597	VIEW X
T598	NORTHING
T599	PSE
T600	VIEW Y
T601	GRID CONV
T602	PSE
T603	VIEW G
T604	PT SCALE FACT
T605	PSE
T606	VIEW S
T607	0
T608	STO Q
T609	NEXT PT [0—1]

Line	Instruction
T610	PSE
T611	INPUT Q
T612	RCL Q
T613	x = 0?
T614	GTO T624
T615	NEW ZONE [0—1]
T616	PSE
T617	0
T618	STO Q
T619	INPUT Q
T620	RCL Q
T621	x = 0?
T622	GTO T055
T623	GTO T028
T624	PROGRAM END
T625	PSE
T626	FS? 1
T627	CF 10
T628	RTN

Notes

(1) The program should be run in RPN mode, as results in ALG mode are unknown.

(2) Latitudes and longitudes should be entered in HP notation, i.e., DDD.MMSS. The grid convergence is displayed in HP notation.

(3) The program may be used for any Transverse Mercator projection, if the appropriate parameters are known. Similarly, any ellipsoid may be used, if its a and e^2 parameters are known. Parameters for a wide range of ellipsoids, all UTM zones and all SPCS TM zones are included at the end of this document.

(4) Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in North America. It is critical to enter the correct sign in calculator when entering values.

(5) Lines with **** are comments only, and should not be entered into the calculator. They are there to make program entry a little easier.

(6) This program is long and often appears to be a stream of meaningless commands. This means that it may be more prone to errors when being entered. It is suggested that the program be entered using the given constants, tested (and the checksum checked), and when it is satisfactory, the constants at the start of the program can be changed to those most suitable for the bulk of the expected work.

(7) The program allows the user to run additional points after each is completed, by prompting. If another point is to be processed, the user also has the option to move to a new zone and ellipsoid, otherwise the previous ellipsoid and projection parameters are

used. Respond 0 for 'NO' and 1 for 'YES' at the Q? prompt. If the user chooses to enter another point, the previous data entered is displayed at the prompts, but angular data are stored in decimal degrees. This should be re-entered in HP notation (or quickly converted with the ↱ →HMS key sequence), even if the same data is being used, because the program will convert the values to decimal degrees again, and so produce erroneous results.

(8) The program appears to work correctly, as tested. However, the grid convergence result has the opposite sign to that produced by the NGS on-line Lat/Long to SPCS conversion package at: http://www.ngs.noaa.gov/cgi-bin/spc_getpc.prl The formulae are correct in this program, and the results agree with the sign convention of Redfearn's formulae, as well as the normal usage of the grid convergence (converting between grid and true azimuths). I am not sure why the NGS program has the opposite sign, but I have asked NGS about it. Until this difference is resolved, be aware that the sign could be the opposite, and work out the correct sign from first principles.

Theory

Converting from geographical co-ordinates (latitude and longitude) to cartesian co-ordinates on a Transverse Mercator projection is a straightforward transformation, if somewhat long-winded.

Given that we have a, e^2, ϕ, k_0, λ and λ_0, we can use the following expressions for the conversion. These are Redfearn's Formulae. Note that these use an extra term in the computations of E' and N', compared to Snyder's book (1987), but this will make only a small difference in the overall values. The results will be a little different to the tabulated values for SPCS, too, owing to the limitations on the SPCS 27 computations. Remembering that the allowable distortion in the SPCS was to be no more than 1 in 10,000, it is acceptable to drop the final term in the formulae, as this doesn't degrade the formulae by anywhere near 1 in 10,000. These formulae will then agree with Snyder's formulae.

For UTM computations, you should use the full number of terms. This is because there is no 'legal' tolerance of distortion in the conversion process. UTM co-ordinates are now printed on 1:24,000 quadrangle maps, with either a grid/graticule or marginal ticks. These UTM co-ordinates are often on the NAD27 datum and need to be converted to NAD83 before they can be used. While there is a marginal note concerning the conversion of latitude and longitude from NAD27 to NAD83 on many of the more recent mapsheets, this value **does not** apply to the UTM co-ordinates (or the SPCS co-ordinates). This is because the latitude and longitude values are, in effect, figured from the origin in Kansas, while the UTM Northing co-ordinates are figured from the Equator. SPCS northings are figured from the zone origin, so will have a different shift for each zone. You should convert the co-ordinates to latitude and longitude for the appropriate system, convert these to NAD83, then convert to UTM or SPCS TM co-ordinates. An approximate set of shifts for UTM can be found in a paper by Welch, R., and Homsey, A., "Datum Shifts for UTM Co-ordinates," in the *Photogrammetric Engineering and Remote Sensing* journal, Volume 63, No. 4, pp. 371–375, published in 1997.

Latitude/Longitude to Transverse Mercator Co-ordinates

Conversion Formulae

Easting

$$
\begin{aligned}
E' = k_0 \{ \; & \nu \omega \cos \phi \\
& + \nu \frac{\omega^3}{6} \cos^3 \phi \, (\psi - t^2) \\
& + \nu \frac{\omega^5}{120} \cos^5 \phi \, [4 \psi^3 (1 - 6 t^2) + \psi^2 (1 + 8 t^2) - \psi (2 t^2) + t^4] \\
& + \nu \frac{\omega^7}{5040} \cos^7 \phi \, (61 - 479 t^2 + 179 t^4 - t^6) \; \}
\end{aligned}
$$

Northing

$$
\begin{aligned}
N' = k_0 \{ \; m \; & + \nu \sin \phi \frac{\omega^2}{2} \cos \phi \\
& + \nu \sin \phi \frac{\omega^4}{24} \cos^3 \phi \, (4 \psi^2 + \psi - t^2) \\
& + \nu \sin \phi \frac{\omega^6}{720} \cos^5 \phi \, [8\psi^4(11-24t^2) - 28\psi^3(1-6t^2) + \psi^2(1-32t^2) - 2\psi t^2 + t^4] \\
& + \nu \sin \phi \frac{\omega^8}{40320} \cos^7 \phi \, (1385 - 3111 t^2 + 543 t^4 - t^6) \; \}
\end{aligned}
$$

Grid Convergence (in radians)

$$
\begin{aligned}
\gamma = \; & - \sin \phi \; \omega \\
& - \sin \phi \; \frac{\omega^3}{3} \cos^2 \phi \, (2 \psi^2 - \psi) \\
& - \sin \phi \; \frac{\omega^5}{15} \cos^4 \phi \, [\psi^4 (11 - 24 t^2) - \psi^3 (11 - 36 t^2) + 2\psi^2 (1 - 7t^2) + \psi t^2] \\
& - \sin \phi \; \frac{\omega^7}{315} \cos^6 \phi \, (17 - 26 t^2 + 2t^4)
\end{aligned}
$$

Point Scale Factor

$$
\begin{aligned}
k = k_0 \{ 1 \; & + \frac{\omega^2}{2} \cos^2 \phi \; \psi \\
& + \frac{\omega^4}{24} \cos^4 \phi \; [4 \psi^3 (1 - 6 t^2) + \psi^2 (1 + 24 t^2) - 4 \psi t^2] \\
& + \frac{\omega^6}{720} \cos^6 \phi \; (61 - 148 t^2 + 16 t^4) \; \}
\end{aligned}
$$

where

$E' = E - {}_0E$ (E_0 is the offset of the central meridian; check the value for each zone. For UTM, E_0 = 500 000·000 meters.)

Latitude/Longitude to Transverse Mercator Co-ordinates

$N' = N - N_0$ (N_0 is the offset of the origin latitude; check the value for each zone. For UTM in the northern hemisphere, $N_0 = 0$; for UTM in the southern hemisphere, N_0 = 10 000 000·000 meters.)

ν = radius of curvature in the prime vertical at ϕ; i.e. $\nu = \dfrac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$

$$\rho = \frac{a\,(1 - e^2)}{(1 - e^2 \sin^2 \phi)^{\frac{3}{2}}}$$ = radius of curvature in the meridian at ϕ

$\omega = \lambda - \lambda_0$

$\psi = \dfrac{\nu}{\rho}$ i.e. ratio of the radii of curvature at ϕ

$t = \tan \phi$

m = meridian distance from equator, computed using the following expression

$m = a\,(A_0\,\phi - A_2 \sin 2\phi + A_4 \sin 4\phi - A_6 \sin 6\phi)$

where ϕ is in radians and

$$A_0 = 1 - \frac{e^2}{4} - \frac{3e^4}{64} - \frac{5e^6}{256}$$

$$A_2 = \frac{3}{8}\left(e^2 + \frac{e^4}{4} + \frac{15e^6}{128}\right)$$

$$A_4 = \frac{15}{256}\left(e^4 + \frac{3e^6}{4}\right)$$

$$A_6 = \frac{35e^6}{3072}$$

With the appropriate values for ellipsoids and scale factors, these formulae will work for any Transverse Mercator projection: UTM, SPCS, AMG, MGA or whatever.

a = semi-major axis of the ellipsoid; a = 6,378,137 m for WGS84 (GRS80)

e^2 = eccentricity of the ellipsoid; e^2 = 0.006 694 3800 for WGS84.

Sample Computations

Example 1

Using the SPCS 1983 (a = 6,378,137 m, e^2 = 0.006 694 3800), the following results are obtained.

Nevada East Zone, 2701, λ_0 = –115° 35', ϕ_0 = 34° 45', k_0 = 0.999 900, E_0 = 200,000.000 m, N_0 = 8,000,000.000 m.

Latitude = 41° 25' 00" Longitude = –115° 45' 20"

Easting (E) = 185,603.123 m Northing (N) = 8,739,929.417 m

Grid Convergence (γ) = 0° 06' 50.1" Point Scale Factor (k) = 0.999 902 55

Example 2

Using the SPCS 1927 ($a = 20925832.2$ ft, $e^2 = 0.006\ 768\ 66$), the following results are obtained.

Nevada East Zone, SPCS 1927, $\lambda_0 = -115° 35'$, $\phi_0 = 34° 45'$, $k_0 = 0.999\ 900$, $E_0 = 500{,}000.000$ ft, $N_0 = 0.000$ ft.

Latitude = 41° 25' 00" Longitude = –115° 45' 20"

Easting (E) = 452,764.960 ft Northing (N) = 2,427,533.222 ft

Grid Convergence (γ) = 0° 06' 50.1" Point Scale Factor (k) = 0.999 902 55

Example 3

Using the ANS ellipsoid ($a = 6{,}378{,}160$ m, $e^2 = 0.006\ 694\ 541\ 855$), the following results are obtained.

AMG Zone 54, $\lambda_0 = +141° 00'$, $\phi_0 = 0° 00'$, $k_0 = 0.999\ 600$,
$E_0 = 500{,}000.000$ m, $N_0 = 10{,}000{,}000.000$ m.

Latitude = –37° 39' 15".5571 Longitude = +143° 55' 30".6330

Easting (E) = 758,053.090 m Northing (N) = 5,828,496.973 m

Grid Convergence (γ) = +1° 47' 16.67" Point Scale Factor (k) = 1.000 420 30

These results agree with those computed in the AGD Technical Manual, 1986.

Running the Program

Press XEQ T , then the ENTER key, to start the program. The calculator briefly displays LAT—LONG 2 TM, then briefly shows CHECK—ENTER A. This is "Point A," discussed below. The program then stops and displays the prompt for entering the semi-major axis value, while displaying the current default value:

A?
6,378,137.0000 (This is for GRS80/WGS84/NAD83)

If you are happy with this value for the semi-major axis of the ellipsoid, press R/S to continue. Otherwise. Key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHECK—ENTER E. The program then stops and displays the prompt for entering the eccentricity of the ellipsoid, e:

E?
0.00669438 (This is for GRS80/WGS84/NAD83)

If this value for the eccentricity is correct, press R/S to continue. Otherwise, key in a different value (for a different ellipsoid) and press R/S to continue.

The calculator briefly displays CHECK—ENTER K. The program then stops and displays the prompt for entering the scale factor at the central meridian (λ_0), which is k_0:

K?
0.9996000 (This is for UTM)

If this value for the scale factor is satisfactory, press R/S to continue. If you want to change it, such as for an SPCS zone, key in the correct value and press R/S. In this case, key in 0.9999 for Nevada East (2701).

The calculator briefly displays CHK—NTR LONG 0. The program then stops and displays the prompt for entering the longitude of the central meridian of the projection, λ_0. Note that in the western hemisphere, this will be a negative value, and should be in HP notation (DDD.MMSS).

B?
—81.000000 (This is for UTM Zone 17)

If this is the correct central meridian, press R/S to continue, if this is not correct, key in the correct value, in HP notation, then press R/S to continue. In this case, key in –115.35 for Nevada East (2701).

The calculator briefly displays CHK—NTR LAT 0. The program then stops and displays the prompt for entering the latitude of the Northing co-ordinate origin, ϕ_0. For UTM, this is 0.000 (the equator), while for SPCS Zones, it is usually a latitude well south of the zone. The value should be entered in HP notation.

C?
0.000000 (This is for UTM)

If this is the correct latitude base, press R/S to continue. If you want a different value, key in that value and press R/S to continue. In this case, key in 34.45 and press R/S to continue.

The calculator briefly displays CHK—NTR N0. The program then stops and prompts for entry of the false northing value, or the northing offset. This is the value of the northing co-ordinate at ϕ_0, λ_0. For UTM in the northern hemisphere, this is 0.0000, while its value varies for different SPCS zones.

I?
0.0000 (This is for UTM)

If this is the correct value, press R/S to continue. If a different value is desired, key in the value and press R/S. In this case, key in 8,000,000.0 and press R/S. This is the N_0 value for Nevada East 2701.

The calculator briefly displays CHK—NTR E0. The program then stops and prompts for the false easting value, or the easting offset. This is the value of the easting at the central meridian (λ_0), denoted E_0.

J?
500,000.0000 (This is for UTM)

If this is the correct value, press R/S to continue. If a different value is required, key in the value and press R/S. In this case, key in 200,000.000 and press R/S. This is the E_0 value for Nevada East 2701.

This is “Point B,” discussed below. The calculator briefly displays ENTER PT LAT. The program stops and displays the prompt for entering the latitude of the point to be converted. This should be in HP notation.

F?
0.0000

Key in the latitude of the point in HP notation and press R/S to continue. In this case, key in 41.25 and press R/S.

Latitude/Longitude to Transverse Mercator Co-ordinates

The calculator briefly displays ENTER PT LONG. The program then stops and displays the prompt for entering the longitude of the point to be converted. This should be in HP notation.

L?
0.0000

Key in the longitude of the point in HP notation and press R/S to continue. In this case, key in —115.452 and press R/S.

The program displays RUNNING for a short while, then displays RESULTS briefly, followed by EASTING briefly. The program then stops and displays the easting value of the point. In this case, the calculator displays:

X=
185,603.1225

This is the easting of the point, in this case in meters. Press R/S to continue. The calculator briefly displays NORTHING, then stops and displays the northing value of the point. In this case, the calculator displays:

Y=
8,739,929.4173

This is the northing of the point, in this case in meters. Press R/S to continue. The calculator briefly displays GRID CONV, then stops and displays the grid convergence value in HP notation. In this case, the calculator displays:

G=
0.0650149

This is the grid convergence in HP notation, and is 0° 06' 50".149 in more conventional notation. Press R/S to continue. The calculator briefly displays PT SCALE FACT, then stops and displays the point scale factor of the point on the Transverse Mercator projection. In this case, the calculator displays:

S=
0.999902550

This is the point scale factor. Press R/S to continue.

You now have the choice of running one or more additional points. The calculator briefly displays NEXT PT [0—1], then stops and displays the prompt for answering questions:

Q?
0.0000

If you want to quit the program, just press R/S. If you want to enter more points, key in 1 and press R/S. In this case, the calculator then prompts to see if you want to use the same parameters. The calculator briefly displays NEW ZONE [0—1], then stops at the question prompt:

Q?
0.0000

If you want to go to a new zone, key in 1 and press R/S, and the calculator will take you to the point where you can change any of the values (Point A above), starting with the ellipsoid parameters. If you want to work in the same zone already entered, just press R/S, and the program will take you to “Point B” and prompt for the latitude of the point to be converted, and continue from there. You can go around the program as many times as necessary.

When you choose to end the program, the calculator briefly displays PROGRAM END and then comes to an end, returning to the point from which it was called, or to normal operations, and resetting Flag 10.

Storage Registers Used

A	Semi-major axis of the ellipsoid being used, a
B	λ_0, the central meridian of the projection
C	ϕ_0, the origin latitude for the co-ordinates
D	temporary storage
E	Eccentricity of the ellipsoid, e^2
F	λ, latitude of the point to be converted
G	γ, the grid convergence of the point being converted
H	meridian distance of the origin latitude, ϕ_0
I	N_0, the offset for the northings (the northing at ϕ_0, λ_0)
J	E_0, the offset for the eastings (the easting at the central meridian, λ_0)
K	k_0, the scale factor at the central meridian, λ_0
L	λ, longitude of the point to be converted
M	m, meridian distance of the point to be converted
N	ν
P	$\psi = \frac{\nu}{\rho}$
Q	used for getting responses to questions about running more points
R	ρ
S	k, point scale factor at the point being converted
T	$\tan \phi$
W	$\omega = \lambda - \lambda_0$
X	Easting co-ordinate of converted point
Y	Northing co-ordinate of converted point

Statistical Registers: not used

Labels Used

Label **T** Length = 2332 Checksum = AAF9

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computations to check proper operation after entry.

Latitude/Longitude to Transverse Mercator Co-ordinates

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Parameters for the Computations

Universal Transverse Mercator (UTM)

For UTM, the ϕ_0 value is 0° (the equator) for both northern and southern hemispheres. The λ_0 values are given for each zone in the table below.

Zone	Central Meridian, λ_0	Zone	Central Meridian, λ_0
1	177° W	31	3°E
2	171° W	32	9°E
3	165° W	33	15°E
4	159° W	34	21°E
5	153° W	35	27°E
6	147° W	36	33°E
7	141° W	37	39°E
8	135° W	38	45°E
9	129° W	39	51°E
10	123° W	40	57°E
11	117° W	41	63°E
12	111° W	42	69°E
13	105° W	43	75°E
14	99° W	44	81°E
15	93° W	45	87°E
16	87° W	46	93°E
17	81° W	47	99°E
18	75° W	48	105°E
19	69° W	49	111°E
20	63° W	50	117°E
21	57° W	51	123°E
22	51° W	52	129°E
23	45° W	53	135°E
24	39° W	54	141°E
25	33° W	55	147°E
26	27° W	56	153°E
27	21° W	57	159°E
28	15° W	58	165°E
29	9° W	59	171°E
30	3° W	60	177°E

The E_0 value for all zones is 500,000.000 meters. The N_0 value for the northern hemisphere is 0.000 meters. The N_0 value for the southern hemisphere is 10,000,000.000 meters.

State Plane Co-ordinate System (SPCS) 1983

Several US states use the Transverse Mercator projection for SPCS 1983. The various parameters for each zone in the 1983 system are given in the table below. Use these parameters with the program, together with the GRS80/WGS84/NAD83 ellipsoid parameters, in meters.

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
Alabama					
East	85° 50'	30° 30'	0.9999600	200000.00	0.00
West	87° 30'	30° 00'	0.9999333	600000.00	0.00
Alaska					
2	142° 00'	54° 00'	0.9999000	500000.00	0.00
3	146° 00'	54° 00'	0.9999000	500000.00	0.00
4	150° 00'	54° 00'	0.9999000	500000.00	0.00
5	154° 00'	54° 00'	0.9999000	500000.00	0.00
6	185° 00'	54° 00'	0.9999000	500000.00	0.00
7	162° 00'	54° 00'	0.9999000	500000.00	0.00
8	166° 00'	54° 00'	0.9999000	500000.00	0.00
9	170° 00'	54° 00'	0.9999000	500000.00	0.00
Arizona					
East	110° 10'	31° 00'	0.9999000	213360.00	0.00
Central	111° 55'	31° 00'	0.9999000	213360.00	0.00
West	113° 45'	31° 00'	0.9999333	213360.00	0.00
Delaware	72° 25'	38° 00'	0.9999950	200000.00	0.00
Florida					
East	81° 00'	24° 20'	0.9999412	200000.00	0.00
West	82° 00'	24° 20'	0.9999412	200000.00	0.00
Georgia					
East	82° 10'	30° 00'	0.9999000	200000.00	0.00
West	84° 10'	30° 00'	0.9999000	700000.00	0.00
Hawaii					
1	155° 30'	18° 50'	0.9999667	500000.00	0.00
2	156° 40'	20° 20'	0.9999667	500000.00	0.00
3	158° 00'	21° 10'	0.9999900	500000.00	0.00
4	159° 30'	21° 50'	0.9999900	500000.00	0.00
5	160° 10'	21° 40'	1.0000000	500000.00	0.00

Latitude/Longitude to Transverse Mercator Co-ordinates

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
Idaho					
East	112° 10'	41° 40'	0.9999474	200000.00	0.00
Central	114° 00'	41° 40'	0.9999474	500000.00	0.00
Illinois					
East	88° 20'	36° 40'	0.9999750	300000.00	0.00
West	90° 10'	36° 40'	0.9999412	700000.00	0.00
Indiana					
East	85° 40'	37° 30'	0.9999667	100000.00	250000.00
West	87° 05'	37° 30'	0.9999667	900000.00	250000.00
Maine					
East	68° 30'	43° 40'	0.9999000	300000.00	0.00
West	70° 10'	42° 50'	0.9999667	900000.00	0.00
Mississippi					
East	88° 50'	29° 30'	0.9999500	300000.00	0.00
West	90° 20'	29° 30'	0.9999500	700000.00	0.00
Missouri					
East	90° 30'	35° 50'	0.9999333	250000.00	0.00
Central	92° 30'	35° 50'	0.9999333	500000.00	0.00
West	94° 30'	36° 10'	0.9999412	850000.00	0.00
Nevada					
East	115° 35'	34° 45'	0.9999000	200000.00	8000000.00
Central	116° 40'	34° 45'	0.9999000	500000.00	6000000.00
West	118° 35'	34° 45'	0.9999000	800000.00	4000000.00
New Hampshire	71° 40'	42° 30'	0.9999667	300000.00	0.00
New Jersey	74° 30'	38° 50'	0.9999000	150000.00	0.00
New Mexico					
East	104° 20'	31° 00'	0.9999091	165000.00	0.00
Central	106° 15'	31° 00'	0.9999000	500000.00	0.00
West	107° 50'	31° 00'	0.9999167	830000.00	0.00

Latitude/Longitude to Transverse Mercator Co-ordinates

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
New York					
East	74° 30'	40° 00'	0.9999000	150000.00	0.00
Central	76° 35'	40° 00'	0.9999375	250000.00	0.00
West	78° 35'	40° 00'	0.9999375	350000.00	0.00
Rhode Island	71° 30'	41° 05'	0.9999938	100000.00	0.00
Vermont	72° 30'	42° 30'	0.9999643	500000.00	0.00
Wyoming					
East	105° 10'	40° 30'	0.9999375	200000.00	0.00
East Central	107° 20'	40° 30'	0.9999375	400000.00	100000.00
West Central	108° 45'	40° 30'	0.9999375	600000.00	0.00
West	110° 05'	40° 30'	0.9999375	800000.00	100000.00

State Plane Co-ordinate System (SPCS) 1927

Several US states used the Transverse Mercator projection for SPCS 1927. The various parameters for each zone in the 1927 system are given in the table below. Use these parameters with the program, together with the Clarke 1866 ellipsoid in feet.

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Alabama					
East	85° 50'	30° 30'	0.9999600	500000.00	0.00
West	87° 30'	30° 00'	0.9999333	500000.00	0.00
Alaska					
2	142° 00'	54° 00'	0.9999000	500000.00	0.00
3	146° 00'	54° 00'	0.9999000	500000.00	0.00
4	150° 00'	54° 00'	0.9999000	500000.00	0.00
5	154° 00'	54° 00'	0.9999000	500000.00	0.00
6	185° 00'	54° 00'	0.9999000	500000.00	0.00
7	162° 00'	54° 00'	0.9999000	700000.00	0.00
8	166° 00'	54° 00'	0.9999000	500000.00	0.00
9	170° 00'	54° 00'	0.9999000	600000.00	0.00

Latitude/Longitude to Transverse Mercator Co-ordinates

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Arizona					
East	110° 10'	31° 00'	0.9999000	500000.00	0.00
Central	111° 55'	31° 00'	0.9999000	500000.00	0.00
West	113° 45'	31° 00'	0.9999333	500000.00	0.00
Delaware	72° 25'	38° 00'	0.9999950	500000.00	0.00
Florida					
East	81° 00'	24° 20'	0.9999412	500000.00	0.00
West	82° 00'	24° 20'	0.9999412	500000.00	0.00
Georgia					
East	82° 10'	30° 00'	0.9999000	500000.00	0.00
West	84° 10'	30° 00'	0.9999000	500000.00	0.00
Hawaii					
1	155° 30'	18° 50'	0.9999667	500000.00	0.00
2	156° 40'	20° 20'	0.9999667	500000.00	0.00
3	158° 00'	21° 10'	0.9999900	500000.00	0.00
4	159° 30'	21° 50'	0.9999900	500000.00	0.00
5	160° 10'	21° 40'	1.0000000	500000.00	0.00
Idaho					
East	112° 10'	41° 40'	0.9999474	500000.00	0.00
Central	114° 00'	41° 40'	0.9999474	500000.00	0.00
West	115° 45'	41° 40'	0.9999333	500000.00	0.00
Illinois					
East	88° 20'	36° 40'	0.9999750	500000.00	0.00
West	90° 10'	36° 40'	0.9999412	500000.00	0.00
Indiana					
East	85° 40'	37° 30'	0.9999667	500000.00	0.00
West	87° 05'	37° 30'	0.9999667	500000.00	0.00
Maine					
East	68° 30'	43° 50'	0.9999000	500000.00	0.00
West	70° 10'	42° 50'	0.9999667	500000.00	0.00

Latitude/Longitude to Transverse Mercator Co-ordinates

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Michigan (old)					
East	83° 40'	41° 30'	0.9999429	500000.00	0.00
Central	85° 45'	41° 30'	0.9999091	500000.00	0.00
West	88° 45'	41° 30'	0.9999091	500000.00	0.00
Mississippi					
East	88° 50'	29° 40'	0.9999600	500000.00	0.00
West	90° 20'	30° 30'	0.9999412	500000.00	0.00
Missouri					
East	90° 30'	35° 50'	0.9999333	500000.00	0.00
Central	92° 30'	35° 50'	0.9999333	500000.00	0.00
West	94° 30'	36° 10'	0.9999412	500000.00	0.00
Nevada					
East	115° 35'	34° 45'	0.9999000	500000.00	0.00
Central	116° 40'	34° 45'	0.9999000	500000.00	0.00
West	118° 35'	34° 45'	0.9999000	500000.00	0.00
New Hampshire	71° 40'	42° 30'	0.9999667	500000.00	0.00
New Jersey	74° 40'	38° 50'	0.9999750	2000000.00	0.00
New Mexico					
East	104° 20'	31° 00'	0.9999091	500000.00	0.00
Central	106° 15'	31° 00'	0.9999000	500000.00	0.00
West	107° 50'	31° 00'	0.9999167	500000.00	0.00
New York					
East	74° 20'	40° 00'	0.9999667	500000.00	0.00
Central	76° 35'	40° 00'	0.9999375	500000.00	0.00
West	78° 35'	40° 00'	0.9999375	500000.00	0.00
Rhode Island	71° 30'	41° 05'	0.9999938	500000.00	0.00
Vermont	72° 30'	42° 30'	0.9999643	500000.00	0.00

Latitude/Longitude to Transverse Mercator Co-ordinates

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Wyoming					
East	105° 10'	40° 40'	0.9999412	500000.00	0.00
East Central	107° 20'	40° 40'	0.9999412	500000.00	0.00
West Central	108° 45'	40° 40'	0.9999412	500000.00	0.00
West	110° 05'	40° 40'	0.9999412	500000.00	0.00

Ellipsoids

There are a range of ellipsoids in common or former use. The table below has the a and e^2 values for a number of common (and less common) ellipsoids.

Ellipsoid	a Semi-major Axis	e^2 Eccentricity
GRS80–WGS94–NAD83	6378137 m	0.006 694 38
Clarke 1866 (NAD27)	6378206.4 m	0.006 768 66
Clarke 1866 (NAD27)	20925832.2 ft	0.006 768 66
ANS (Australian)	6378160 m	0.006 694 541 855
Airy 1830	6377563.4 m	0.006 670 54
Bessel 1841	6377397.16 m	0.006 674 372
Clarke 1880	6378249.15 m	0.006 803 511
Everest 1830	6377276.35 m	0.006 637 847
Fischer 1960 (Mercury)	6378166 m	0.006 693 422
Fischer 1968	6378150 m	0.006 693 422
Hough 1956	6378270 m	0.006 722 67
International	6378388 m	0.006 722 67
Krassovsky 1940	6378245 m	0.006 693 422
South American 1960	6378160 m	0.006 694 542
GRS 1967	6378160 m	0.006 694 605
GRS 1975	6378140 m	0.006 694 385
WGS 60	6378165 m	0.006 693 422
WGS 66	6378145 m	0.006 694 542
WGS 72	6378135 m	0.006 694 317 778
WGS 84	6378137 m	0.006 694 38

Reference

SNYDER, J.P., 1987. *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper 1395. Washington: US Government Printing Office.

HP-35s Calculator Program **TM 2**

Convert Transverse Mercator Co-ordinates (UTM, SPCS, etc.) to Latitude and Longitude

Programmer: Dr. Bill Hazelton

Date: April, 2008. **Version**: 1.0 **Mnemonic**: **Y** for TM X & **Y** to Lat/Long

Line	Instruction	Display	User Instructions
Y001	LBL Y		LBL Y
Y002	CLSTK		CLEAR 5
Y003	FS? 10		FLAGS 3 .0
Y004	GTO Y008		
Y005	SF 1		FLAGS 1 1
Y006	SF 10		FLAGS 1 .0
Y007	GTO Y009		
Y008	CF 1		FLAGS 2 1
Y009	TM 2 LAT–LONG		(Key in using EQN RCL T, RCL M, etc.)
Y010	PSE		PSE
Y011	CL x		CLEAR 1
Y012	STO X		STO X
Y013	STO Y		STO Y
Y014	6378137		a value for ellipsoid (WGS84)
Y015	STO A		STO A
Y016	0.00669438		e^2 value for ellipsoid (WGS84)
Y017	STO E		STO E
Y018	0.9996		k_0 value for zone (UTM 17)
Y019	STO K		STO K
Y020	–81		λ_0 for zone (UTM 17)
Y021	STO D		STO D
Y022	0		ϕ_0 for zone (UTM 17)
Y023	STO C		STO C
Y024	500000		E_0 for zone (UTM 17)
Y025	STO I		STO I
Y026	0		N_0 for zone (UTM 17)
Y027	STO J		STO J
Y028	CHECK–ENTER A		(Key in using EQN RCL C, RCL H, etc.)
Y029	PSE		PSE
Y030	INPUT A		INPUT A
Y031	CHECK–ENTER E		(Key in using EQN RCL C, RCL H, etc.)
Y032	PSE		PSE
Y033	INPUT E		INPUT E
Y034	CHECK–ENTER K		(Key in using EQN RCL C, RCL H, etc.)
Y035	PSE		PSE
Y036	INPUT K		INPUT K
Y037	CHK–NTR LONG 0		(Key in using EQN RCL C, RCL H, etc.)
Y038	PSE		PSE

Transverse Mercator Co-ordinates to Latitude/Longitude

Line	Instruction
Y039	INPUT D
Y040	CHK–NTR LAT 0
Y041	PSE
Y042	INPUT C
Y043	CHK–NTR E 0
Y044	PSE
Y045	INPUT I
Y046	CHK–NTR N 0
Y047	PSE
Y048	INPUT J
Y049	RCL D
Y050	HMS→
Y051	STO D
Y052	RCL C
Y053	HMS→
Y054	STO C
Y055	ENTER EASTING
Y056	PSE
Y057	INPUT X
Y058	ENTER NORTHING
Y059	PSE
Y060	INPUT Y
Y061	RCL I
Y062	STO— X
Y063	RCL J
Y064	STO— Y
****	Compute b
Y065	1
Y066	RCL— E
Y067	√x
Y068	RCL× A
Y069	STO B
****	Compute n
Y070	RCL A
Y071	RCL— B
Y072	RCL A
Y073	RCL+ B
Y074	÷
Y075	STO O
****	Compute ΔN
Y076	RCL Y
Y077	RCL÷ K
Y078	STO M
****	Compute m_0 and m
Y079	RCL C

Line	Instruction
Y080	→RAD
Y081	CF 10
Y082	(1 – E ÷ 4 – 3 × E^2 ÷ 64 – 5 × E^3 ÷ 256)
Y083	×
Y084	STO V
Y085	0.375 × (E + E^2 ÷ 4 + 15 × E^3 ÷ 128) × SIN(2 × C)
Y086	STO— V
Y087	15 ÷ 256 × (E^2 + 0.75 × E^3) × SIN(4 × C)
Y088	STO+ V
Y089	35 × E^3 ÷ 3072 × SIN(6 × C)
Y090	STO— V
Y091	RCL A
Y092	RCL× V
Y093	STO+ M
****	Compute G and σ
Y094	RAD [Key in as MODE 2]
Y095	(1 + 2.25 × O^2 + 225× O^4 ÷ 64) × (1 – O) × (1 – O^2)
Y096	RCL× A
Y097	RCL M
Y098	x < > y
Y099	÷
Y100	STO S
****	Compute ϕ' (the foot-point latitude)
Y101	RCL S
Y102	STO F
Y103	(1.5 × O – 25 × O^3 ÷ 32) × SIN(2 × S)
Y104	STO+ F
Y105	(21 × O^2 ÷ 16 – 55 × O^4 ÷ 32) × SIN(4 × S)
Y106	STO+ F
Y107	151 × O^3 ÷ 96 × SIN(6 × S)
Y108	STO+ F
Y109	1097 × O^4 ÷ 512 × SIN(8 × S)
Y110	STO+ F
****	Compute ν', t', ρ', ψ' and x
Y111	RCL F
Y112	TAN
Y113	STO T
Y114	A ÷ SQRT(1 — E × SIN(F)^2)
Y115	STO N
Y116	A × (1 — E) ÷ ((1 — E × SIN(F)^2)^1.5)
Y117	STO R
Y118	N ÷ R
Y119	STO P
Y120	X ÷ N ÷ K
Y121	STO U

Transverse Mercator Co-ordinates to Latitude/Longitude

Line	Instruction
****	Compute ϕ (latitude)
Y122	U × T × X ÷ K ÷ R ÷ 2
Y123	STO— F
Y124	(12 × T^2 + 9 × P × (1 — T^2) — 4 × P^2) × U^3 × X × T ÷ K ÷ R ÷ 24
Y125	STO+ F
Y126	(8 × P^4 × (11 – 24 × T^2) – 12 × P^3 × (21 – 71 × T^2) + 15 × P^2 × (15 – 98 × T^2 +
†	… 15 × T^4) + 180 × P × (5 × T^2 – 3 × T^4) + 360 × T^4) × X × T ÷ K ÷ R ÷ 720
Y127	RCL U
Y128	5
Y129	y^x
Y130	×
Y131	STO— F
Y132	(1385 + 3633 × T^2 + 4095 × T^4 + 1575 × T^6) × X × T ÷ K ÷ R ÷ 40320 × U^7
Y133	STO+ F
****	Calculate λ (longitude)
Y134	1
Y135	STO W
Y136	(P + 2 × T^2) × U^2 ÷ 6
Y137	STO— W
Y138	(24 × T^4 + 72 × P × T^2 + P^2 × (9 – 68 × T^2) – 4 × P^3 × (1 – 6 × T^2)) × U^4 ÷ 120
Y139	STO+ W
Y140	(61 + 662 × T^2 + 1320 × T^4 + 720 × T^6) × U^6 ÷ 5040
Y141	STO— W
Y142	RCL U
Y143	STO× W
Y144	RCL T
Y145	ATAN
Y146	COS
Y147	STO÷ W
****	Calculate γ (grid convergence)
Y148	—1
Y149	STO G
Y150	(T^2 + 3 × P – 2 × P^2) × U^2 ÷ 3
Y151	STO+ G
Y152	(P^4 × (11 – 24 × T^2) – 3 × P^3 × (8 – 23 × T^2) + 5 × P^2 × (3 – 14 × T^2) + 30 × P
†	… × T^2 + 3 × T^4) × U^4 ÷ 15
Y153	STO— G
Y154	(17 + 77 × T^2 + 105 × T^4 + 45 × T^6) × U^8 ÷ 315
Y155	STO+ G
Y156	RCL T
Y157	RCL× U
Y158	STO× G
****	Calculate k (point scale factor)
Y159	RCL X
Y160	STO× U

Transverse Mercator Co-ordinates to Latitude/Longitude

Line	Instruction
Y161	RCL K
Y162	STO÷ U
Y163	RCL R
Y164	STO÷ U
Y165	((4 × P × (1 – 6 × T^2) – 3 × (1 – 16 × T^2) – 24 × T^2 ÷ P) × U^2 ÷ 24 + U ÷ 2 + 1 +
†	… U^3 ÷ 720) × K
Y166	STO S

Line	Instruction
****	Prepare results
Y167	DEG [Key in as MODE 1]
Y168	SF 10
Y169	RCL F
Y170	→DEG
Y171	→HMS
Y172	STO F
Y173	RCL W
Y174	→DEG
Y175	RCL+ D
Y176	→HMS
Y177	STO L
Y178	RCL G
Y179	→DEG
Y180	→HMS
Y181	STO G
****	Show results
Y182	RESULTS
Y183	PSE
Y184	LATITUDE
Y185	PSE
Y186	VIEW F
Y187	LONGITUDE
Y188	PSE
Y189	VIEW L
Y190	GRID CONV
Y191	PSE

Line	Instruction
Y192	VIEW G
Y193	PT SCALE FACT
Y194	PSE
Y195	VIEW S
Y196	0
Y197	STO Q
Y198	NEXT PT [0—1]
Y199	PSE
Y200	INPUT Q
Y201	RCL Q
Y202	x = 0?
Y203	GTO Y213
Y204	NEW ZONE [0—1]
Y205	PSE
Y206	0
Y207	STO Q
Y208	INPUT Q
Y209	RCL Q
Y210	x = 0?
Y211	GTO Y055
Y212	GTO Y028
Y213	PROGRAM END
Y214	PSE
Y215	FS? 1
Y216	CF 10
Y217	RTN

Notes

(1) The program should be run in RPN mode, as results in ALG mode are unknown.

(2) Latitudes and longitudes should be entered in HP notation, i.e., DDD.MMSS. The grid convergence is displayed in HP notation.

(3) The program may be used for any Transverse Mercator projection, if the appropriate parameters are known. Similarly, any ellipsoid may be used, if its a and e^2 parameters are known. Parameters for a wide range of ellipsoids, all UTM zones and all SPCS TM zones are included at the end of this document.

(4) Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in North America. It is critical to enter the correct sign in the calculator when entering values.

(5) Lines with **** are comments only, and should not be entered into the calculator. They are there to make program entry a little easier.

(6) This program is long and often appears to be a stream of meaningless commands. This means that it may be more prone to errors when being entered. It is suggested that the program be entered using the given constants, tested (and the checksum checked), and when it is satisfactory, the constants at the start of the program can be changed to those most suitable for the bulk of the expected work.

(7) In order to reduce the apparent length of the program (which otherwise would have been well over 600 lines), equations were used for the bulk of the computations. Some equations are too long to fit on a single line in the above listing, so are continued to the line below (Y126, Y152, Y165. In this case, the line number is replaced by a † and a … appears at the start of the continuing line. Neither the † nor the … should be entered into the calculator.

The use of equations, rather than direct instruction code, does slow the computation process a little, but makes the program a bit shorter and so possibly easier to enter.

(8) The program allows the user to run additional points after each is completed, by prompting. If another point is to be processed, the user also has the option to move to a new zone and ellipsoid, otherwise the previous ellipsoid and projection parameters are used. Respond 0 for 'NO' and 1 for 'YES' at the Q? prompt. If the user chooses to enter another point, the previous data entered is displayed at the prompts, but angular data are stored in decimal degrees. This should be re-entered in HP notation (or quickly converted with the ↱ →HMS key sequence), even if the same data is being used, because the program will convert the values to decimal degrees again, and so produce erroneous results.

(9) There are two lines where the calculator's mode is changed from DEGREES to RADIANS, and *vice versa*. The instructions for these lines (Y094 and Y167) are keyed in using the MODE button. A note about this is included on the relevant line of the code, in red, to avoid confusion with the →RAD and →DEG function, which are used elsewhere in the program.

(10) The program appears to work correctly, as tested. However, the grid convergence result has the opposite sign to that produced by the NGS on-line Lat/Long to SPCS conversion package at: http://www.ngs.noaa.gov/cgi-bin/spc_getpc.prl The formulae are correct in this program, and the results agree with the sign convention of Redfearn's formulae, as well as the normal usage of the grid convergence (converting between grid and true azimuths). I am not sure why the NGS program has the opposite sign, but I have asked NGS about it. Until this difference is resolved, be aware that the sign could be the opposite, and work out the correct sign from first principles.

Theory

Converting from cartesian co-ordinates (E, N or X, Y) to geographical co-ordinates (latitude and longitude) on a Transverse Mercator projection is a straightforward transformation, if somewhat long-winded. This program uses equations to help reduce the bulk of the program a little.

Given that we have a, e^2, ϕ, k_0, λ and λ_0, we can use the following expressions for the conversion. These are Redfearn's Formulae. Note that these use an extra term in the computations, compared to Snyder's book (1987), but this will make a negligible difference in the overall values. The results will be a little different to the tabulated values for SPCS, too, owing to the limitations on the SPCS 27 computations. Remembering that the allowable distortion in the SPCS was to be no more than 1 in 10,000, it is acceptable to drop the final term in the formulae, as this doesn't degrade the formulae by anywhere near 1 in 10,000. Such modified formulae will then agree with Snyder's formulae, remembering that Snyder was setting up the formulae for mapping, rather than geodetic use.

For UTM computations, you should use the full number of terms. This is because there is no 'legal' tolerance of distortion in the conversion process. UTM co-ordinates are now printed on 1:24,000 quadrangle maps, with either a grid/graticule or marginal ticks. These UTM co-ordinates are often on the NAD27 datum and need to be converted to NAD83 before they can be used. While there is a marginal note concerning the conversion of latitude and longitude from NAD27 to NAD83 on many of the more recent mapsheets, this value **does not** apply to the UTM co-ordinates (or the SPCS co-ordinates). This is because the latitude and longitude values are, in effect, figured from the origin in Kansas, while the UTM Northing co-ordinates are figured from the Equator. SPCS northings are figured from the zone origin, so will have a different shift for each zone. You should convert the co-ordinates to latitude and longitude for the appropriate system, convert these to NAD83, then convert to UTM or SPCS TM co-ordinates. An approximate set of shifts for UTM can be found in a paper by Welch, R., and Homsey, A., "Datum Shifts for UTM Co-ordinates," in the *Photogrammetric Engineering and Remote Sensing* journal, Volume 63, No. 4, pp. 371–375, published in 1997.

Conversion Formulae

Starting with the following general formulae, these can be applied in the following conversion formulae.

$E' = E - E_0$ (E_0 is the offset of the central meridian; check the value for each zone. For UTM, E_0 = 500 000·000 meters.)

$N' = N - N_0$ (N_0 is the offset of the origin latitude; check the value for each zone. For UTM in the northern hemisphere, N_0 = 0; for UTM in the southern hemisphere, N_0 = 10 000 000·000 meters.)

ν = radius of curvature in the prime vertical at ϕ; i.e. $\nu = \dfrac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$

$$\rho = \frac{a\,(1 - e^2)}{(1 - e^2 \sin^2 \phi)^{\frac{3}{2}}}$$ = radius of curvature in the meridian at ϕ

$$\omega = \lambda - \lambda_0$$

$\psi = \dfrac{\nu}{\rho}$ i.e. ratio of the radii of curvature at ϕ

$t = \tan\phi$

m = meridian distance from equator, computed using the following expression

$$m = a\,(A_0\,\phi - A_2 \sin 2\phi + A_4 \sin 4\phi - A_6 \sin 6\phi)$$

where ϕ is in radians and

$$A_0 = 1 - \frac{e^2}{4} - \frac{3e^4}{64} - \frac{5e^6}{256}$$

$$A_2 = \frac{3}{8}\left(e^2 + \frac{e^4}{4} + \frac{15e^6}{128}\right)$$

$$A_4 = \frac{15}{256}\left(e^4 + \frac{3e^6}{4}\right)$$

$$A_6 = \frac{35e^6}{3072}$$

With the appropriate values for ellipsoids and scale factors, these formulae will work for any Transverse Mercator projection: UTM, SPCS, AMG, MGA or whatever.

In order to convert the given northing to latitude, we first need to calculate what is known as the foot-point latitude, ϕ', which is the latitude for which the meridian distance is equal to $\frac{N'}{k_0}$. This value can be calculated directly provided three other values, namely n, G and σ are calculated first. The choice of variable names for these three values is historical and isn't related to any other use of them.

$$n = \frac{a - b}{a + b}$$ where a and b are the semi-major and semi-minor axes

$$b = a\sqrt{1 - e^2}$$

$$G = a\,(1 - n)\,(1 - n^2)\,\left(1 + \frac{9}{4}\,n^2 + \frac{225}{64}\,n^4\right)\frac{\pi}{180}$$

= mean length of an arc of one degree of the meridian

$$\sigma = \frac{m\,\pi}{180\,G} \qquad \text{use } m = \frac{N'}{k_0}$$

$$\phi' = \sigma + \left(\frac{3n}{2} - \frac{27n^3}{32}\right)\sin 2\sigma + \left(\frac{21n^2}{16} - \frac{55n^4}{32}\right)\sin 4\sigma + \left(\frac{151n^3}{96}\right)\sin 6\sigma + \left(\frac{1097n^4}{512}\right)\sin 8\sigma$$

Transverse Mercator Co-ordinates to Latitude/Longitude

With these values we can calculate the geographical co-ordinates directly. Note that t', ψ', ρ' and ν' are functions of the foot-point latitude and using the same formulae as listed above.

Latitude (in radians)

$$\text{Let } x = \frac{E'}{k_0\,\nu'}$$

$$\begin{aligned}\phi = \phi' &- \frac{t'}{k_0\,\rho'}\ x\ \frac{E'}{2} \\ &+ \frac{t'}{k_0\,\rho'}\ \frac{x^3\,E'}{24}\ [-4\,\psi'^2 + 9\,\psi'\,(1 - t'^2) + 12\,t'^2] \\ &- \frac{t'}{k_0\,\rho'}\ \frac{x^5\,E'}{720}\ [8\,\psi'^4\,(11 - 24\,t'^2) - 12\,\psi'^3\,(21 - 71\,t'^2) + 15\,\psi'^2\,(15 - 98\,t'^2 + 15\,t'^4) \\ &\qquad + 180\,\psi'\,(5\,t'^2 - 3\,t'^4) + 360\,t'^4] \\ &+ \frac{t'}{k_0\,\rho'}\ \frac{x^7\,E'}{40320}\,(1385 + 3633\,t'^2 + 4095\,t'^4 + 1575\,t'^6)\end{aligned}$$

Longitude (in radians)

$$\text{Let } x = \frac{E'}{k_0\,\nu'}$$

$$\begin{aligned}\omega = \sec\phi'\ x &\\ &- \sec\phi'\ \frac{x^3}{6}\ (\psi' + 2\,t'^2) \\ &+ \sec\phi'\ \frac{x^5}{120}\ [-4\,\psi'^3\,(1 - 6\,t'^2) + \psi'^2\,(9 - 68\,t'^2) + 72\,\psi'\,t'^2 + 24\,t'^4] \\ &- \sec\phi'\ \frac{x^7}{5040}\ (61 + 662\,t'^2 + 1320\,t'^4 + 720\,t'^6)\end{aligned}$$

Grid Convergence (in radians)

$$\text{Let } x = \frac{E'}{k_0\,\nu'}$$

$$\begin{aligned}\gamma = &- t'\,x \\ &+ t'\,\frac{x^3}{3}\ (-2\,\psi'^2 + 3\,\psi' + t'^2) \\ &- t'\,\frac{x^5}{15}\ [\psi'^4\,(11 - 24\,t'^2) - 3\,\psi'^3\,(8 - 23\,t'^2) + 5\,\psi'^2\,(3 - 14\,t'^2) + 30\,\psi'\,t'^2 + 3\,t'^4] \\ &+ t'\,\frac{x^7}{315}\ (17 + 77\,t'^2 + 105\,t'^4 + 45\,t'^6)\end{aligned}$$

Transverse Mercator Co-ordinates to Latitude/Longitude

Point Scale Factor

Let $x = \dfrac{E'^2}{k_0^2\, \nu'\, \rho'}$ (note the different value of x for this formula)

$$k = k_0 \left(1 + \frac{x}{2} + \frac{x^2}{24}\left(4\,\psi'\,(1 - 6\,t'^2) - 3\,(1 - 16\,t'^2) - \frac{24\,t'^2}{\psi'}\right) + \frac{x^3}{720} \right)$$

Given these values, the latitude and grid convergence are converted to degrees, minutes and seconds, while ω is converted to the longitude, λ, using the formula below, and then converted to degrees, minutes and seconds.

$$\lambda = \omega + \lambda_0$$

Sample Computations

Example 1

Using the SPCS 1983 (a = 6,378,137 m, e^2 = 0.006 694 3800), the following results are obtained.

Nevada East Zone, 2701, λ_0 = –115° 35', ϕ_0 = 34° 45', k_0 = 0.999 900, E_0 = 200,000.000 m, N_0 = 8,000,000.000 m.

Easting (E) = 185,603.123 m Northing (N) = 8,739,929.417 m

Latitude = 41° 25' 00".000 Longitude = –115° 45' 20".000

Grid Convergence (γ) = 0° 06' 50.1" Point Scale Factor (k) = 0.999 902 55

Example 2

Using the SPCS 1927 (a = 20925832.2 ft, e^2 = 0.006 768 66), the following results are obtained.

Nevada East Zone, SPCS 1927, λ_0 = –115° 35', ϕ_0 = 34° 45', k_0 = 0.999 900, E_0 = 500,000.000 ft, N_0 = 0.000 ft.

Easting (E) = 452,764.960 ft Northing (N) = 2,427,533.222 ft

Latitude = 41° 25' 00".000 Longitude = –115° 45' 20".000

Grid Convergence (γ) = 0° 06' 50.1" Point Scale Factor (k) = 0.999 902 55

Example 3

Using the ANS ellipsoid (a = 6,378,160 m, e^2 = 0.006 694 541 855), the following results are obtained.

AMG Zone 54, λ_0 = +141° 00', ϕ_0 = 0° 00', k_0 = 0.999 600,
E_0 = 500,000.000 m, N_0 = 10,000,000.000 m.

Easting (E) = 758,053.090 m Northing (N) = 5,828,496.973 m

Latitude = –37° 39' 15".557 Longitude = +143° 55' 30".6330

Grid Convergence (γ) = +1° 47' 16.67" Point Scale Factor (k) = 1.000 420 30

Example 4

Using the WGS-72 ellipsoid (a = 6,378,135 m, e^2 = 0.006 694 317 778), the following results are obtained.

UTM Zone 58, λ_0 = +165° 00', ϕ_0 = 0° 00', k_0 = 0.999 600,
E_0 = 500,000.000 m, N_0 = 10,000,000.000 m.

Easting (E) = 758,053.090 m Northing (N) = 5,828,496.973 m

Latitude = –37° 39' 15".557 Longitude = +143° 55' 30".6330

Grid Convergence (γ) = +1° 47' 16.67" Point Scale Factor (k) = 1.000 420 30

These last two sets of results agree with those computed in the AGD Technical Manual, 1986.

Running the Program

Press XEQ Y , then the ENTER key, to start the program. The calculator briefly displays TM 2 LAT—LONG, then briefly shows CHECK—ENTER A. This is "Point A," discussed below. The program then stops and displays the prompt for entering the semi-major axis value, while displaying the current default value:

A?
6,378,137.0000 (This is for GRS80/WGS84/NAD83)

If you are happy with this value for the semi-major axis of the ellipsoid, press R/S to continue. Otherwise. Key in a different value (for a different ellipsoid, e.g., 6378135 for WGS72) and press R/S to continue. (This discussion will use the data from Example 4, in the Sample Computations section, above.)

The calculator briefly displays CHECK—ENTER E. The program then stops and displays the prompt for entering the eccentricity of the ellipsoid, e:

E?
0.00669438 (This is for GRS80/WGS84/NAD83)

If this value for the eccentricity is correct, press R/S to continue. Otherwise, key in a different value (for a different ellipsoid, e.g., 0.006 694 317 778 for WGS-72) and press R/S to continue.

The calculator briefly displays CHECK—ENTER K. The program then stops and displays the prompt for entering the scale factor at the central meridian (λ_0), which is k_0:

K?
0.9996000 (This is for UTM)

If this value for the scale factor is satisfactory, press R/S to continue. If you want to change it, such as for an SPCS zone, key in the correct value and press R/S.

The calculator briefly displays CHK—NTR LONG 0. The program then stops and displays the prompt for entering the longitude of the central meridian of the projection, λ_0. Note that in the western hemisphere, this will be a negative value, and should be in HP notation (DDD.MMSS).

D?
—81.000000 (This is for UTM Zone 17)

If this is the correct central meridian, press R/S to continue, if this is not correct, key in the correct value, in HP notation, then press R/S to continue. In this case, key in 165 for UTM Zone 58, then press R/S.

The calculator briefly displays CHK—NTR LAT 0. The program then stops and displays the prompt for entering the latitude of the Northing co-ordinate origin, ϕ_0. For UTM, this is 0.000 (the equator), while for SPCS Zones, it is usually a latitude well south of the zone. The value should be entered in HP notation.

C?
0.000000 (This is for UTM)

If this is the correct latitude base, press R/S to continue. If you want a different value, key in that value and press R/S to continue. In this case, press R/S to continue.

The calculator briefly displays CHK—NTR E 0. The program then stops and prompts for entry of the false easting value, or the easting offset, denoted E_0. This is the value of the easting co-ordinate at the central meridian, λ_0. For UTM, this is 500,000.0000, while its value varies for different SPCS zones.

I?
500,000.0000 (This is for UTM)

If this is the correct value, press R/S to continue. If a different value is desired, key in the value and press R/S. In this case, press R/S to continue.

The calculator briefly displays CHK—NTR N 0. The program then stops and prompts for the false northing value, or the northing offset. This is the value of the northing at the point ϕ_0, λ_0, denoted N_0.

J?
0.0000 (This is for UTM in the northern hemisphere)

If this is the correct value, press R/S to continue. If a different value is required, key in the value and press R/S. In this case, key in 10,000,000.000 and press R/S. This is the N_0 value for UTM in the southern hemisphere.

This is "Point B," discussed below. The calculator briefly displays ENTER EASTING. The program stops and displays the prompt for entering the easting of the point to be converted.

X?
0.0000

Key in the easting of the point and press R/S to continue. In this case, key in 787420.487 and press R/S.

The calculator briefly displays ENTER NORTHING. The program then stops and displays the prompt for entering the northing of the point to be converted.

Y?
0.0000

Key in the northing of the point and press R/S to continue. In this case, key in 6782165.201 and press R/S.

Transverse Mercator Co-ordinates to Latitude/Longitude

The program displays RUNNING for some little time, then displays RESULTS briefly, followed by LATITUDE briefly. The program then stops and displays the latitude value of the point, in HP notation. In this case, the calculator displays:

F=
–29.03231530

This is the latitude of the point, in this case being 29° 03' 23".1530 S in more conventional notation. Press R/S to continue. The calculator briefly displays LONGITUDE, then stops and displays the longitude value of the point, in HP notation. In this case, the calculator displays:

L=
167.57066320

This is the longitude of the point, in this case being 167° 57' 06".6320 E in more conventional notation. Press R/S to continue. The calculator briefly displays GRID CONV, then stops and displays the grid convergence value in HP notation. In this case, the calculator displays:

G=
1.26045907

This is the grid convergence in HP notation, and is 1° 26' 04".59 in more conventional notation. Press R/S to continue. The calculator briefly displays PT SCALE FACT, then stops and displays the point scale factor of the point on the Transverse Mercator projection. In this case, the calculator displays:

S=
1.00061955

This is the point scale factor. Press R/S to continue.

You now have the choice of running one or more additional points. The calculator briefly displays NEXT PT [0—1], then stops and displays the prompt for answering questions:

Q?
0.0000

If you want to quit the program, just press R/S. If you want to enter more points, key in 1 and press R/S. In this case, the calculator then prompts to see if you want to use the same parameters. The calculator briefly displays NEW ZONE [0—1], then stops at the question prompt:

Q?
0.0000

If you want to go to a new zone, key in 1 and press R/S, and the calculator will take you to the point where you can change any of the values (Point A above), starting with the ellipsoid parameters. If you want to work in the same zone already entered, just press R/S, and the program will take you to “Point B” and prompt for the latitude of the point to be converted, and continue from there. You can go around the program as many times as necessary.

When you choose to end the program, the calculator briefly displays PROGRAM END and then comes to an end, returning to the point from which it was called, or to normal operations, and resetting Flag 10.

Transverse Mercator Co-ordinates to Latitude/Longitude

Storage Registers Used

A	Semi-major axis of the ellipsoid being used, a
B	Semi-minor axis of the ellipsoid being used, b
C	ϕ_0, the origin latitude for the co-ordinates
D	λ_0, the central meridian of the projection
E	Eccentricity of the ellipsoid, e^2
F	ϕ', foot-point latitude, then ϕ, latitude of the point that has been converted
G	γ, the grid convergence of the point being converted
H	meridian distance of the origin latitude, ϕ_0
I	E_0, the offset for the eastings (the easting at λ_0)
J	N_0, the offset for the northings (the northing at ϕ_0, λ_0)
K	k_0, the scale factor along the central meridian, λ_0
L	λ, longitude of the point that has been converted
M	m, preliminary meridian distance of the point to be converted
N	ν'
O	n, a constant for the ellipsoid
P	$\psi' = \frac{\nu'}{\rho'}$
Q	used for getting responses to questions about running more points
R	ρ
S	σ, an intermediate value; then k, point scale factor at the point being converted
T	$\tan \phi'$
U	$x = E' \div k_0 \nu'$
V	m_o, the meridian distance to ϕ_0
W	$\omega = \lambda - \lambda_0$
X	Easting co-ordinate of point to be converted
Y	Northing co-ordinate of point to be converted

Statistical Registers: not used

Labels Used

Label **Y**	Length = 1707	Checksum = 72F1

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computations to check proper operation after entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.

Parameters for the Computations

Universal Transverse Mercator (UTM)

For UTM, the ϕ_0 value is 0° (the equator) for both northern and southern hemispheres. The λ_0 values are given for each zone in the table below.

Zone	Central Meridian, λ_0	Zone	Central Meridian, λ_0
1	177° W	31	3°E
2	171° W	32	9°E
3	165° W	33	15°E
4	159° W	34	21°E
5	153° W	35	27°E
6	147° W	36	33°E
7	141° W	37	39°E
8	135° W	38	45°E
9	129° W	39	51°E
10	123° W	40	57°E
11	117° W	41	63°E
12	111° W	42	69°E
13	105° W	43	75°E
14	99° W	44	81°E
15	93° W	45	87°E
16	87° W	46	93°E
17	81° W	47	99°E
18	75° W	48	105°E
19	69° W	49	111°E
20	63° W	50	117°E
21	57° W	51	123°E
22	51° W	52	129°E
23	45° W	53	135°E
24	39° W	54	141°E
25	33° W	55	147°E
26	27° W	56	153°E
27	21° W	57	159°E
28	15° W	58	165°E
29	9° W	59	171°E
30	3° W	60	177°E

The E_0 value for all zones is 500,000.000 meters. The N_0 value for the northern hemisphere is 0.000 meters. The N_0 value for the southern hemisphere is 10,000,000.000 meters.

State Plane Co-ordinate System (SPCS) 1983

Several US states use the Transverse Mercator projection for SPCS 1983. The various parameters for each zone in the 1983 system are given in the table below. Use these parameters with the program, together with the GRS80/WGS84/NAD83 ellipsoid parameters, in meters.

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
Alabama					
East	85° 50'	30° 30'	0.9999600	200000.00	0.00
West	87° 30'	30° 00'	0.9999333	600000.00	0.00
Alaska					
2	142° 00'	54° 00'	0.9999000	500000.00	0.00
3	146° 00'	54° 00'	0.9999000	500000.00	0.00
4	150° 00'	54° 00'	0.9999000	500000.00	0.00
5	154° 00'	54° 00'	0.9999000	500000.00	0.00
6	185° 00'	54° 00'	0.9999000	500000.00	0.00
7	162° 00'	54° 00'	0.9999000	500000.00	0.00
8	166° 00'	54° 00'	0.9999000	500000.00	0.00
9	170° 00'	54° 00'	0.9999000	500000.00	0.00
Arizona					
East	110° 10'	31° 00'	0.9999000	213360.00	0.00
Central	111° 55'	31° 00'	0.9999000	213360.00	0.00
West	113° 45'	31° 00'	0.9999333	213360.00	0.00
Delaware	72° 25'	38° 00'	0.9999950	200000.00	0.00
Florida					
East	81° 00'	24° 20'	0.9999412	200000.00	0.00
West	82° 00'	24° 20'	0.9999412	200000.00	0.00
Georgia					
East	82° 10'	30° 00'	0.9999000	200000.00	0.00
West	84° 10'	30° 00'	0.9999000	700000.00	0.00
Hawaii					
1	155° 30'	18° 50'	0.9999667	500000.00	0.00
2	156° 40'	20° 20'	0.9999667	500000.00	0.00
3	158° 00'	21° 10'	0.9999900	500000.00	0.00
4	159° 30'	21° 50'	0.9999900	500000.00	0.00
5	160° 10'	21° 40'	1.0000000	500000.00	0.00

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
Idaho					
East	112° 10'	41° 40'	0.9999474	200000.00	0.00
Central	114° 00'	41° 40'	0.9999474	500000.00	0.00
Illinois					
East	88° 20'	36° 40'	0.9999750	300000.00	0.00
West	90° 10'	36° 40'	0.9999412	700000.00	0.00
Indiana					
East	85° 40'	37° 30'	0.9999667	100000.00	250000.00
West	87° 05'	37° 30'	0.9999667	900000.00	250000.00
Maine					
East	68° 30'	43° 40'	0.9999000	300000.00	0.00
West	70° 10'	42° 50'	0.9999667	900000.00	0.00
Mississippi					
East	88° 50'	29° 30'	0.9999500	300000.00	0.00
West	90° 20'	29° 30'	0.9999500	700000.00	0.00
Missouri					
East	90° 30'	35° 50'	0.9999333	250000.00	0.00
Central	92° 30'	35° 50'	0.9999333	500000.00	0.00
West	94° 30'	36° 10'	0.9999412	850000.00	0.00
Nevada					
East	115° 35'	34° 45'	0.9999000	200000.00	8000000.00
Central	116° 40'	34° 45'	0.9999000	500000.00	6000000.00
West	118° 35'	34° 45'	0.9999000	800000.00	4000000.00
New Hampshire	71° 40'	42° 30'	0.9999667	300000.00	0.00
New Jersey	74° 30'	38° 50'	0.9999000	150000.00	0.00
New Mexico					
East	104° 20'	31° 00'	0.9999091	165000.00	0.00
Central	106° 15'	31° 00'	0.9999000	500000.00	0.00
West	107° 50'	31° 00'	0.9999167	830000.00	0.00

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (m)	False Northing N_0 (m)
New York					
East	74° 30'	40° 00'	0.9999000	150000.00	0.00
Central	76° 35'	40° 00'	0.9999375	250000.00	0.00
West	78° 35'	40° 00'	0.9999375	350000.00	0.00
Rhode Island	71° 30'	41° 05'	0.9999938	100000.00	0.00
Vermont	72° 30'	42° 30'	0.9999643	500000.00	0.00
Wyoming					
East	105° 10'	40° 30'	0.9999375	200000.00	0.00
East Central	107° 20'	40° 30'	0.9999375	400000.00	100000.00
West Central	108° 45'	40° 30'	0.9999375	600000.00	0.00
West	110° 05'	40° 30'	0.9999375	800000.00	100000.00

State Plane Co-ordinate System (SPCS) 1927

Several US states used the Transverse Mercator projection for SPCS 1927. The various parameters for each zone in the 1927 system are given in the table below. Use these parameters with the program, together with the Clarke 1866 ellipsoid in feet.

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Alabama					
East	85° 50'	30° 30'	0.9999600	500000.00	0.00
West	87° 30'	30° 00'	0.9999333	500000.00	0.00
Alaska					
2	142° 00'	54° 00'	0.9999000	500000.00	0.00
3	146° 00'	54° 00'	0.9999000	500000.00	0.00
4	150° 00'	54° 00'	0.9999000	500000.00	0.00
5	154° 00'	54° 00'	0.9999000	500000.00	0.00
6	185° 00'	54° 00'	0.9999000	500000.00	0.00
7	162° 00'	54° 00'	0.9999000	700000.00	0.00
8	166° 00'	54° 00'	0.9999000	500000.00	0.00
9	170° 00'	54° 00'	0.9999000	600000.00	0.00

Transverse Mercator Co-ordinates to Latitude/Longitude

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Arizona					
East	110° 10'	31° 00'	0.9999000	500000.00	0.00
Central	111° 55'	31° 00'	0.9999000	500000.00	0.00
West	113° 45'	31° 00'	0.9999333	500000.00	0.00
Delaware	72° 25'	38° 00'	0.9999950	500000.00	0.00
Florida					
East	81° 00'	24° 20'	0.9999412	500000.00	0.00
West	82° 00'	24° 20'	0.9999412	500000.00	0.00
Georgia					
East	82° 10'	30° 00'	0.9999000	500000.00	0.00
West	84° 10'	30° 00'	0.9999000	500000.00	0.00
Hawaii					
1	155° 30'	18° 50'	0.9999667	500000.00	0.00
2	156° 40'	20° 20'	0.9999667	500000.00	0.00
3	158° 00'	21° 10'	0.9999900	500000.00	0.00
4	159° 30'	21° 50'	0.9999900	500000.00	0.00
5	160° 10'	21° 40'	1.0000000	500000.00	0.00
Idaho					
East	112° 10'	41° 40'	0.9999474	500000.00	0.00
Central	114° 00'	41° 40'	0.9999474	500000.00	0.00
West	115° 45'	41° 40'	0.9999333	500000.00	0.00
Illinois					
East	88° 20'	36° 40'	0.9999750	500000.00	0.00
West	90° 10'	36° 40'	0.9999412	500000.00	0.00
Indiana					
East	85° 40'	37° 30'	0.9999667	500000.00	0.00
West	87° 05'	37° 30'	0.9999667	500000.00	0.00
Maine					
East	68° 30'	43° 50'	0.9999000	500000.00	0.00
West	70° 10'	42° 50'	0.9999667	500000.00	0.00

Transverse Mercator Co-ordinates to Latitude/Longitude

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Michigan (old)					
East	83° 40'	41° 30'	0.9999429	500000.00	0.00
Central	85° 45'	41° 30'	0.9999091	500000.00	0.00
West	88° 45'	41° 30'	0.9999091	500000.00	0.00
Mississippi					
East	88° 50'	29° 40'	0.9999600	500000.00	0.00
West	90° 20'	30° 30'	0.9999412	500000.00	0.00
Missouri					
East	90° 30'	35° 50'	0.9999333	500000.00	0.00
Central	92° 30'	35° 50'	0.9999333	500000.00	0.00
West	94° 30'	36° 10'	0.9999412	500000.00	0.00
Nevada					
East	115° 35'	34° 45'	0.9999000	500000.00	0.00
Central	116° 40'	34° 45'	0.9999000	500000.00	0.00
West	118° 35'	34° 45'	0.9999000	500000.00	0.00
New Hampshire	71° 40'	42° 30'	0.9999667	500000.00	0.00
New Jersey	74° 40'	38° 50'	0.9999750	2000000.00	0.00
New Mexico					
East	104° 20'	31° 00'	0.9999091	500000.00	0.00
Central	106° 15'	31° 00'	0.9999000	500000.00	0.00
West	107° 50'	31° 00'	0.9999167	500000.00	0.00
New York					
East	74° 20'	40° 00'	0.9999667	500000.00	0.00
Central	76° 35'	40° 00'	0.9999375	500000.00	0.00
West	78° 35'	40° 00'	0.9999375	500000.00	0.00
Rhode Island	71° 30'	41° 05'	0.9999938	500000.00	0.00
Vermont	72° 30'	42° 30'	0.9999643	500000.00	0.00

Transverse Mercator Co-ordinates to Latitude/Longitude

	Central Meridian λ_o	Latitude Origin ϕ_0	Central Scale k_0	False Easting E_0 (ft)	False Northing N_0 (ft)
Wyoming					
East	105° 10'	40° 40'	0.9999412	500000.00	0.00
East Central	107° 20'	40° 40'	0.9999412	500000.00	0.00
West Central	108° 45'	40° 40'	0.9999412	500000.00	0.00
West	110° 05'	40° 40'	0.9999412	500000.00	0.00

Ellipsoids

There are a range of ellipsoids in common or former use. The table below has the a and e^2 values for a number of common (and less common) ellipsoids.

Ellipsoid	a Semi-major Axis	e^2 Eccentricity
GRS80–WGS94–NAD83	6378137 m	0.006 694 38
Clarke 1866 (NAD27)	6378206.4 m	0.006 768 66
Clarke 1866 (NAD27)	20925832.2 ft	0.006 768 66
ANS (Australian)	6378160 m	0.006 694 541 855
Airy 1830	6377563.4 m	0.006 670 54
Bessel 1841	6377397.16 m	0.006 674 372
Clarke 1880	6378249.15 m	0.006 803 511
Everest 1830	6377276.35 m	0.006 637 847
Fischer 1960 (Mercury)	6378166 m	0.006 693 422
Fischer 1968	6378150 m	0.006 693 422
Hough 1956	6378270 m	0.006 722 67
International	6378388 m	0.006 722 67
Krassovsky 1940	6378245 m	0.006 693 422
South American 1960	6378160 m	0.006 694 542
GRS 1967	6378160 m	0.006 694 605
GRS 1975	6378140 m	0.006 694 385
WGS 60	6378165 m	0.006 693 422
WGS 66	6378145 m	0.006 694 542
WGS 72	6378135 m	0.006 694 317 778
WGS 84	6378137 m	0.006 694 38

Reference

SNYDER, J.P., 1987. *Map Projections—A Working Manual*. U.S. Geological Survey Professional Paper 1395. Washington: US Government Printing Office.

Plane Triangle Solutions

Programmer: Dr. Bill Hazelton

Date: November, 2007.

Line	Instruction	Display	Program Entry Instructions
T001	LBL T		↱ **LBL** T
T002	CLSTK		↱ **CLEAR** 5
T003	FS? 10		↰ **FLAGS** SF? . 0
T004	GTO T008		
T005	SF 1		↰ **FLAGS** SF 1
T006	SF 10		↰ **FLAGS** SF . 0
T007	GTO T009		
T008	CF 1		↰ **FLAGS** CF 1
T009	TRIANGLE SOLNS		EQN RCL T, RCL R, etc., ENTER to end
T010	PSE		
T011	0		
T012	STO T		
T013	3 SIDES		EQN 3 ↱ <SPACE> RCL S etc.
T014	PSE		
T015	INPUT T		
T016	RCL T		
T017	*x* > 0?		↱ ***x*?0** 4
T018	GTO T044		
T019	2 SIDES INC θ		EQN 2 ↱ <SPACE> RCL S etc.
T020	PSE		
T021	INPUT T		
T022	RCL T		
T023	*x* > 0?		↱ ***x*?0** 4
T024	GTO T120		
T025	2 SIDE N–INC θ		EQN 2 ↱ <SPACE> RCL S, etc.
T026	PSE		
T027	INPUT T		
T028	RCL T		
T029	*x* > 0?		↱ ***x*?0** 4
T030	GTO T286		
T031	2 θ INC SIDE		EQN 2 ↱ <SPACE> ↱ **θ** RCL I, etc.
T032	PSE		
T033	INPUT T		
T034	RCL T		
T035	*x* > 0?		↱ ***x*?0** 4
T036	GTO T180		
T037	2 θ N–INC SIDE		EQN 2 ↱ <SPACE> ↱ **θ** RCL I, etc.
T038	PSE		
T039	INPUT T		

Plane Triangle Solutions

Line	Instruction
T040	RCL T
T041	$x > 0$?
T042	GTO T233
T043	GTO T011
T044	ENTER SIDE 1
T045	PSE
T046	INPUT S
T047	RCL S
T048	STO A
T049	ENTER SIDE 2
T050	PSE
T051	INPUT S
T052	RCL S
T053	STO B
T054	ENTER SIDE 3
T055	PSE
T056	INPUT S
T057	RCL S
T058	STO C
T059	x^2
T060	RCL B
T061	x^2
T062	+
T063	RCL A
T064	x^2
T065	–
T066	RCL÷ B
T067	RCL÷ C
T068	2
T069	÷
T070	ACOS
T071	→HMS
T072	ANGLE 1 =
T073	PSE
T074	STOP
T075	RCL C
T076	x^2
T077	RCL A
T078	x^2
T079	+
T080	RCL B
T081	x^2
T082	–
T083	RCL÷ C
T084	RCL÷ A

Line	Instruction
T085	2
T086	÷
T087	ACOS
T088	→HMS
T089	ANGLE 2 =
T090	PSE
T091	STOP
T092	RCL B
T093	x^2
T094	RCL A
T095	x^2
T096	+
T097	RCL C
T098	x^2
T099	–
T100	RCL÷ A
T101	RCL÷ B
T102	2
T103	÷
T104	ACOS
T105	STO D
T106	→HMS
T107	ANGLE 3 =
T108	PSE
T109	STOP
T110	RCL D
T111	SIN
T112	2
T113	÷
T114	RCL× A
T115	RCL× B
T116	AREA =
T117	PSE
T118	STOP
T119	GTO T387
T120	ENTER SIDE 1
T121	PSE
T122	INPUT S
T123	RCL S
T124	STO A
T125	ENTER SIDE 2
T126	PSE
T127	INPUT S
T128	RCL S
T129	STO B

Line	Instruction
T130	ENTER ANGLE 3
T131	PSE
T132	INPUT Q
T133	RCL Q
T134	HMS→
T135	STO C
T136	COS
T137	RCL× A
T138	RCL× B
T139	2
T140	×
T141	RCL A
T142	x^2
T143	x < > y
T144	–
T145	RCL B
T146	x^2
T147	+
T148	√x
T149	STO D
T150	SIDE 3 =
T151	PSE
T152	VIEW D
T153	RCL C
T154	SIN
T155	RCL÷ D
T156	STO D
T157	RCL× A
T158	ASIN
T159	→HMS
T160	ANGLE 1 =
T161	PSE
T162	STOP
T163	RCL D
T164	RCL× B
T165	ASIN
T166	→HMS
T167	ANGLE 2 =
T168	PSE
T169	STOP
T170	RCL C
T171	SIN
T172	RCL× A
T173	RCL× B
T174	2

Plane Triangle Solutions

Line	Instruction
T175	÷
T176	AREA =
T177	PSE
T178	STOP
T179	GTO T387
T180	ENTER ANGLE 1
T181	PSE
T182	INPUT Q
T183	RCL Q
T184	HMS→
T185	STO A
T186	ENTER ANGLE 2
T187	PSE
T188	INPUT Q
T189	RCL Q
T190	HMS→
T191	STO B
T192	ENTER SIDE 3
T193	PSE
T194	INPUT S
T195	RCL S
T196	STO C
T197	180
T198	RCL– A
T199	RCL– B
T200	STO D
T201	→HMS
T202	ANGLE 3 =
T203	PSE
T204	STOP
T205	RCL C
T206	RCL D
T207	SIN
T208	STO E
T209	÷
T210	STO F
T211	RCL A
T212	SIN
T213	×
T214	STO× E
T215	SIDE 1 =
T216	PSE
T217	STOP
T218	RCL F
T219	RCL B

Line	Instruction
T220	SIN
T221	×
T222	STO× E
T223	SIDE 2 =
T224	PSE
T225	STOP
T226	RCL E
T227	2
T228	÷
T229	AREA =
T230	PSE
T231	STOP
T232	GTO T387
T233	ENTER ANGLE 1
T234	PSE
T235	INPUT Q
T236	RCL Q
T237	HMS→
T238	STO A
T239	ENTER ANGLE 3
T240	PSE
T241	INPUT Q
T242	RCL Q
T243	HMS→
T244	STO B
T245	ENTER SIDE 3
T246	PSE
T247	INPUT S
T248	RCL S
T249	STO C
T250	RCL B
T251	SIN
T252	÷
T253	STO D
T254	RCL A
T255	SIN
T256	×
T257	SIDE 1 =
T258	PSE
T259	STOP
T260	180
T261	RCL– A
T262	RCL– B
T263	STO E
T264	→HMS

Line	Instruction
T265	ANGLE 2 =
T266	PSE
T267	STOP
T268	RCL E
T269	SIN
T270	RCL× D
T271	STO E
T272	SIDE 2 =
T273	PSE
T274	STOP
T275	RCL E
T276	RCL× C
T277	RCL A
T278	SIN
T279	2
T280	÷
T281	×
T282	AREA =
T283	PSE
T284	STOP
T285	GTO T387
T286	ENTER SIDE 1
T287	PSE
T288	INPUT S
T289	RCL S
T290	STO A
T291	ENTER SIDE 2
T292	PSE
T293	INPUT S
T294	RCL S
T295	STO B
T296	ENTER ANGLE 1
T297	PSE
T298	INPUT Q
T299	RCL Q
T300	HMS→
T301	STO C
T302	SIN
T303	RCL÷ A
T304	STO D
T305	RCL× B
T306	ASIN
T307	STO E
T308	180
T309	x < > y

Plane Triangle Solutions

T310	–
T311	RCL– C
T312	STO F
T313	SIN
T314	RCL× A
T315	RCL C
T316	SIN
T317	÷
T318	STO G
T319	RCL F
T320	SIN
T321	RCL× A
T322	RCL× B
T323	2
T324	÷
T325	STO H
T326	SOLUTION 1
T327	PSE
T328	ANGLE 2 =
T329	PSE
T330	RCL E
T331	→HMS
T332	STOP
T333	ANGLE 3 =
T334	PSE
T335	RCL F
T336	→HMS
T337	STOP
T338	SIDE 3 =
T339	PSE
T340	RCL G
T341	STOP
T342	AREA =
T343	PSE
T344	RCL H
T345	STOP
T346	180
T347	RCL– E
T348	STO E
T349	180
T350	x < > y
T351	–
T352	RCL– C
T353	STO F
T354	SIN
T355	RCL× A

T356	RCL C
T357	SIN
T358	÷
T359	STO G
T360	RCL F
T361	SIN
T362	RCL× A
T363	RCL× B
T364	2
T365	÷
T366	STO H
T367	SOLUTION 2
T368	PSE
T369	ANGLE 2 =
T370	PSE
T371	RCL E
T372	→HMS
T373	STOP
T374	ANGLE 3 =
T375	PSE
T376	RCL F
T377	→HMS
T378	STOP
T379	SIDE 3 =
T380	PSE
T381	RCL G
T382	STOP
T383	AREA =
T384	PSE
T385	RCL H
T386	STOP
T387	FS? 1
T388	CF 10
T389	RTN

Plane Triangle Solutions

Notes

(1) Program for solving a plane triangle's unspecified angles, side lengths and area, given three inputs (angles and side lengths) that include at least one side length.

(2) Angles are entered and displayed in HP notation, i.e., DDD.MMSS.

(3) Whatever linear units are used (and they should be the same for all three sides, of course), the area will be presented in those units squared. That is, if the lengths are in feet, the area is in square feet; if the lengths are in meters, the area is in square meters; if the lengths are in cubits, the area is in square cubits.

(4) The purpose of the EQN entries in the program is to provide a prompt ahead of input or output. The program sets flag 10 to display rather than evaluate equations. Letters of the message must be entered with the RCL key, i.e., to enter HI, press RCL H then RCL I. Spaces can be entered with the SPACE key combination. Digits can be keyed in directly.

(5) The two angles and included side problem is essentially a surveying 'intersection' problem, and can also be interpreted as the 'two missing sides' problem.

(6) In the '2 sides and an angle not between them' problem, there are two possible solutions. Each solution is presented separately. Note that if one of the solutions is not physically possible, the program may return an error.

(7) The program has all the possible solutions coded within it, so the user is first presented with a set of choices, one at a time, to select the appropriate solution. When a selection is made, the calculator then moves to the correct part of the program and proceeds with a stand-alone calculation. At the end of the operation, the program returns Flag 10 to its original setting, before returning to the calling point.

Label Used

Label **T** Length = 1664 Checksum = EB95

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry. Length and checksum values are based on single spaces between words, numbers and equal signs in prompts.

Storage Registers Used

A Input 1

B Input 2

C Input 3

D Intermediate result storage

E Intermediate result storage

F Intermediate result storage

G Intermediate result storage

Plane Triangle Solutions

H Intermediate result storage

Q Temporary storage for entered angles

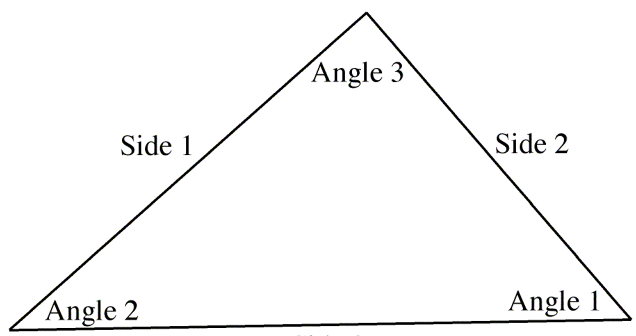
S Temporary storage of entered side length

T Temporary storage of choice value for initial selection of sub-program.

Theory

This program accepts the three components of a triangle, then uses the sine rule, cosine rule and angle sum conditions to compute the three remain components. The area is computed using the length of two sides and half the sine of the angle between them.

The numbering scheme for the sides and angles is as shown in the diagram. Sides are opposite the angle with the same number.



The resulting angles are presented in HP notation. Plane surveying assumptions apply. The program uses no error checking on entered data or results. It is a good move to check that the angles all sum to 180°.

Running the Program

Press XEQ T and press ENTER.

The calculator displays TRIANGLE SOLNS briefly.

The calculator then moves into a large loop to determine the particular type of triangle that needs to be solved. It displays the options for solution in turn, and offers a prompt for selection. This loop continues until a choice is made.

A. The calculator displays 3 SIDES briefly.

If the three side lengths of the triangle are known, then this is the solution required. If not, it isn't.

The calculator displays a prompt and default value, and waits for input:

T?
0.00000

If the three known sides solution is required, key in 1 and press R/S. If not, key in zero and press R/S, or just press R/S, as zero is the default.

If 1 was entered, the program jumps to the 3 SIDES solution, discussed below at **B**.

Plane Triangle Solutions

If a zero was entered, or just R/S pressed, the calculator briefly displays:

2 SIDES INC θ

then the prompt for input:

T?
0.00000

This is the point for the solution where two side lengths are known, together with the angle between the two known sides. If this is desired, key in 1 and press R/S. if not, just press R/S to go to the next choice.

If 1 was entered, the program jumps to the 2 SIDES INC θ solution, discussed below at **C**.

If a zero was entered, or just R/S pressed, the calculator briefly displays:

2 SIDES N–INC θ

then the prompt for input:

T?
0.00000

This is the point for the solution where two side lengths are known, together with an angle that is not between the two known sides. If this is desired, key in 1 and press R/S. if not, just press R/S to go to the next choice.

If 1 was entered, the program jumps to the 2 SIDES N–INC θ solution, discussed below at **D**.

If a zero was entered, or just R/S pressed, the calculator briefly displays:

2 θ INC SIDE

then the prompt for input:

T?
0.00000

This is the point for the solution where two angles are known, together with the side length between the two angles. If this is desired, key in 1 and press R/S. if not, just press R/S to go to the next choice.

If 1 was entered, the program jumps to the 2 θ INC SIDE solution, discussed below at **E**.

If a zero was entered, or just R/S pressed, the calculator briefly displays:

2 θ N–INC SIDE

then the prompt for input:

T?
0.00000

This is the point for the solution where two angles are known, together with a side length that is not directly between the two angles. If this is desired, key in 1 and press R/S, if not, just press R/S to go to the next choice.

Plane Triangle Solutions

If 1 was entered, the program jumps to the 2 θ N–INC SIDE solution, discussed below at **F**.

If a zero was entered, or just R/S pressed, the calculator returns to the first option, 3 SIDES, as it has run through all of the possibilities with plain triangles. So the calculator returns to point **A**, above.

B. 3 SIDES solution

If the 3 SIDES solution was chosen the calculator jumps to this point, and the proceeds as follows.

Screen shows ENTER SIDE 1 briefly, then prompts with S?

Enter the length of side 1 and press R/S.

Screen shows ENTER SIDE 2 briefly, then prompts with S?

Enter the length of side 2 and press R/S.

Screen shows ENTER SIDE 3 briefly, then prompts with S?

Enter the length of side 3 and press R/S.

Screen shows ANGLE 1 = briefly, then shows Angle 1 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 2 = briefly, then shows Angle 2 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 3 = briefly, then shows Angle 3 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to end program. This sets Flag 10 to its original value, as it was set at the start of the program.

Sample Computations

	Triangle 1	Triangle 2
Inputs:	Side Length 1 = 100.000	Side Length 1 = 10.000
	Side Length 2 = 100.000	Side Length 2 = 10.000
	Side Length 3 = 100.000	Side Length 3 = 18.000
Results:	Angle 1 = 60° 00' 00"	Angle 1 = 25° 50' 31"
	Angle 2 = 60° 00' 00"	Angle 2 = 25° 50' 31"
	Angle 3 = 60° 00' 00"	Angle 3 = 128° 18' 58"
	Area = 4,330.127	Area = 39.230
	Check angle sum = 180° 00' 00"	Check = 180° 00' 00"

Plane Triangle Solutions

C. 2 SIDES INC θ solution

If the 2 SIDES INC θ solution was chosen, the program jumps to this point, and proceeds as follows.

Screen shows ENTER SIDE 1 briefly, then prompts with S?

Enter the length of side 1 and press R/S.

Screen shows ENTER SIDE 2 briefly, then prompts with S?

Enter the length of side 2 and press R/S.

Screen shows ENTER ANGLE 3 briefly, then prompts with Q?

Enter angle 3 in HP notation and press R/S.

Screen shows SIDE 1 = briefly, then shows Side 1 in the lower (X) register, with D= above.

Press R/S to continue.

Screen shows ANGLE 1 = briefly, then shows Angle 1 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 2 = briefly, then shows Angle 2 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to end program. This resets Flag 10 to its previous value, as it was set at the start of the program.

Sample Computations

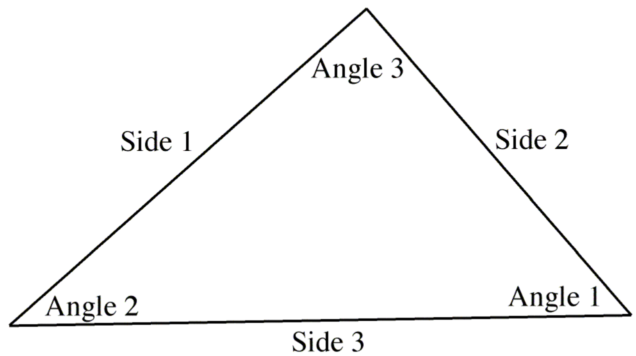
	Triangle 1	Triangle 2
Inputs:	Side Length 1 = 100.000	Side Length 1 = 10.000
	Side Length 2 = 100.000	Side Length 2 = 10.000
	Angle 3 = 60° 00' 00"	Angle 3 = 128° 18' 58"
Results:	Side 3 = 100.000	Side 3 = 18.000
	Angle 1 = 60° 00' 00"	Angle 1 = 25° 50' 31"
	Angle 2 = 60° 00' 00"	Angle 2 = 25° 50' 31"
	Area = 4,330.127	Area = 39.230
	Check angle sum = 180° 00' 00"	Check = 180° 00' 00"

Plane Triangle Solutions

D. 2 SIDES N–INC θ Solution

This part of the program accepts two side lengths of the triangle and an angle not between them, then uses the sine rule to compute one of the other angles. The third angle is computed by subtracting the sum of the other angles from 180°. The remaining side is computed using the sine rule. The area is computed using the length of two sides and half the sine of the angle between them.

The numbering scheme for the sides and angles is as shown in the diagram. Sides are opposite the angle with the same number. In this case, Side 1 and Side 2 are known, along with Angle 1.



There are two possible solutions, depending upon the solutions to Angle 2. Because it is determined using the sine rule, and arcsin(x) can have two values, there is one solution where Angle 2 lies between 0° and 90°, and a second where Angle 2 lies between 90° and 180°. Both these solutions are computed. The results are presented in two groups, with suitable text prompts.

Solution 1 is based on Angle 2 being less than 90°. Solution 2 is based on Angle 2 being greater than 90°.

When the 2 SIDES N–INC θ solution is chosen, the calculator jumps to this point, and proceeds as follows.

Screen shows ENTER SIDE 1 briefly, then prompts with S?

Enter the length of side 1 and press R/S.

Screen shows ENTER SIDE 2 briefly, then prompts with S?

Enter the length of side 2 and press R/S.

Screen shows ENTER ANGLE 1 briefly, then prompts with Q?

Enter angle 1 in HP notation and press R/S.

Screen shows SOLUTION 1 briefly.

Screen shows ANGLE 2 = briefly, then shows Angle 2 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 3 = briefly, then shows Angle 3 in HP notation in the lower (X) register.

Press R/S to continue.

Plane Triangle Solutions

Screen shows SIDE 3 = briefly, then shows Side 3 in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to continue.

Screen shows SOLUTION 2 briefly.

Screen shows ANGLE 2 = briefly, then shows Angle 2 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 3 = briefly, then shows Angle 3 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows SIDE 3 = briefly, then shows Side 3 in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to end program. This resets Flag 10 to its original value, as it was set at the start of the program.

Sample Computations

	Triangle 1	Triangle 2
Inputs:	Side Length 1 = 100.000	Side Length 1 = 10.000
	Side Length 2 = 100.000	Side Length 2 = 10.000
	Angle 1 = 60° 00' 00"	Angle 1 = 25° 50' 31"
Results:	**Solution 1**	
	Angle 2 = 60° 00' 00"	Angle 2 = 25° 50' 31"
	Angle 3 = 60° 00' 00"	Angle 3 = 128° 18' 58"
	Side Length 3 = 100.000	Side Length 3 = 18.000
	Area = 4,330.127	Area = 39.230
	[Check angle sum = 180° 00' 00"	Check = 180° 00' 00"]
	Solution 2	
	Angle 2 = 120° 00' 00"	Angle 2 = 154° 09' 29"
	Angle 3 = 0° 00' 00"	Angle 3 = 0° 00' 00" (very small)
	Side Length 3 = 0.000	Side Length 3 = 0.000
	Area = 0.000	Area = 0.000
	Check angle sum = 180° 00' 00"	Check = 180° 00' 00"

Plane Triangle Solutions

Clearly, neither of the results for Solution 2 are particularly good solutions (despite being mathematically valid and correct), because of the zero angles involved. This, as well as negative angles, is one way to spot an unsuitable solution. However, it is possible to have two perfectly reasonable solutions, in which case you will need to look beyond the given data to decide which is the required solution.

E. 2 θ INC SIDE Solution

When the 2 θ INC SIDE solution is chosen, the calculator jumps to this point, and proceeds as follows.

Screen shows ENTER ANGLE 1 briefly, then prompts with Q?

Enter angle 1 in HP notation and press R/S.

Screen shows ENTER ANGLE 3 briefly, then prompts with Q?

Enter angle 3 in HP notation and press R/S.

Screen shows ENTER SIDE 3 briefly, then prompts with S?

Enter the length of side 3 and press R/S.

Screen shows SIDE 1 = briefly, then shows Side 1 in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 1 = briefly, then shows Angle 1 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows SIDE 2 = briefly, then shows Side 2 in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to end program. This sets Flag 10 to its original value, as it was set at the start of the program.

Sample Computations

	Triangle 1	Triangle 2
Inputs:	Angle 1 = 60° 00' 00"	Angle 1 = 25° 50' 31"
	Angle 2 = 60° 00' 00"	Angle 2 = 25° 50' 31"
	Side Length 3 = 100.000	Side Length 3 = 18.000
Results:	Angle 3 = 60° 00' 00"	Angle 3 = 128° 18' 58"
	Side 1 = 100.000	Side 1 = 10.000
	Side 2 = 100.000	Side 2 = 10.000
	Area = 4,330.127	Area = 39.230
	Check angle sum = 180° 00' 00"	Check = 180° 00' 00"

Plane Triangle Solutions

F. 2 θ N–INC SIDE Solution

When the 2 θ N–INC SIDE solution is chosen, the calculator jumps to this point, and proceeds as follows.

Screen shows ENTER ANGLE 1 briefly, then prompts with Q?

Enter angle 1 in HP notation and press R/S.

Screen shows ENTER ANGLE 3 briefly, then prompts with Q?

Enter angle 3 in HP notation and press R/S.

Screen shows ENTER SIDE 3 briefly, then prompts with S?

Enter the length of side 3 and press R/S.

Screen shows SIDE 1 = briefly, then shows Side 1 in the lower (X) register.

Press R/S to continue.

Screen shows ANGLE 1 = briefly, then shows Angle 1 in HP notation in the lower (X) register.

Press R/S to continue.

Screen shows SIDE 2 = briefly, then shows Side 2 in the lower (X) register.

Press R/S to continue.

Screen shows AREA = briefly, then shows the area in the lower (X) register.

Press R/S to end program. This resets Flag 10 to its original value, as it was set at the start of the program.

Sample Computations

	Triangle 1	Triangle 2
Inputs:	Angle 1 = 60° 00' 00"	Angle 1 = 25° 50' 31"
	Angle 3 = 60° 00' 00"	Angle 3 = 128° 18' 58"
	Side Length 3 = 100.000	Side Length 3 = 18.000
Results:	Side 1 = 100.000	Side 1 = 10.000
	Angle 2 = 60° 00' 00"	Angle 2 = 25° 50' 31"
	Side 2 = 100.000	Side 2 = 10.000
	Area = 4,330.127	Area = 39.230
	Check angle sum = 180° 00' 00"	Check = 180° 00' 00"

HMS+

Programmer: Dr. Bill Hazelton

Date: October, 2007.

Line	Instruction	Display	User Instructions
C001	LBL C		Enter first angle. Press ENTER.
C002	HMS→		Enter second angle. Press XEQ C ENTER.
C003	x < > y		
C004	HMS→		(Angles must be in DDD.MMSS format)
C005	+		
C006	→HMS		
C007	RTN		Angle sum displayed (in HP notation)

Notes

(1) General program to add two angles, azimuths or directions in DDD.MMSS format (HP notation), and produce a result in the same format.

(2) Key in the first angle. Press ENTER. Key in the second angle. Stack will contain:

Stack Register	Contents
T	
Z	
Y	First angle in D.MS (line 1)
X	Second angle in D.MS (line 2)

Press XEQ C ENTER. The sum of the two angles in HP notation will be in the X register (line 1)

(3) Negative values will work correctly.

Sample Computation

123° 45' 56" + 321° 54' 32" = 445° 40' 28"

Storage Registers Used

None

Labels Used

Label **C** Length = 21 Checksum = F341

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

HMS–

Programmer: Dr. Bill Hazelton

Date: October, 2007.

Line	Instruction	Display	User Instructions
D001	LBL D		Enter first angle. Press ENTER.
D002	HMS→		Enter second angle. Press XEQ D ENTER.
D003	x < > y		
D004	HMS→		(Angles in DDD.MMSS format)
D005	x < > y		
D006	–		
D007	→HMS		
D008	RTN		Angle sum displayed (in HP notation)

Notes

(1) General program to get the difference between two angles, azimuths or directions in DDD.MMSS format (HP notation), and produce a result in the same format.

(2) Key in the first angle. Press ENTER. Key in the second angle. Stack will contain:

Stack Register	Contents
T	
Z	
Y	First angle in DMS (line 1)
X	Second angle in DMS (line 2)

Press XEQ D ENTER. The difference between the two angles in HP notation will be in the X register. The second angle will be subtracted from the first.

(3) Negative values will work correctly.

Sample Computation

321° 54' 32" – 123° 45' 56" = 198° 08' 36"

Storage Registers Used

None

Labels Used

Label **D** Length = 24 Checksum = E0E0

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

HP-35s Calculator Program Utility 3

Enter Vector with D.MMSS Azimuth for Complex Number

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Mnemonic**: V for **V**ector Building

Line	Instruction	Display	User Instructions
V001	LBL V		Press XEQ V ENTER.
V002	10		
V003	STO I		
V004	RCL A		
V005	STO (I)		STO (I) ((I) is on the zero key)
V006	1		
V007	STO + I		
V008	RCL D		
V009	STO (I)		
V010	1		
V011	STO + I		
V012	INPUT A	A?	Prompts for azimuth in D.MMSS
V013	RCL A		
V014	HMS→		
V015	STO A		
V016	INPUT D	D?	Prompts for distance (D?)
V017	RCL A		
V018	COS		
V019	RCL × D		
V020	RCL A		
V021	SIN		
V022	RCL × D		
V023	0 **i** 1		[Key in as 0, then **i**, then 1, press ENTER]
V024	×		
V025	+		
V026	STO (I)		
V027	10		
V028	STO I		
V029	RCL (I)		
V030	STO A		
V031	1		
V032	STO + I		
V033	RCL (I)		
V034	STO D		
V035	1		
V036	STO + I		
V037	CLSTK		
V038	RCL (I)		
V039	RTN		Complex number now in stack in X

Notes

(1) A program that allows the user to enter a vector as two separate components, azimuth and distance, with the azimuth in D.MMSS (HP format), and have it converted to a complex number, with the azimuth component in decimal degrees.

(2) Calculator should be set in DEGREES mode. Press MODE, then 1.

(3) Because the calculator uses data entry into the A and D storage registers to allow simple prompting, there is the potential of this program deleting any data already in those storage registers. This could be a problem if this program was called as a sub-routine from within another program. To avoid this problem, the program copies the contents of storage register A to storage register 10, and the contents of storage register D to storage register 11. At the conclusion of the program, the values are copied back into storage registers A and D.

(4) Because the program replaces everything that was on the stack before it ran, the stack is cleared of all data before the result of the calculation is returned to the stack.

(5) While copying back the contents of registers 10 and 11, the program stores the result of the calculation in storage register 12.

(6) If using a program that requires storage registers 10, 11 and 12, change the value of 10 in lines V002 and V027 to a suitable number, so that the set of three storage registers selected aren't used elsewhere.

Operation

Press XEQ V ENTER.

The calculator prompts with A? to enter the azimuth in degrees, minutes and seconds (HP notation). Key in the azimuth and press R/S.

The calculator prompts with D? to enter the distance. Key in the distance and press R/S.

The calculator displays the complex number representing the vector in the X register, or line 2 of the display. The remainder of the stack is zeros.

Examples

1. 63° 15' 47" and 105.528 will give a complex number of 105.5280 θ 63.2631 in polar mode (with FIX 4 set for the display), or 47.4765 **i** 94.2451 in rectangular mode.

2. 128° 15' 47" and 105.528 will give a complex number of 105.5280 θ 128.2631, or –65.3506 **i** 82.8580.

3. 237° 15' 47" and 105.528 will give a complex number of 105.5280 θ –122.7369, or –57.0677 **i** –88.7662.

4. 333° 15' 47" and 105.528 will give a complex number of 105.5280 θ –26.7369, or 94.2451 **i** –47.4765.

5. 397° 15' 47" and 105.528 produce 105.5280 θ 37.2631, or 83.9859 **i** 63.8946.

Storage Registers Used

A	Stores the azimuth value, initially in D.MMSS, then in decimal degrees.
D	Stores the distance value.
I	Used to store the value for indirect addressing of registers 10, 11 and 12.
10	Temporary storage for the contents of storage register A while the program runs.
11	Temporary storage for the contents of storage register D while the program runs.
12	Temporary storage for the answer while copying back A and D.

Labels Used

Label **V**	Length = 128	Checksum = 39FE

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

HP-35s Calculator Program — Utility 4

Extract Real and Imaginary Parts of a Complex Number

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Mnemonic**: X for e**X**tract Co-ordinates

Line	Instruction	Display	User Instructions
X001	LBL X		Press XEQ X ENTER.
X002	10		
X003	STO I		
X004	R↓		
X005	STO (I)		STO (I) ((I) is on the zero key)
X006	ARG		
X007	COS		
X008	RCL (I)		
X009	ABS		
X010	×		
X011	RCL (I)		
X012	ARG		
X013	SIN		
X014	RCL (I)		
X015	ABS		
X016	×		
X017	RTN		Components now on stack

Notes

(1) The program is designed to operate with the complex number on the stack in the X register (line 2 of the display).

(2) The program stores the vector in storage register 10 for quick retrieval during operation. If this storage register is in use, change the value in line X002 to a suitable number that doesn't clash with other needs.

(3) The program will work regardless of the display mode for complex numbers, and regardless of the angle unit mode of the calculator.

(4) The program will over-write values in the stack prior to it being called.

Operation

(1) Put the complex number into the X register on the stack, either by keying it in, or by recalling it there from wherever it is stored.

(2) Press XEQ X ENTER. The program returns the real (or Northing or Y) component or co-ordinate to the Y register (line 1 of the display), and the imaginary part or co-ordinate (or Easting or X) to the X register (line 2 of the display).

Examples

1. The complex number 123.0000 **i** 456.0000 is on the stack. The result is 123.0000 in the Y register and 456.000 in the X register.

2. The complex number 100.0000 θ 45.0000 is on the stack. The result is 70.7107 in the Y register and 70.7107 in the X register.

3. The complex number 100.0000 θ 230.0000 is on the stack. The result is –64.2788 in the Y register and –76.6044 in the X register.

Storage Registers Used

I Used for the indirect addressing of storage register 10.

10 Used to store the complex number during operations.

Labels Used

Label **X** Length = 53 Checksum = C46D

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

HP-35s Calculator Program   Utility 5

Enter Vector with D.MMSS Azimuth for Complex Number, but check for a negative distance

Programmer: Dr. Bill Hazelton

Date: December, 2007.   **Mnemonic**: **W** for Vector **W**rangling

Line	Instruction	Display	User Instructions
W001	LBL W		Press XEQ W ENTER.
W002	10		
W003	STO I		
W004	RCL A		
W005	STO (I)		STO (I) ((I) is on the zero key)
W006	1		
W007	STO + I		
W008	RCL D		
W009	STO (I)		
W010	1		
W011	STO + I		
W012	INPUT A	A?	Prompts for azimuth in D.MMSS
W013	RCL A		
W014	HMS→		
W015	STO A		
W016	INPUT D	D?	Prompts for distance (D?)
W017	RCL D		
W018	$x \geq 0$?		
W019	GTO W023		
W020	ABS		
W021	STO D		
W022	SF 3		
W023	RCL A		
W024	COS		
W025	RCL × D		
W026	RCL A		
W027	SIN		
W028	RCL × D		
W029	0 **i** 1		[Key in as 0, then **i**, then 1, press ENTER]
W030	×		
W031	+		
W032	STO (I)		
W033	10		
W034	STO I		
W035	RCL (I)		
W036	STO A		
W037	1		
W038	STO + I		

W039	RCL (I)		
W040	STO D		
W041	1		
W042	STO + I		
W043	CLSTK		
W044	RCL (I)		
W045	RTN		Complex number now in stack in X

Special Note

This program is designed to work with the B program (Closure 5) that computes a traverse closure and area, but allows for curves in the polygon boundary. In the Closure 5 program, the way to signal that an entered side is a chord is to enter a negative distance. This would ordinarily result in a complex number that was perfectly valid for computation, which meant that there was no simple way to signal the entry of a chord. The V program (Utility 3) was modified to check if the distance was negative, and if so, to set Flag 3, take the absolute value of the distance, and then proceed as before. The Closure 5 program checks if Flag 3 is set when the W sub-program returns, and processes the line based on the state of Flag 3.

Notes

(1) A program that allows the user to enter a vector as two separate components, azimuth and distance, with the azimuth in D.MMSS (HP format), and have it converted to a complex number, with the azimuth component in decimal degrees. If the distance entered is negative, the value is made positive and Flag 3 is set to indicate the negative distance.

(2) Calculator should be set in DEGREES mode. Press MODE, then 1.

(3) Because the calculator uses data entry into the A and D storage registers to allow simple prompting, there is the potential of this program deleting any data already in those storage registers. This could be a problem if this program was called as a sub-routine from within another program. To avoid this problem, the program copies the contents of storage register A to storage register 10, and the contents of storage register D to storage register 11. At the conclusion of the program, the values are copied back into storage registers A and D.

(4) Because the program replaces everything that was on the stack before it ran, the stack is cleared of all data before the result of the calculation is returned to the stack.

(5) While copying back the contents of registers 10 and 11, the program stores the result of the calculation in storage register 12.

(6) If using a program that requires storage registers 10, 11 and 12, change the value of 10 in lines W002 and W033 to a suitable number, so that the set of three storage registers selected aren't used elsewhere.

Operation

Press XEQ W ENTER.

The calculator prompts with A? to enter the azimuth in degrees, minutes and seconds (HP notation). Key in the azimuth and press R/S.

The calculator prompts with D? to enter the distance. Key in the distance and press R/S.

The calculator displays the complex number representing the vector in the X register, or line 2 of the display. The remainder of the stack is zeros.

Examples

1. 63° 15' 47" and 105.528 will give a complex number of 105.5280 θ 63.2631 in polar mode (with FIX 4 set for the display), or 47.4765 **i** 94.2451 in rectangular mode.

2. 128° 15' 47" and 105.528 will give a complex number of 105.5280 θ 128.2631, or –65.3506 **i** 82.8580.

3. 237° 15' 47" and 105.528 will give a complex number of 105.5280 θ –122.7369, or –57.0677 **i** –88.7662.

4. 333° 15' 47" and 105.528 will give a complex number of 105.5280 θ –26.7369, or 94.2451 **i** –47.4765.

5. 397° 15' 47" and 105.528 will produce 105.5280 θ 37.2631, or 83.9859 **i** 63.8946.

6. 63° 15' 47" and –105.528 will give a complex number of 105.5280 θ 63.2631 in polar mode, or 47.4765 **i** 94.2451 in rectangular mode. This is not what would be expected in normal mathematical work, but a consequence of always taking the distance to be positive.

7. 128° 15' 47" and –105.528 will give a complex number of 105.5280 θ 128.2631, or –63.3506 **i** 82.8580. Again, not standard mathematically.

Storage Registers Used

A Stores the azimuth value, initially in D.MMSS, then in decimal degrees.

D Stores the distance value.

I Used to store the value for indirect addressing of registers 10, 11 and 12.

10 Temporary storage for the contents of storage register A while the program runs.

11 Temporary storage for the contents of storage register D while the program runs.

12 Temporary storage for the answer while copying back A and D.

Labels Used

Label **W** Length = 146 Checksum = A1A3

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

HP-35S Calculator Program — Utilities 2A

Azimuth and Distance from Co-ordinates ('Inverse')

Programmer: Dr. Bill Hazelton

Date: September, 2008. Version: 1.0 Mnemonic: **I** for '**I**nverse' Program

Line	Instruction	Display	User Instructions
I001	LBL I		LBL I
I002	CLSTK		CLEAR 5
I003	SF 10		FLAGS 1 .0
I004	COORD INVERSE		(Key in as EQN RCL C, RCL O, etc.; ENTER to end)
I005	PSE		PSE
I006	NTR FAR POINT		(Key in as EQN RCL N, RCL T, etc.; ENTER to end)
I007	PSE		PSE
I008	XEQ I029		
I009	RCL P		
I010	STO F		STO F
I011	NTR NEAR POINT		(Key in as EQN RCL N, RCL T, etc.; ENTER to end)
I012	PSE		PSE
I013	XEQ I029		
I014	RCL P		
I015	STO N		STO N
I016	RCL F		
I017	RCL– N		
I018	STO V		STO V
I019	ARG		ARG
I020	x ≥ 0?		x ? 0 5
I021	GTO I024		
I022	360		
I023	+		
I024	→HMS		→HMS
I025	RCL V		
I026	ABS		ABS
I027	STOP		Press R/S
I028	GTO I006		
I029	CLx		CLEAR 1
I030	STO N		STO N
I031	STO E		STO E
I032	KEY IN N		(Key in as EQN RCL K, RCL E, etc.; ENTER to end)
I033	PSE		PSE
I034	INPUT N		INPUT N
I035	KEY IN E		(Key in as EQN RCL K, RCL E, etc.; ENTER to end)
I036	PSE		PSE
I037	INPUT E		INPUT E
I038	RCL E		
I039	0 *i* 1		(Key in as 0, then **i**, then 1, press ENTER.)

Azimuth and Distance from Co-ordinates ('Inverse')

I040	×		
I041	RCL+ N		
I042	STO P		STO P
I043	RTN		RTN

This program is designed to hide completely the complex number work that the calculator performs to compute the azimuth and distance from the co-ordinates. All values entered and returned are presented in much the same manner as with the equivalent operation (Rectangular to Polar) in the HP-33S calculator.

Notes

(1) Set the calculator into DEGREES mode (press MODE 1) before starting, to make sure that you get degrees, minutes and seconds for the azimuth.

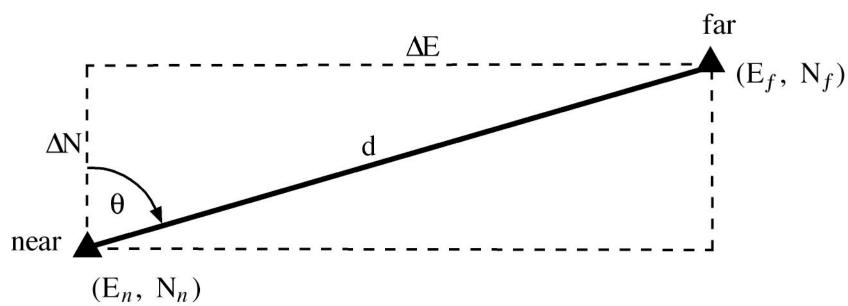
(2) This is a basic co-ordinate 'inverse' program that computes the azimuth and distance between two points whose co-ordinates are supplied. While the program works with complex number representations internally, this is hidden from the user.

(3) Azimuths are by themselves in HP notation, i.e., DDD.MMSSss.

(4) In order to display the prompts, this program sets Flag 10. However, the program never formally ends, because it is up to the user to decide when to stop and move control elsewhere. So the program never clears Flag 10. If you require Flag 10 to be clear, in order to process equations, you must do this manually.

Theory

If the co-ordinates of two points are given, there are two ways by which the azimuth and distance between them may be derived. These are equivalent, and will produce the same result. This HP-35s program uses the second method, viz., vectors stored as complex numbers.



1. **Using co-ordinates** alone, as in the figure above the differences between the easting co-ordinates of the points (ΔE) and difference between the northing co-ordinates of the two points (ΔN) are obtained. The point from which the azimuth is desired to come is termed the 'near' point, while the point to which the azimuth points is termed the 'far' point. So:

$$\Delta E = E_f - E_n$$

$$\Delta N = N_f - N_n$$

-

Azimuth and Distance from Co-ordinates ('Inverse')

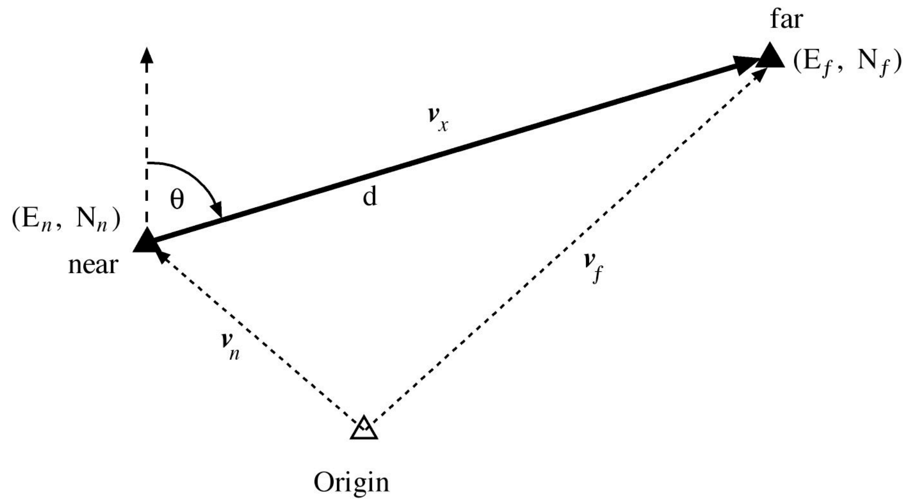
The distance, d, between the two points is computed using:

$$d = \sqrt{\Delta E^2 + \Delta N^2}$$

The azimuth, θ, between the two points is computed using:

$$\theta = \arctan\left(\frac{\Delta E}{\Delta N}\right)$$

The ATAN2 function (in some programming languages and Excel) takes the two components, ΔE and ΔN, as separate arguments, and returns an azimuth in the range –180° to +180°. This is easily converted to an azimuth in the range 0° to 360° The HP calculator function R→P performed a similar process, but it is not present on the HP-35s, so the angle must be put into its correct quadrant manually, as the ATAN function returns a value between –90° and +90°.



2. **Using vectors**, as shown in the figure above, the co-ordinates of each point are entered as the components of a pair of 2-D vectors. In the HP-35s, 2-D vectors are best handled as complex numbers, so that the co-ordinates are stored as the two vectors from the origin, $\mathbf{v}_f$ and $\mathbf{v}_n$, as follows:

$$\mathbf{v}_f = N_f + \boldsymbol{i}\ E_f$$

$$\mathbf{v}_n = N_n + \boldsymbol{i}\ E_n$$

The vector from the near point to the far point, $\mathbf{v}_x$, is then the difference between the two vectors:

$$\mathbf{v}_x = \mathbf{v}_f - \mathbf{v}_n$$

This calculation is handled by a simple vector (albeit using complex numbers) subtraction in the HP-35s.

To obtain the length of the vector $\mathbf{v}_x$, its absolute value must be taken (using the ABS function):

$$d = |\mathbf{v}_x|$$

-

Azimuth and Distance from Co-ordinates ('Inverse')

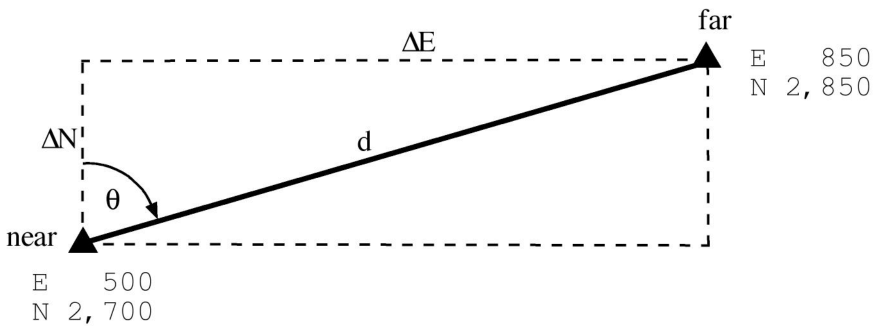
The azimuth of the vector $\mathbf{v}_x$ is calculated as the argument of the vector, using the ARG function:

$$\theta = \arg(\mathbf{v}_x)$$

The HP-35s returns the azimuth in the units in which the calculator us currently set, usually degrees. The value will be in the range –180° to +180°. The program brings this value into the range 0° to 360°, and converts it to degrees, minutes and seconds, in HP notation (i.e., HHH.MMSSsss).

Sample Computations and Running the Program

Example 1



In the above example, $\Delta E = 350$ and $\Delta N = 150$. Using the first method discussed above:

$$d = \sqrt{\Delta E^2 + \Delta N^2} = \sqrt{122{,}500 + 22{,}500} = \sqrt{145{,}000} = 380.789$$

$$\theta = \arctan\left(\frac{\Delta E}{\Delta N}\right) = \arctan\left(\frac{350}{150}\right) = \arctan(2.333\ldots) = 66°\ 48'\ 05''$$

Using the second method, the complex numbers formed are:

$$\mathbf{v}_f = 2{,}850 + \boldsymbol{i}\ 850$$

$$\mathbf{v}_n = 2{,}700 + \boldsymbol{i}\ 500$$

and the difference between them is:

$$\mathbf{v}_x = 150 + \boldsymbol{i}\ 350$$

so $d = |\mathbf{v}_x| = \sqrt{350^2 + 150^2} = \sqrt{122{,}500 + 22{,}500} = \sqrt{145{,}000} = 380.789$

and $\theta = \arg(\mathbf{v}_x) = 66°\ 48'\ 05''$

Using the calculator for this process, begin by pressing XEQ I and then the Enter key.

The calculator briefly displays COORD INVERSE, to indicate the program, then NTR FAR POINT, to tell the user to enter the co-ordinates of the far point, and finally KEY IN N. The calculator then prompts for entry of the northing co-ordinate of the far point, as follows:

N?

0.0000

Key in the northing of the far point, in this case 2850, and press R/S.

The calculator briefly displays KEY IN E, then prompts for keying in the easting of the far point:

E?
0.0000

Key in the easting of the far point, in this case 850, and press R/S.

The calculator briefly displays NTR NEAR POINT, then KEY IN N, then prompts the user to key in the northing of the near point, displaying:

N?
0.0000

Key in the northing of the near point, in this case 2700, and press R/S.

The calculator briefly displays KEY IN E, then prompts for keying in the easting of the near point:

E?
0.0000

Key in the easting of the near point, in this case 500, and press R/S.

The calculator then displays the azimuth of the line in line 1 (the upper line of the display) in HP notation (DDD.MMSSsss), and the length of the line in line 2 (the lower line of the display. In this case, the display shows:

66.4805	(meaning 66° 48' 05", the azimuth of the line)
380.7887	(the length of the line)

At this point, if the user presses R/S, the program prompts for the northing of the far point, as discussed above. Otherwise, the user can keep working on whatever they were doing before. Note that Flag 10 will now be set.

Example 2

Near Point	E 500	Far Point	E 850
	N 2,700		N 2,600

Begin by pressing XEQ I and then the Enter key.

The calculator briefly displays COORD INVERSE, to indicate the program, then NTR FAR POINT, to tell the user to enter the co-ordinates of the far point, and finally KEY IN N. The calculator then prompts for entry of the northing co-ordinate of the far point, as follows:

Azimuth and Distance from Co-ordinates ('Inverse')

N?
0.0000

Key in the northing of the far point, in this case 2600, and press R/S.

The calculator briefly displays KEY IN E, then prompts for keying in the easting of the far point:

E?
0.0000

Key in the easting of the far point, in this case 850, and press R/S.

The calculator briefly displays NTR NEAR POINT, then KEY IN N, then prompts the user to key in the northing of the near point, displaying:

N?
0.0000

Key in the northing of the near point, in this case 2700, and press R/S.

The calculator briefly displays KEY IN E, then prompts for keying in the easting of the near point:

E?
0.0000

Key in the easting of the near point, in this case 500, and press R/S.

The calculator then displays the azimuth of the line in line 1 (the upper line of the display) in HP notation (DDD.MMSSsss), and the length of the line in line 2 (the lower line of the display. In this case, the display shows:

105.5643	(meaning 105° 56' 43", the azimuth of the line)
364.0055	(the length of the line)

At this point, if the user presses R/S, the program prompts for the northing of the far point, as discussed above. Otherwise, the user can keep working on whatever they were doing before. Note that Flag 10 will now be set.

Distance = 364.0055 Azimuth = 105° 56' 43"

Example 3

Near Point	E	500	Far Point	E	450
	N	2,700		N	2,500

Process the co-ordinates through the calculator as above, to get the following results:

Distance = 206.1553 Azimuth = 194° 02' 10"

-

Azimuth and Distance from Co-ordinates ('Inverse')

Example 4

Near Point	E 500	Far Point	E 300
	N 2,700		N 2,800

Process the co-ordinates through the calculator as above, to get the following results:

Distance = 223.6068 Azimuth = 296° 33' 54"

Storage Registers Used

E Easting co-ordinates of the entered points.

F Co-ordinates of far point, as a complex number.

N Northing co-ordinate of the entered points, then the co-ordinates of the near point, as a complex number.

P Temporary storage of co-ordinates of entered point, as a complex number.

V Vector from the near point to the far point, stored as a complex number.

Statistical Registers: not used.

Labels Used

Label **I** Length = 191 Checksum = BA0E

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Routines Called

None. The program does call the same segment of code twice, for co-ordinate entry, but this is within the program.

Flags Used

The program sets Flag 10, to allow the display of prompts. However, as the program never formally ends (it is used by the user as needed), it never clears this flag. If the flag is needed to be cleared, clear it manually.

HP-35s Calculator Program XYZ 1

Convert Latitude, Longitude and Height on any ellipsoid to XYZ Geocentric Co-ordinates

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Mnemonic**: **G** for 'lat/long to **G**eocentric co-ordinates.'

Line	Instruction	Display	User Instructions
G001	LBL G		Press XEQ G ENTER to run program
G002	CLSTK		
G003	FS? 10		
G004	GTO G008		
G005	SF 1		
G006	SF 10		
G007	GTO G009		
G008	CF 1		
G009	LATLONG TO XYZ		
G010	PSE		
G011	6378137		Value of a for WGS84/NAD83/GRS80
G012	STO A		
G013	0.006694381		Value of e^2 for WGS84/NAD83/GRS80
G014	STO E		
G015	SEMIMAJOR AXIS		
G016	PSE		
G017	INPUT A	6378137	Enter value of a if different
G018	E SQUARED		
G019	PSE		
G020	INPUT E	0.006694381	Enter value of e^2 if different
G021	ENTER LATITUDE		
G022	PSE		
G023	INPUT F		Enter ϕ of point (DDD.MMSSss)
G024	ENTR LONGITUDE		
G025	PSE		
G026	INPUT L		Enter λ of point (DDD.MMSSss)
G027	ENTER HEIGHT		
G028	PSE		
G029	INPUT H		Enter h of point
G030	RCL A		
G031	1		
G032	RCL F		
G033	HMS→		
G034	SIN		
G035	x^2		
G036	RCL× E		
G037	–		
G038	$\sqrt{x}$		

Lat/Long/Ht to XYZ Geocentric Co-ordinates

Line	Instruction	Display	User Instructions
G039	÷		
G040	STO V		
G041	RCL+ H		
G042	RCL F		
G043	HMS→		
G044	COS		
G045	×		
G046	RCL L		
G047	HMS→		
G048	COS		
G049	×		
G050	STO X		
G051	GEOCENTRIC X		
G052	PSE		
G053	VIEW X	X co-ordinate	X co-ordinate is displayed
G054	RCL L		
G055	HMS→		
G056	TAN		
G057	×		
G058	STO Y		
G059	GEOCENTRIC Y		
G060	PSE		
G061	VIEW Y	Y co-ordinate	Y co-ordinate is displayed
G062	RCL V		
G063	1		
G064	RCL– E		
G065	×		
G066	RCL+ H		
G067	RCL F		
G068	HMS→		
G069	SIN		
G070	×		
G071	STO Z		
G072	GEOCENTRIC Z		
G073	PSE		
G074	VIEW Z	Z co-ordinate	Z co-ordinate is displayed
G075	FS? 1		
G076	CF 10		
G077	RCL Z		
G078	RCL Y		
G079	RCL X		X, Y, Z values left on stack
G080	RTN		Program ends

Lat/Long/Ht to XYZ Geocentric Co-ordinates

Notes

(1) A program to convert latitude, longitude and ellipsoidal height on any ellipsoid to X, Y, Z geocentric co-ordinates.

(2) The assumption is that the distances are in meters, but by using feet for the semi-major axis of the ellipsoid, co-ordinates in feet will be produced.

(3) The program pre-enters the parameters for the WGS84/NAD83/GRS80 ellipsoid by default (in meters), to save you having to remember these. If you want a different ellipsoid, enter the appropriate a and e^2 values at the prompts (A and E).

(4) The resulting co-ordinates are displayed with a prompt or label. Note that the program does not clear registers after use. You can get ν for the point by using the RCL V keystrokes, for example.

(5) The latitude and longitude are entered in HP notation, i.e., DDD.MMSS. The height is assumed to be in the same units as the semi-major axis, by default, meters.

(6) It is critical to follow the sign convention with latitudes, longitudes and heights. Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in the US are negative. Heights below the ellipsoid must be entered as negative.

(7) The program sets Flag 10 to allow display of equations as prompts. However, it stores the state of Flag 10 (in Flag 1), and restores the original state after the program ends. If you don't finish the program, you may need to check and reset Flag 10.

(8) When the program ends, it leaves the X, Y and Z values on the stack, in the X, Y and Z stack registers, respectively. They can then be used by another program. The input values are still in the original storage registers (in HP angle notation for latitude and longitude), and can be recalled later or by other programs. This program may be called by another program as a sub-routine, and will return the X, Y, Z values on the stack.

Theory

The program implements the following four equations:

$$X = (\nu + h)\ \cos\phi\ \cos\lambda \qquad [1]$$

$$Y = (\nu + h)\ \cos\phi\ \sin\lambda \qquad [2]$$

$$Z = [\nu(1 - e^2) + h]\ \sin\phi \qquad [3]$$

$$\nu = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \qquad [4]$$

These provide a direct solution. Values for the ellipsoid (a and e^2) are requested, although default values for NAD83/WGS84/GRS80 are supplied (these can be overwritten). The latitude, longitude and ellipsoidal height of the point are requested of the user. The program supplies the solution in three pieces.

Lat/Long/Ht to XYZ Geocentric Co-ordinates

Sample Computation

Inputs	a = 6 378 137 m
	e^2 = 0.006 694 381 (WGS84/NAD83/GRS80 parameters)
	ϕ = 35° 00' 00" N (entered as 35.0000) Latitude of point
	λ = 75° 00' 00" W (entered as –75.0000) Longitude of point
	h = 200 m Ellipsoidal height of point
Results	X = 1 353 776.483 m
	Y = –5 052 362.616 m
	Z = 3 637 981.622 m

Running the Program

Begin by pressing XEQ G ENTER to start the program. The calculator briefly displays LATLONG TO XYZ, then SEMIMAJOR AXIS, briefly.

The calculator displays: A?
6,378,137.0000

This is the NAD83/WGS84/GRS80 ellipsoid semi-major axis. If this is OK, press R/S; if not key in correct value and press R/S. The calculator displays E SQUARED, briefly.

The calculator displays: E?
0.006694381 (suitably rounded, according to your settings)

This is the eccentricity of the NAD83/WGS84/GRS80 ellipsoid, e^2. If this OK, press R/S; if not, key in correct value and press R/S. The calculator displays ENTER LATITUDE, briefly.

The calculator displays: F?
[*Whatever value happens to be in this register*]

Key in the latitude of the point and press R/S. Use negative values for the southern hemisphere. In the given example, key in 35 and press R/S. The calculator displays ENTR LONGITUDE, briefly.

The calculator displays: L?
[*Whatever value happens to be in this register*]

Key in the longitude of the point and press R/S. Use negative values in the western hemisphere. In the given example, key in –75 and press R/S. The calculator displays ENTER HEIGHT, briefly.

The calculator displays: H?
[*Whatever value happens to be in this register*]

Key in the ellipsoidal height for the point and press R/S. Use negative values for heights below the ellipsoid. In the given example, key in 200 and press R/S.

The calculator displays: GEOCENTRIC X, briefly, then:
X=
1,353,776.4829 Press R/S

The calculator displays: GEOCENTRIC Y, briefly, then:
Y=
–5,052,362.6164 Press R/S

Lat/Long/Ht to XYZ Geocentric Co-ordinates

The calculator displays: GEOCENTRIC Z, briefly, then:
Z=
3,637,981.6216 Press R/S

The calculator now completes the program, resetting Flag 10, and placing the computed values in the stack. The stack now holds the following values:

Stack Level	Contents
T	Geocentric Z Co-ordinate
Z	Geocentric Z Co-ordinate
Y	Geocentric Y Co-ordinate
X	Geocentric X Co-ordinate

These calculations agree with the NGS website computations to within 0.003 m.

Storage Registers Used

A Semi-major axis of the ellipsoid, *a*. By default, set to 6378137 m

E Squared eccentricity of the ellipsoid, e^2. By default, set to 0.006694381.

F Latitude (geodetic) of the point, ϕ (in HP notation).

H Ellipsoidal height of the point, h.

L Longitude of the point, λ (in HP notation).

V The radius of curvature of the ellipsoid in the prime vertical, ν.

X Geocentric X co-ordinate of the point.

Y Geocentric Y co-ordinate of the point.

Z Geocentric Z co-ordinate of the point.

Labels Used

Label **G** Length = 373 Checksum = 77D4

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

Note that the checksum depends upon entering prompts exactly as written in the program listing.

Reference

The NGS website for the interactive XYZ ⇔ lat/long/height converter:

http://www.ngs.noaa.gov/TOOLS/XYZ/xyz.shtml

HP-35s Calculator Program XYZ 2

Convert XYZ Geocentric Co-ordinates on any ellipsoid to Latitude, Longitude and Height

Programmer: Dr. Bill Hazelton

Date: October, 2007. **Mnemonic**: **L** for 'XYZ to **L**at/**L**ong'

Line	Instruction	Display	User Instructions
L001	LBL L		Press XEQ L ENTER to run program
L002	CLSTK		
L003	FS? 10		
L004	GTO L008		
L005	SF 1		
L006	SF 10		
L007	GTO L009		
L008	CF 1		
L009	XYZ TO LATLONG		
L010	PSE		
L011	6378137		Value of a for WGS84/NAD83/GRS80
L012	STO A		
L013	0.006694381		Value of e^2 for WGS84/NAD83/GRS80
L014	STO E		
L015	SEMIMAJOR AXIS		
L016	PSE		
L017	INPUT A	6378137	Enter value of a if different; Press R/S
L018	E SQUARED		
L019	PSE		
L020	INPUT E	0.006694381	Enter value of e^2 if different; Press R/S
L021	ENTER X CO-ORD		
L022	PSE		
L023	INPUT X		Enter X co-ordinate of point; Press R/S
L024	ENTER Y CO-ORD		
L025	PSE		
L026	INPUT Y		Enter Y co-ordinate of point; Press R/S
L027	ENTER Z CO-ORD		
L028	PSE		
L029	INPUT Z		Enter Z co-ordinate of point; Press R/S
L030	RCL Y		
L031	RCL X		
L032	÷		
L033	ATAN		
L034	STO L		
L035	1		
L036	RCL– E		
L037	RCL A		
L038	x^2		

XYZ Geocentric Co-ordinates to Lat/Long/Ht

Line	Instruction
L039	×
L040	√x
L041	STO B
L042	RCL A
L043	x^2
L044	RCL B
L045	x^2
L046	–
L047	RCL B
L048	x^2
L049	÷
L050	STO D
L051	RCL X
L052	x^2
L053	RCL Y
L054	x^2
L055	+
L056	√x
L057	STO P
L058	RCL Z
L059	x < > y
L060	÷
L061	RCL A
L062	RCL÷ B
L063	×
L064	ATAN
L065	STO U
L066	SIN
L067	3
L068	y^x
L069	RCL× B
L070	RCL× D
L071	RCL+ Z
L072	RCL U
L073	COS
L074	3
L075	y^x
L076	RCL× A
L077	RCL× E
L078	RCL P
L079	x < > y
L080	–
L081	÷
L082	ATAN
L083	STO F

Line	Instruction
L084	RCL A
L085	1
L086	RCL F
L087	SIN
L088	x^2
L089	RCL× E
L090	–
L091	√x
L092	÷
L093	STO V
L094	RCL F
L095	COS
L096	×
L097	RCL L
L098	COS
L099	×
L100	STO C
L101	RCL– X
L102	x^2
L103	STO H
L104	RCL C
L105	x^2
L106	STO G
L107	RCL C
L108	RCL L
L109	TAN
L110	×
L111	STO C
L112	x^2
L113	STO+ G
L114	RCL C
L115	RCL– Y
L116	x^2
L117	STO+ H
L118	RCL V
L019	1
L120	RCL– E
L121	×
L122	RCL F
L123	SIN
L124	×
L125	STO C
L126	x^2
L127	STO+ G
L128	RCL C

Line	Instruction
L129	RCL– Z
L130	x^2
L131	STO+ H
L132	RCL X
L133	x^2
L134	RCL Y
L135	x^2
L136	+
L137	RCL Z
L138	x^2
L139	+
L140	√x
L141	RCL G
L142	√x
L143	–
L144	ENTER
L145	ABS
L146	÷
L147	RCL H
L148	√x
L149	×
L150	STO H
L151	RCL F
L152	→HMS
L153	STO F
L154	LATITUDE
L155	PSE
L156	VIEW F
L157	RCL L
L158	→HMS
L159	STO L
L160	LONGITUDE
L161	PSE
L162	VIEW L
L163	ELLIPSE HEIGHT
L164	PSE
L165	VIEW H
L166	RCL H
L167	RCL L
L168	RCL F
L169	FS? 1
L170	CF 10
L171	RTN

XYZ Geocentric Co-ordinates to Lat/Long/Ht

Notes

(1) A program to convert X, Y, Z geocentric co-ordinates to latitude, longitude and ellipsoidal height on any ellipsoid.

(2) The assumption is that the distances are in meters, but by using feet for the semi-major axis of the ellipsoid, co-ordinates in feet can be produced.

(3) The program pre-enters the parameters for the WGS84/NAD83/GRS80 ellipsoid by default (in meters), to save you having to remember these. If you want a different ellipsoid, enter the appropriate a and e^2 values at the prompt (A and E). To use the provided values, just press R/S when they appear.

(4) The resulting latitude, longitude and height are displayed with a prompt or label. Note that the program does not clear registers after use. You can get ν for the point by using the RCL V keystrokes.

(5) The latitude and longitude are displayed in HP notation, i.e., DDD.MMSS. The height is assumed to be in the same units as the semi-major axis, by default, meters.

(6) The sign convention with latitudes, longitudes and heights is the standard one, as follows. Latitudes in the southern hemisphere are negative. Longitudes west of Greenwich are negative, i.e., all longitudes in the US are negative. Heights below the ellipsoid are shown as negative.

(7) Pay particular attention to the sign of the co-ordinate values for the point. West of longitude 90° W, all X values will be negative; west of Greenwich (i.e., all of the US) all Y values are negative; south of the equator, all Z values will be negative. Incorrect signs will throw the position out rather dramatically.

(8) Owing to rounding in the calculator, it is possible for a value like 80° 00' 00" to be displayed as 79.5960, rather than 80.0000. You can convert the results to the appropriate representation in your head or on paper, as there is no difference in the internal calculations or the results.

(9) When the program finishes, it places the computed values on the stack. The height of the point above the ellipsoid is in the Z stack register. The longitude (in HP notation) is in the Y stack register. The latitude (in HP notation) is in the X stack register. This allows the program to be called from another program (as a sub-routine) and return values on the stack for further processing.

(10) The program sets Flag 10, to allow prompts to be displayed. When the program finishes, it sets Flag 10 back to its original value. To do this, the program uses Flag 1.

Theory

The program implements the following equations:

$$\lambda = \arctan\left(\frac{Y}{X}\right) \qquad [1]$$

$$b^2 = a^2(1 - e^2) \qquad \text{[the semi-minor axis length]} \qquad [2]$$

XYZ Geocentric Co-ordinates to Lat/Long/Ht

$$p = \sqrt{X^2 + Y^2} \qquad [3]$$

$$\tan u = \left(\frac{Z}{p}\right)\left(\frac{a}{b}\right) \qquad [4]$$

$$\nu = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \qquad [5]$$

$$\phi = \arctan\left(\frac{Z + \varepsilon b \sin^3 u}{p - e^2 a \cos^3 u}\right) \qquad [6]$$

$$\varepsilon = e'^2 = \frac{(a^2 - b^2)}{b^2} \quad \text{[the second eccentricity]} \qquad [7]$$

$$X = \nu \cos\phi \cos\lambda \qquad [8]$$

$$Y = \nu \cos\phi \sin\lambda \qquad [9]$$

$$Z = \nu(1 - e^2) \sin\phi \qquad [10]$$

Equation [1] provides a direct solution for the longitude, λ. The program then computes a variety of intermediate results, before using equation [6] to compute the latitude. This is a direct solution. The program then computes the X, Y, Z location for the point on the ellipsoid at φ, λ, using equations [8] to [10]. The distance between this point and the given X, Y Z co-ordinates is computed to determine *h*, the ellipsoidal height.

The distances to the Earth's center from the original X, Y, Z co-ordinates and the point on the ellipsoid are computed to get the correct sign for *h*, i.e., is the point above or below the surface of the ellipsoid.

The equations are from Bowering (1976).

Sample Computation

Inputs

a = 6 378 137 m
e^2 = 0.006 694 381 (WGS84/NAD83/GRS80 parameters)
X = 1 353 776.483 m
Y = –5 052 362.616 m
Z = 3 637 981.622 m

Results

φ = 35° 00' 00" N (displayed as F = 35.000000)
λ = 75° 00' 00" W (displayed as L = –75.000000)
h = 200.000 m (displayed as H = 200.0000049)

Note that the precision of the answer depends upon the precision of the input. One millimeter (0.001m) at the surface of the ellipsoid equates to 0.00003" of arc of latitude and generally a smaller amount of longitude. Consider your input precisions and adjust the quoted precision of the outputs to match. You can use the SHOW function to see all the digits in a number.

XYZ Geocentric Co-ordinates to Lat/Long/Ht

Running the Program

Begin by pressing XEQ L ENTER

The calculator displays XYZ to LATLONG, briefly, then displays SEMIMAJOR AXIS, briefly.

The calculator displays: A?
6,378,137.0000

This is the NAD83/WGS84/GRS80 ellipsoid semi-major axis. If this is OK, press R/S; if not key in correct value and press R/S.

The calculator displays: E SQUARED, briefly.
E?
0.006694381 (suitably rounded, according to your settings)

This is the eccentricity of the NAD83/WGS84/GRS80 ellipsoid, e^2. If this OK, press R/S; if not, key in correct value and press R/S.

The calculator displays: ENTER X COORD, briefly.
X?
[*Whatever value happens to be in this register*]

Key in the X co-ordinate of the point and press R/S. Remember to use the appropriate sign, if negative. In the given example, key in 1353776.483 and press R/S.

The calculator displays: ENTER Y COORD, briefly.
Y?
[*Whatever value happens to be in this register*]

Key in the Y co-ordinate of the point and press R/S. Remember to use the appropriate sign, if negative. In the given example, key in –5052362.616 and press R/S.

The calculator displays: ENTER Z COORD, briefly.
Z?
[*Whatever value happens to be in this register*]

Key in the Z co-ordinate for the point and press R/S. Remember to use the appropriate sign, if negative. In the given example, key in 3637981.622 and press R/S.

The calculator displays RUNNING for a short while.

The calculator displays: LATITUDE, briefly.
F=
35.00000000 Press R/S

The calculator display LONGITUDE, briefly.
L=
–75.0000000 Press R/S

The calculator displays: ELLIPSE HEIGHT, briefly.
H=
200.0000049 Press R/S

The calculator now completes its run, placing the latitude, longitude and height on the stack, and resetting Flag 10 to its state when the program started.

XYZ Geocentric Co-ordinates to Lat/Long/Ht

These calculations agree with the NGS website computations to within 0.001 m in height and 0.00002" in latitude and longitude.

Storage Registers Used

A Semi-major axis of the ellipsoid, a. By default, set to 6378137 m.

B Semi-minor axis of the ellipsoid. Computed from a and e^2.

C Intermediate value.

D Second eccentricity of ellipsoid.

E Eccentricity of the ellipsoid, e^2. By default, set to 0.006694381.

F Latitude (geodetic) of the point, ϕ.

G Distance from ellipsoid point to center of Earth.

H Ellipsoidal height of the point, *h*.

L Longitude of the point, λ.

P Intermediate value.

U Intermediate value.

V The radius of curvature of the ellipsoid in the prime vertical, ν.

X Geocentric X co-ordinate of the point.

Y Geocentric Y co-ordinate of the point.

Z Geocentric Z co-ordinate of the point.

Labels Used

Label **L** Length = 643 Checksum = 36FE

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computation to check proper operation after entry.

References

Bowering, B.R., 1976. Transformation from spatial to geographical co-ordinates. *Survey Review*, No. 181, pp. 323–327.

The NGS website for the interactive XYZ ⇔ lat/long/height converter:

http://www.ngs.noaa.gov/TOOLS/XYZ/xyz.shtml

Programming and Working with Indirectly Addressed Memory on the HP-35s

Introduction

The HP-35s calculator is a continuation of the line of scientific calculators that began with the HP-45 in the 1970s. Its immediate ancestor is the HP-33S, a calculator still in production. One of the significant advances of the HP-35s beyond the HP-33S is the much greater access to the 30K of memory that the calculator possesses.

By being able to transfer control within a program to any program line, using XEQ or GTO, it is possible to do a great deal more in programming, without running out of program labels. In addition, the calculator can address 26 variables (identified with a letter), 6 statistical registers, and an additional 801 storage locations. This memorandum is primarily concerned with working with the 801 additional storage locations.

Accessing Memory

Memory locations that are accessible by letter, e.g., STO A and RCL A, may also be address indirectly. Similarly, the statistical registers may be addressed indirectly. The 800 additional storage locations must be addressed indirectly.

Indirect access to memory is done through two special memory locations, the variables I and J. These are ordinary memory locations, but the calculator can take the contents of the I and J locations and use it as the address of another memory location. In this sense, I and J can be used like pointers in regular programming.

To indirectly access a memory location, an integer is stored in I or J. To store a value in the memory location pointed to by the value in I or J, the key sequence STO (I) or STO (J) is used, respectively. This stores the value currently in the x register of the stack (the bottom of the stack, and the lower line of the display) into the memory location pointed to by the value in I or J, respectively. Similarly, RCL (I) and RCL (J) recall the value in the memory location pointed to by the value in I or J, respectively, and copies it to the x register of the stack. Register arithmetic, e.g. STO + (I), RCL ÷ (J), is also possible, and indirect memory locations can be used in the same way as ordinary memory locations. Note that I and J can be used in any way; there are two of them to make programming easier when working with pairs of values, such as co-ordinates of paired statistical data.

The (I) and (J) symbols cannot be entered using the parentheses key. They have their own keys, i.e., 0 and the decimal point keys. They can be seen there in red, acting like the letters for variables and labels. Use these keys to operate indirect memory addressing.

The addresses used in the I and J variables are as follows. To access the variables A through Z, use the numbers –1 through –26. The statistical registers use the numbers –27 through –32. The 801 additional locations use positive numbers 0 through 800.

Examples

Suppose that I want to store a value (say 123.456) into memory location 321. I put the value 321 (the memory location) into the x register of the stack and press STO I. Then I put 123.456 (the value I want to store) into the x register of the stack and press STO (I). The STO function then moves the value in the x register of the stack to the memory location specified in the I variable.

If I want to recall the value in memory location 230, I put 230 into the x register of the stack and press STO J. I then press RCL (J), and the value in memory location 230 is recalled to the x register of the stack.

Memory Allocation

The calculator has the memory locations A through Z and the statistical registers enabled at all times. However, the 801 additional memory locations are not allocated until needed, and can be de-allocated (freed) when no longer needed. As this indirectly addressed memory is shared with program memory, this allows it to be held for either purpose. However, the calculator has no explicit functions for allocating and de-allocating memory. This happens semi-automatically.

When a non-zero value is stored in a specific memory location (from the 801), using STO (I) or STO (J), memory is allocated for storage up to the memory location used for that STO operation. For example, if I store the value 123.456 in memory location 500, using STO (I) or STO (J), memory locations 0 through 500 will be allocated to storage and will be accessible. They should also be given initial values of zero.

So memory allocation takes place automatically, and there is no need to initialize the memory to zero. In fact, attempting to do so may cause an INVALID (I) or (J) error, because you tried to access memory that wasn't yet allocated. This will happen if you try to STO a value of zero into memory that isn't allocated.

However, if you are depending upon the STO function to allocate memory, remember that it requires a non-zero value being stored at a location to allocate that memory. If there is any possibility that the value that you are storing is zero, that memory location will not be allocated. This may not be a problem if you are simply storing a string of values, as unless the last value is zero, later non-zero values will allocate the memory and set it to zero.

A better solution is to decide about how much memory you will need, and allocate that ahead of time. That avoids problems with valid data values of zero, as well as initializing the memory locations to zero. If you make sure that there is an initial STO (I) or STO (J) operation, rather than STO+ or similar, then there will be a valid value in the memory location before you start any register arithmetic.

For a code example, try the following to allocate 101 memory locations (0 through 100):

```
100
STO I
STO (I)
```

You could insert some other value between the STO I and STO (I) lines, but the above code fragment will allocate all memory locations from 0 through 100 that are not already allocated, store the value 100 in memory location 100, and for those memory locations not already allocated, set their contents to zero.

De-allocating memory also has no explicit command on the calculator. Memory is de-allocated when it is returned to zero, but this cannot happen if there are still memory locations allocated above the memory location being set to zero (i.e., with a greater/higher address number). If you explicitly set the memory location at the top of allocated memory to zero, i.e., STO a value of zero in that memory location, that location will be de-allocated, but no other locations will be de-allocated, even if they are filled with zeros.

The solution to de-allocate memory is to set up a loop that sets memory locations to zero from the top down, such as the following code fragment, based on de-allocating the memory locations from 100 down to 20:

Line	**Instruction**
K240	100
K241	STO I
K242	0
K243	STO (I)
K244	20
K245	RCL I
K246	x < y?
K247	GTO K251
K248	1
K249	STO— I
K250	GTO K242
K251	PROGRAM END

Of course, care must be taken to see that the starting value is at the top of allocated memory, otherwise the code simply sets memory locations to zero without any de-allocation. Unfortunately, there appears to be no easy way to ascertain how many memory locations are currently allocated within a program. You can see it as the left number on the lower line of the display when the ↰ **MEM** function is used, but this function is not programmable. Note that this value is the number of memory locations allocated, so 25 means the ‘top’ memory location is 24, as memory locations 0 through 24 are allocated.

When programming to de-allocate memory locations, it is best to de-allocate only those that you are sure you allocated, otherwise you will get an INVALID (I) or (J) error. So there are advantages to being quite explicit about allocating and de-allocating memory locations. Remember also that if a program that allocates memory is halted before the end, it will not have de-allocated memory locations, so control may have to be moved to this section of the code and it executed to de-allocate the memory locations.

Why De-allocate Memory Locations?

The calculator’s 30K of memory is shared by programs, memory locations, equations, and a range of internal applications, such as SOLVE, and integration (∫ FN). These all use a fair amount of memory, integration quite a large amount (albeit only while running). To avoid a MEMORY FULL error, which will halt execution, it is wise to de-allocate memory locations that are not needed, so that they can be used by other calculator applications which need temporary memory.

Note also that if you have a lot of program memory space used, this may cut into the available memory locations, so that there may not be the full 801 available. Program memory appears to be allocated and de-allocated automatically, without any need for explicit operation on the user’s part.

Dr. Bill Hazelton.

March, 2008.

HP-35s Calculator Program — Radiations 1

Calculate a Radiation from Two Offsets

Programmer: Dr. Bill Hazelton

Date: March, 2008.

Line	Instruction	Display	User Instructions
0001	LBL O		LBL O
0002	CLSTK		CLEAR 5
0003	FS? 10		FLAGS 3 .0
0004	GTO O008		
0005	SF 1		FLAGS 1 1
0006	SF 10		FLAGS 1 .0
0007	GTO O009		
0008	CF 1		FLAGS 2 1
0009	RAD FM OFFSETS		(Key in using EQN RCL R, RCL A, etc.)
0010	PSE		PSE
0011	ENTER RIGHT AZ		(Key in using EQN RCL E, RCL N, etc.)
0012	PSE		PSE
0013	INPUT R	R?	INPUT R
0014	RCL R		
0015	HMS→		HMS→
0016	STO A		STO A
0017	ENTER LEFT AZ		(Key in using EQN RCL E, RCL N, etc.)
0018	PSE		PSE
0019	INPUT L	L?	INPUT L
0020	RCL A		
0021	RCL L		
0022	HMS→		HMS→
0023	–		
0024	STO E		STO E
0025	ENTER RIGHT OS		(Key in using EQN RCL E, RCL N, etc.)
0026	PSE		PSE
0027	INPUT R	R?	INPUT R
0028	ENTER LEFT OS		(Key in using EQN RCL E, RCL N, etc.)
0029	PSE		PSE
0030	INPUT L	L?	INPUT L
0031	RCL R		
0032	RCL E		
0033	SIN		
0034	×		
0035	RCL E		
0036	COS		
0037	RCL× R		
0038	RCL+ L		
0039	÷		

Radiation from Two Offsets

Line	Instruction	Display	User Instructions
0040	ATAN		↱ ATAN
0041	STO C		↱ STO C
0042	RCL A		
0043	x < > y		
0044	–		
0045	→HMS		↱ →HMS
0046	STO B		↱ STO B
0047	RADIATION AZ		(Key in using EQN RCL R, RCL A, etc.)
0048	PSE		↱ PSE
0049	VIEW B		↰ VIEW B
0050	RCL R		
0051	RCL C		
0052	SIN		
0053	÷		
0054	STO D		↱ STO D
0055	RAD LENGTH		(Key in using EQN RCL R, RCL A, etc.)
0056	PSE		↱ PSE
0057	VIEW D		↰ VIEW D
0058	PROGRAM END		(Key in using EQN RCL P, RCL R, etc.)
0059	PSE		↱ PSE
0060	FS? 1		↰ FLAGS 3 1
0060	CF 10		↰ FLAGS 2 .0
0060	RTN		↰ RTN

Notes

(1) This program allows a radiation (azimuth and distance) to an object to be computed from two offsets measured from two lines of known azimuth to the object.

(2) The two lines from which the offsets are measured do not have to be orthogonal. The closer they are to orthogonal, the better, but the program will work with any realistic set of measurements. It will also work across the 0° line, and with traverse lines in different quadrants.

(3) The purpose of the program is to facilitate conversion of older survey data, in which corners were often located by pairs of offsets, to a form in which a radiation is employed. This will simplify calculations based on traverses and radiations, i.e., vectors.

(4) Azimuths are entered and displayed in HP notation, i.e., DDD.MMSS, at all times.

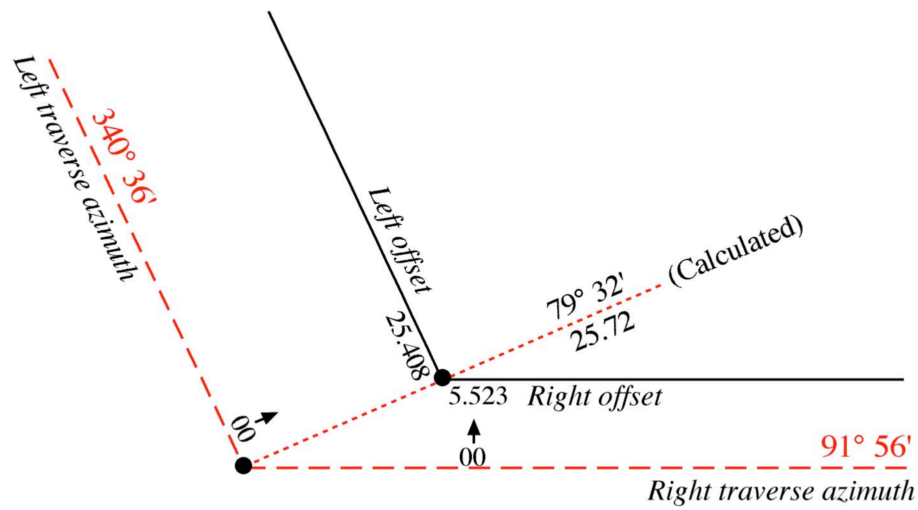
(5) Feet or meters (or any other linear units) can be used, provided their use is consistent.

(6) The original code for this program was developed by Philip R. Price at the State Rivers and Water Supply Commission (SR&WSC) Survey Branch, Victoria, Australia, in November, 1976, for the HP-25 calculator. This program is an update and adaptation for the HP-35s, but is based on Phil Price's original solution and HP-25 implementation.

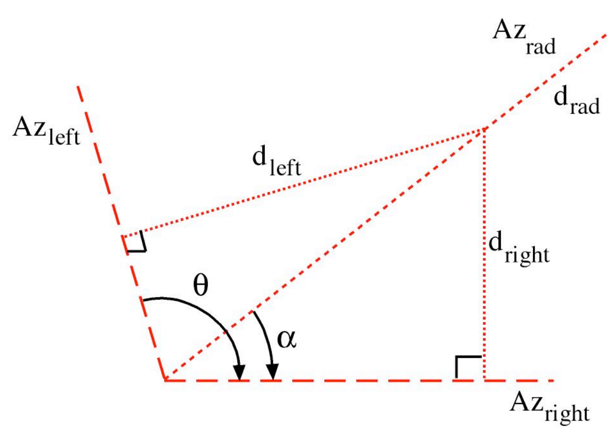
Radiation from Two Offsets

Theory

If an object, commonly a corner, is located by offsets, these can be converted to the equivalent radiation using the program. In the example below, the corner was located by two offsets, being 5.523 on the right and 25.408 on the left. The offsets are measured at right angles to their respective traverse lines, and are the distance between the object and the traverse line. The right traverse line has an azimuth of 91° 56', while the left traverse line has an azimuth of 340° 36'. See the figure below:



Generalizing the diagram and giving it symbols, the general situation becomes as follows:



The angle θ is the difference between the two known azimuths, Az_{left} and Az_{right}. The offset from the right traverse line to the object is d_{right}, while the offset from the left traverse line to the object is d_{left}. Because the offsets are measured at right angles to the traverse lines, their azimuths are also known.

The angle α is the angle between the right traverse line and the radiation, and is determined using the following formula:

Radiation from Two Offsets

$$\alpha = \arctan\left(\frac{d_{right}\ \sin\theta}{d_{left} + d_{right}\ \cos\theta}\right)$$

Subtracting α from Az_{right} gives the azimuth of the radiation, Az_{rad}. The length of the radiation, d_{rad}, is derived using:

$$d_{rad} = \frac{d_{right}}{\sin\alpha}$$

Precision of Results

Note that the precision of the derived azimuth and distance is limited by the precision with which the offsets were measured, as well as the geometry of the offsets and radiation. The program provides the one solution given the data, but without redundant measurements the precision of the solution is unknown. It would be unwise to use azimuths for the radiation that are more precise that about one minute of arc. Experiment with changing the values of the offsets by small amounts that are consistent with their probable precision, as well as changing the azimuths by amounts consistent with their precision over the distances to the offsets, and see what happens to the azimuth and distance values for the radiation.

Sample Computations

1. Using the example given, where the right azimuth is 91° 56', the left azimuth is 340° 36', the right offset is 5.523 and the left offset is 25.408, the radiation has an azimuth of 79° 32' and a distance of 25.72.

2. Using a right azimuth of 160° 36' and a left azimuth of 91° 56', a right offset of 13.272 and a left offset of 15.693, the radiation has an azimuth of 129° 32' and a distance of 25.72.

3. Using a right azimuth of 268° 35', a left azimuth of 195° 12', a right offset of 11.782 and a left offset of 9.467, the radiation has an azimuth of 227° 15' and a distance of 17.84.

4. Using a right azimuth of 340° 36', a left azimuth of 271° 56', a right offset of 11.57 and a left offset of 8.78, the radiation has an azimuth of 300° 55' and a distance of 18.12.

Running the Program

With everything to hand, press XEQ O, then press ENTER.

The calculator briefly displays RAD FM OFFSETS, then briefly displays ENTER RIGHT AZ, then prompts R?

> Key in the value of the azimuth of the right traverse line, in HP notation (DDD.MMSS). Press R/S.

The calculator briefly displays ENTER LEFT AZ, then prompts L?

> Key in the value of the azimuth of the left traverse line, in HP notation (DDD.MMSS). Press R/S.

The calculator briefly displays ENTER RIGHT OS, then prompts R?

Radiation from Two Offsets

Key in the value of the right offset. Press R/S.

The calculator briefly displays ENTER LEFT OS, then prompts L?

Key in the value of the left offset. Press R/S.

The calculator briefly displays RADIATION AZ, then shows B= and the azimuth in HP notation (DDD.MMSS). Press R/S.

The calculator briefly displays RAD LENGTH, then shows D= and the length of the radiation.

Press R/S, the calculator briefly displays PROGRAM END, completes the program and resets Flag 10 to its state at the start of the program. The azimuth (in HP notation) and the distance of the radiation remain on the stack, in the Y and X locations, respectively.

To compute additional radiations from offset, press XEQ O, then press ENTER, and start the program again.

Storage Registers Used

A Azimuth of right traverse line (in decimal degrees)

B Azimuth of the radiation (in HP notation, DDD.MMSS)

C α, the angle between the right traverse line and the radiation (in decimal degrees)

D Length of the radiation

E θ, the angle between the traverse lines (in decimal degrees)

L Left traverse azimuth (in HP notation) or left offset (temporary storage)

R Right traverse azimuth (in HP notation) or right offset (temporary storage)

Labels Used

Label **O** Length = 287 Checksum = 55B5

Use the length (LN=) and Checksum (CK=) values to check if program was entered correctly. Use the sample computations to check proper operation after entry.

Flags Used

Flags 1 and 10 are used by this program. Flag 10 is set for this program, so that equations can be shown as prompts. Flag 1 is used to record the setting of Flag 10 before the program begins. At the end of the program, Flag 10 is reset to its original value, based on the value in Flag 1.