HP-35s Calculator Program Closure 4

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		

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		

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

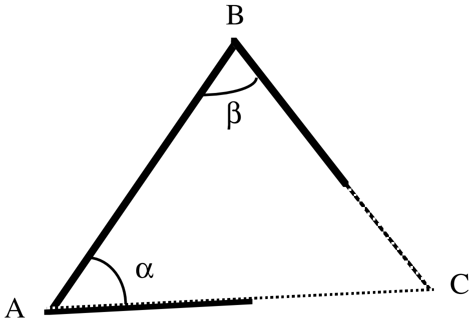
The intersection program is based on a triangle solution to the situation of knowing two angles and the included side. In this case, as co-ordinates are available, the solution can go directly to the co-ordinates of the unknown point.

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:

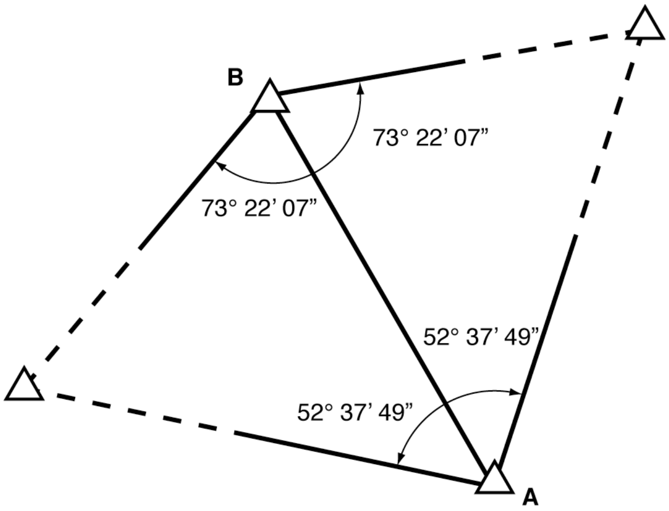
X_C = 16 313·13

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.