

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

Programmer: Dr. Bill Hazelton

Date: March, 2006.

Line	Instruction	Display	User Instructions
N0001	LBL N		
N0002	SF 10		→ FLAGS SF .0
N0003	ENTER X1		→ EQN RCL E RCL N etc. ENTER to end
N0004	PSE		
N0005	INPUT X		
N0006	ENTER Y1		→ EQN RCL E RCL N etc. ENTER to end
N0007	PSE		
N0008	INPUT Y		
N0009	RCL X		
N0010	STO A		
N0011	RCL Y		
N0012	STO B		
N0013	ENTER X2		→ EQN RCL E RCL N etc. ENTER to end
N0014	PSE		
N0015	INPUT X		
N0016	ENTER Y2		→ EQN RCL E RCL N etc. ENTER to end
N0017	PSE		
N0018	INPUT Y		
N0019	RCL X		
N0020	STO C		
N0021	RCL Y		
N0022	STO D		
N0023	ENTER ANG 1		→ EQN RCL E RCL N etc. ENTER to end
N0024	PSE		
N0025	INPUT E		
N0026	ENTER ANG 2		→ EQN RCL E RCL N etc. ENTER to end
N0027	PSE		
N0028	INPUT F		
N0029	RCL E		
N0030	→ HR		
N0031	STO E		
N0032	RCL F		
N0033	→ HR		
N0034	STO F		
N0035	RCL B		
N0036	RCL- D		
N0037	RCL A		
N0038	RCL F		
N0039	TAN		

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

N0040	÷	
N0041	+	
N0042	RCL C	
N0043	RCL E	
N0044	TAN	
N0045	÷	
N0046	+	
N0047	RCL E	
N0048	TAN	
N0049	1/x	
N0050	RCL F	
N0051	TAN	
N0052	1/x	
N0053	+	
N0054	STO G	
N0055	÷	
N0056	STO X	
N0057	X OF POINT	⇒ EQN RCL X SPACE etc. ENTER to end
N0058	PSE	
N0059	VIEW X	
N0060	RCL C	
N0061	RCL- A	
N0062	RCL B	
N0063	RCL F	
N0064	TAN	
N0065	÷	
N0066	+	
N0067	RCL D	
N0068	RCL E	
N0069	TAN	
N0070	÷	
N0071	+	
N0072	RCL G	
N0073	÷	
N0074	STO Y	
N0075	Y OF POINT	⇒ EQN RCL Y SPACE etc. ENTER to end
N0076	PSE	
N0077	VIEW Y	
N0078	RCL X	
N0079	RCL× G	
N0080	RCL D	
N0081	RCL- B	
N0082	2	
N0083	×	
N0084	+	
N0085	RCL÷ G	
N0086	STO U	

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

N0087	ALT X OF POINT		EQN RCL A RCL L etc. ENTER to end
N0088	PSE		
N0089	VIEW U		
N0090	RCL Y		
N0091	RCL× G		
N0092	RCL A		
N0093	RCL- C		
N0094	2		
N0095	×		
N0096	+		
N0097	RCL÷ G		
N0098	STO V		
N0099	ALT Y OF POINT		
N0100	PSE		
N0101	VIEW V		
N0102	GTO N		

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 N may be used, if necessary, without any adverse consequences to the program.

Co-ordinate-based Intersection Program (2-D)**Theory**

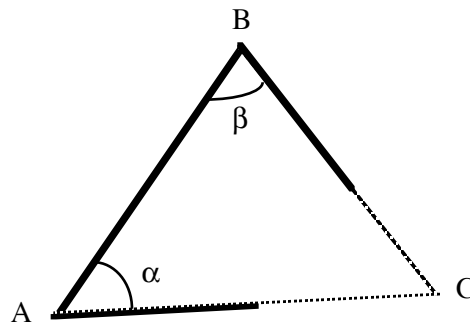
The intersection program is based on a triangle solution to the situation of knowing two angles and the included side. In this case, as co-ordinates are available, the solution can go directly to the co-ordinates of the unknown point.

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 β at A and 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 do a least squares adjustment in this case.

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.

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

Sample Computation 1

	X	Y	Base angle
A (1)	20 579.80	12 842.70	$\alpha (1) = 52^\circ 37' 49''$
B (2)	15 236.30	13 294.80	$\beta (2) = 73^\circ 22' 07''$

Initial Solution:

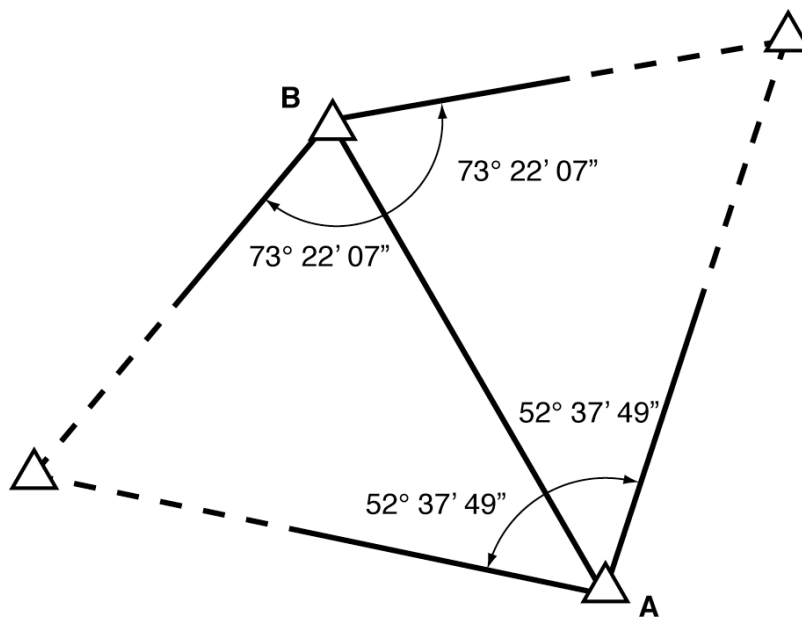
$$X_C = 16\,313.13$$

$$Y_C = 8\,138.18$$

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	$\beta (1) = 73^\circ 22' 07''$
A (2)	20 579.80	12 842.70	$\alpha (2) = 52^\circ 37' 49''$

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** $\cot \alpha + \cot \beta$
- 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 N Length = 432 Checksum = 5112

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 N.

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.

This is the end of the program, If you press the C key (lower left-hand corner), the calculator will stop the program and have the U and V values in the stack.

If you press R/S again, the program will return to the start and prompt for the next X1 co-ordinate.

To run additional intersected points from the same pair of known points, press C to get out of the program, press GTO .N0023 (note the dot before the N), then press R/S to start the program. The program then prompts for the first angle, with the co-ordinate of the two known points already in the correct storage registers.