

**Helmert Co-ordinate Transformation**

**Programmer:** Dr. Bill Hazelton

**Date:** May, 2005.

**Purpose:** The purpose of this collection of routines is to allow the user to transform co-ordinates from one co-ordinate system (which for convenience we'll call the 'old' system) to another co-ordinate system (which we'll call the 'new' system).

As the procedure is designed for surveyors, the transformation used is the standard Helmert transformation. This is the one most commonly used for transforming between plane co-ordinate systems. The procedure is also designed for 2-D transformations. A more complex procedure is required for 3-D transformations.

**Introduction**

The basic Helmert transformation is fairly straightforward and lends itself to using least squares adjustment to compute the transformation parameters. However, the HP-33S calculator is singularly ill-suited to least squares, by virtue of having just 26 addressable general registers and no way to work with matrices at a native level. So we will develop a simpler approach here.

Because of the nature of the process, it will be broken down into several steps, each with its own program or keystroke routine. This will also enable you to go to a later part of the process in the event that you know the transformation parameters already.

**Theory**

The Helmert transformation uses the formulae:

$$\begin{aligned}y_1 &= c_1 + x_1\lambda \cos \theta + x_2\lambda \sin \theta \\y_2 &= c_2 - x_1\lambda \sin \theta + x_2\lambda \cos \theta\end{aligned}$$

where  $y_1$  and  $y_2$  are the co-ordinates in the new co-ordinate system;

$x_1$  and  $x_2$  are the co-ordinates in the old co-ordinate system;

$c_1$  and  $c_2$  are the translation (shift) parameters;

$\lambda$  is the scale factor from old to new; and

$\theta$  is the amount of rotation from old to new.

This is also known by the ubiquitous  $\mathbf{y} = \mathbf{c} + \lambda \mathbf{R} \mathbf{x}$  equation.

If we replace  $\lambda \sin \theta$  by  $a$  and  $\lambda \cos \theta$  by  $b$ , we get a pair of linear equations which can be readily solved using least squares adjustment.

As least squares is not available in this case, we will obtain the various transformation parameters ( $c_1$ ,  $c_2$ ,  $\lambda$ ,  $\theta$ ) by other means.

**Helmert Co-ordinate Transformation****Algorithm**

1. Take a selection of points in the old co-ordinate system, noting which they are. Using the Utility 4 [Co-ordinate 'Inverse' or Bearing and Distance from Co-ordinates] keystroke routine (attached), compute the bearing and distance between various pairs of points. Points that are farther apart are better as this minimizes the effect of random errors. Record these values (on paper).
2. Taking the same pairs of points in the new co-ordinate system, use the Utility 4 keystroke routine to compute the bearings and distances between these pairs of points. Record these values.
3. For each pair of points, take the distance computed for the new co-ordinate system and divide it by the distance computed for the old co-ordinate system. Record these values and take the mean of them. This will be the scale factor,  $\lambda$ .
4. For each pair of points, take the bearing computed for the new co-ordinate system and subtract from it the bearing computed for the old co-ordinate system. Use the H.MS-program (Utility 2, attached), if that helps.
5. Use the first program below (Helmert 1) to compute the  $c_1$  and  $c_2$  parameters for each co-ordinate pair. The program will also compute the mean value of these parameters from all the co-ordinate pairs used.
6. Use the second program below (Helmert 2) to compute co-ordinates in the new system given co-ordinates in the old system and the transformation parameters.

**Notes**

If you already know the transformation parameters, you can go directly to the Helmert 2 program.

Once the rotation and scale factor parameters have been determined, the translations parameters can be easily computed using:

$$c_1 = y_1 - x_1\lambda \cos \theta - x_2\lambda \sin \theta$$

$$c_2 = y_2 + x_1\lambda \sin \theta - x_2\lambda \cos \theta$$

These are the core formulae for program Helmert 1.

The rotation value ( $\theta$ ) in both programs is entered in DDD.MMSS format (HP notation) and is converted by the program into decimal degrees and stored that way for use. However, if you use the program again for the same data, you will have to re-enter the rotation value, or else the program will convert it again, giving incorrect results.

**HP-33S Calculator Programs**  
**Helmert Co-ordinate Transformation**

**Worked Example 1** (using real data with errors)

<b>Intersection</b>	<b>X<sub>1</sub> (y)</b>	<b>X<sub>2</sub> (x)</b>	<b>Y<sub>1</sub> (N)</b>	<b>Y<sub>2</sub> (E)</b>
Oxford & Sanitorium	9.8	2.9	4757500	472700
Wonderland & Southdale	5.96	4.86	4753800	476500
Wharncliffe & Stanley	8.56	7.32	4758200	478600
Oxford & Wharncliffe	9.67	7.58	4759800	478400
Wellington & Southdale	4.9	8.24	4754300	481500
Highbury & Hamilton	6.67	11.32	4758200	484100
Trafalgar & Clarke	7.56	13.17	4759700	486200
Adelaide & Dundas	8.59	9.49	4759400	481100
Highbury & Fanshawe	12.68	11.5	4765400	481500
Richmond & Huron	10.73	8.28	4761500	478800

The above co-ordinate pairs were determined as follows. The old system co-ordinates ( $x_1, x_2$ ) were digitized from a paper map and are in the digitizing table's own co-ordinate system. The new system co-ordinates ( $y_1, y_2$ ) were measured in UTM co-ordinates from another source. Note that the UTM co-ordinates are rounded to the nearest 100 units, so we cannot expect results much better than 100-200 units for any co-ordinates produced by this system.

To simplify matters, the pairs of points taken were Oxford & Sanitorium and each of the other intersections. Distances were computed in each of the co-ordinate systems (new and old) and these are tabulated in the first two columns. The new distance is then divided by the old distance to give the scale factor for that point pair.

Each point pair has a bearing computed between the points in each of the co-ordinate systems. These are tabulated in the next two columns, and the difference between each pair of bearings is given in the final column. Note that all bearings are in HP notation (DDD.MMSS). The bearing difference is corrected to be in the range  $0^\circ$  to  $360^\circ$ .

<b>Intersection</b>	<b>New Distance</b>	<b>Old Distance</b>	<b>Scale Factor</b>	<b>New Bearing</b>	<b>Old Bearing</b>	<b>Difference</b>
Oxford & Sanitorium						
Wonderland & Southdale	5304	4.31	1230.21	134.1410	152.5734	341.1636
Wharncliffe & Stanley	5941	4.59	1294.24	83.1402	105.4016	337.3346
Oxford & Wharncliffe	6147	4.68	1312.86	68.0132	91.3528	336.2604
Wellington & Southdale	9364	7.25	1292.01	109.5859	132.3223	337.2637
Highbury & Hamilton	11421	8.98	1271.46	86.2910	110.2331	336.0540
Trafalgar & Clarke	13678	10.51	1301.26	80.4439	102.1815	338.2624
Adelaide & Dundas	8612	6.70	1285.37	77.1517	100.2415	336.5102
Highbury & Fanshawe	11826	9.07	1303.92	48.0505	71.2907	336.3559
Richmond & Huron	7295	5.46	1336.04	56.4444	80.1133	336.3311
<b>Means</b>			1291.93			337.2822

The mean scale factor and mean rotation are then determined. These compare pretty well with the values determined by a rigorous least squares adjustment of the same data (in this case 1308.77 for  $\lambda$  and  $338^\circ 05' 45''$  for  $\theta$ ). For this example, we will use the values determined by our own calculations.

**Helmert Co-ordinate Transformation**

Putting these values into the Helmert 1 program and running through all of the co-ordinates for the points given above, we obtain the following values for the translation parameters.

Intersection	c <sub>1</sub>	c <sub>2</sub>
Oxford & Sanitorium	4747240.55	464388.60
Wonderland & Southdale	4749093.16	467750.30
Wharncliffe & Stanley	4751608.06	465627.74
Oxford & Wharncliffe	4752012.13	464568.05
Wellington & Southdale	4752531.10	469241.44
Highbury & Hamilton	4755843.37	467289.82
Trafalgar & Clarke	4757196.97	466741.59
Adelaide & Dundas	4753846.34	465523.32
Highbury & Fanshawe	4755960.41	461500.26
Richmond & Huron	4752793.66	463608.04
Mean	4752812.58	465623.92

We can compare these with the values from the rigorous least squares adjustment, which were:

$$c_1 = 4744310.55 \quad \text{and} \quad c_2 = 473815.78$$

[We would expect some differences here, because we have use just some of the available combinations, while the least squares adjustment will use all possible data to derive the most probable solution. But this should not be a problem in this case, where we are dealing with low quality data.]

We now have the four parameters required for the Helmert transformation, so we can go ahead and run the various values through the Helmert 2 program. For this example, we will take the four parameters we have calculated and use them to compute the new co-ordinates (UTM) for each of the intersections, using the old (digitizer) co-ordinates. We can then compare the computed new co-ordinates with the given new co-ordinates and determine residuals for this case.

Intersection	Y <sub>1</sub> (meas)	Y <sub>2</sub> (meas)	Y <sub>1</sub> (comp)	Y <sub>2</sub> (comp)	ΔY <sub>1</sub>	ΔY <sub>2</sub>	Dist.
Oxford & Sanitorium	4757500	472700	4763072	473935	-5572	-1235	5707.2
Wonderland & Southdale	4753800	476500	4757519	474374	-3719	2126	4283.7
Wharncliffe & Stanley	4758200	478600	4759405	478596	-1205	4	1205.0
Oxford & Wharncliffe	4759800	478400	4760600	479456	-800	-1056	1324.8
Wellington & Southdale	4754300	481500	4754581	477882	-281	3618	3628.8
Highbury & Hamilton	4758200	484100	4755169	482434	3031	1666	3458.6
Trafalgar & Clarke	4759700	486200	4755316	485082	4384	1118	4524.3
Adelaide & Dundas	4759400	481100	4758366	481201	1034	-101	1038.9
Highbury & Fanshawe	4765400	481500	4762252	485624	3148	-4124	5188.1
Richmond & Huron	4761500	478800	4761519	480816	-19	-2016	2016.0

The mean distance by which the computed co-ordinates miss the measured co-ordinates is 3238 units. This is quite large. The least squares mean difference was 142 units. This indicates that the more comprehensive the adjustment of the data, the better the result should be. It also indicates that the program here is satisfactory, but not with poor data, such as that used here (this data set was selected because of its known low quality).

**Helmert Co-ordinate Transformation****Worked Example 2** (using artificial data without errors)

We have the following data set.

Point	Old X	Old Y	New X	New Y
A	10	10	100.00	100.00
B	20	10	114.14	114.14
C	20	20	100.00	128.28
D	10	20	85.86	114.14

Going through the same process as above, we determine the various distances and bearings between the various points.

Line	New Distance	Old Distance	Scale Factor	New Bearing	Old Bearing	Difference
AB	20	10	2.00	45.0000	90.0000	315.0000
BC	20	10	2.00	315.0000	0.0000	315.0000
CD	20	10	2.00	225.0000	270.0000	315.0000
DA	20	10	2.00	135.0000	180.0000	315.0000

We find that the rotation value ( $\theta$ ) is  $315^\circ 00' 00''$ , while the scale factor ( $\lambda$ ) is 2.00. These values can be used in the Helmert 1 program to find the translations parameters.

Point	$c_1$	$c_2$
A	100.00	71.72
B	100.00	71.71
C	100.00	71.71
D	100.00	71.71

The mean parameters (from the calculator program) are

$$c_1 = 100.000$$

$$c_2 = 71.714$$

We can then use these four values in the Helmert 2 program, compute the new co-ordinates of the four points from the old co-ordinates and see if there are any differences.

Point	Old X	Old Y	New X (Comp)	New Y (Comp)	$\Delta X$	$\Delta Y$
A	10	10	100.000	99.998	0.000	-0.002
B	20	10	114.142	114.140	0.002	0.000
C	20	20	100.000	128.283	0.000	0.003
D	10	20	85.858	114.140	-0.002	0.000

It can be seen that the program returns values that are at least as good as the inputs, in terms of the precision, i.e., the inputs were given to 0.01, while the returned values are good to within 0.003.

**Helmert Co-ordinate Transformation**

HP-33S Calculator Program

Helmert

1

**Computation of Helmert Translation Parameters**

Programmer: Dr. Bill Hazelton

Date: May, 2005.

Line	Instruction	Display	User Instructions
Q0001	LBL Q		
Q0002	INPUT T		Enter rotation value ( $\theta$ ) in D.MMSS
Q0003	INPUT L		Enter scale factor ( $\lambda$ )
Q0004	RCL T		
Q0005	$\rightarrow$ HR		
Q0006	STO T		
Q0007	COS		
Q0008	$\times$		
Q0009	STO A		
Q0010	RCL T		
Q0011	SIN		
Q0012	RCL L		
Q0013	$\times$		
Q0014	STO B		
Q0015	0		
Q0016	STO i		
Q0017	STO E		
Q0018	STO F		
I0001	LBL I		
I0002	INPUT X		Enter old X co-ordinate
I0003	-99		(if value is -99, assumes end of data)
I0004	RCL X		
I0005	$x = y ?$		
I0006	GTO J		
I0007	INPUT Y		Enter old Y co-ordinate
I0008	INPUT P		Enter new X co-ordinate
I0009	INPUT Q		Enter new Y co-ordinate
I0010	RCL P		
I0011	RCL A		
I0012	RCL X		
I0013	$\times$		
I0014	-		
I0015	RCL B		
I0016	RCL Y		
I0017	$\times$		
I0018	-		
I0019	STO C		
I0020	STO+ E		
I0021	RCL Q		

**Helmert Co-ordinate Transformation**

Line	Instruction	Display	User Instructions
I0022	RCL B		
I0023	RCL X		
I0024	×		
I0025	+		
I0026	RCL A		
I0027	RCL Y		
I0028	×		
I0029	–		
I0030	STO D		
I0031	STO+ F		
I0032	1		
I0033	STO+ i		
I0034	VIEW C		
I0035	VIEW D		
I0036	GTO I		
J0001	LBL J		
J0002	RCL E		
J0003	RCL i		
J0004	÷		
J0005	STO E		
J0006	VIEW E		
J0007	RCL F		
J0008	RCL i		
J0009	÷		
J0010	STO F		
J0011	VIEW F		
J0012	RTN		

Current  $c_1$  value shown. Press R/S  
 Current  $c_2$  value shown. Press R/S

Mean  $c_1$  parameter value shown. Press R/S

Mean  $c_2$  parameter value shown.

**Notes**

Program is used to compute the mean  $c_1$  and  $c_2$  parameters (the translation parameters) for a simple Helmert transformation, assuming that the scale factor ( $\lambda$ ) and rotation ( $\theta$ ) for the transformation are known.

Rotation value ( $\theta$ ) is entered in DDD.MMSS (HP notation) and converted to decimal degrees by the program. If you run the program a second time with the same data, the rotation value will need to be re-entered, or else it will be converted a second time by the program, which will give erroneous results.

For an example of this program’s use, see the accompanying discussion on the Helmert Transformation in these notes.

**Labels Used**

- Label **Q**      Length = 66              Checksum = 0C22
- Label **I**      Length = 132              Checksum = 790B
- Label **J**      Length = 36              Checksum = 5EE4

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

**Helmert Co-ordinate Transformation****Storage Registers Used**

<b>A</b>	$\lambda \cos \theta$
<b>B</b>	$\lambda \sin \theta$
<b>C</b>	Current $c_1$ value
<b>D</b>	Current $c_2$ value
<b>E</b>	Accumulated $c_1$ values, then mean $c_1$ value
<b>F</b>	Accumulated $c_2$ values, then mean $c_2$ value
<b>L</b>	Scale factor ( $\lambda$ )
<b>P</b>	New co-ordinate system x value
<b>Q</b>	New co-ordinate system y value
<b>T</b>	Rotation parameter ( $\theta$ )
<b>X</b>	Old co-ordinate system x value
<b>Y</b>	Old co-ordinate system y value

**Operation**

1. Begin by pressing XEQ Q.
2. The calculator asks for an input for T, the rotation parameter ( $\theta$ ). Enter this in DDD.MMSS format (HP notation). Press R/S.
3. The calculator asks for an input for L, the scale factor ( $\lambda$ ). Enter this value. Press R/S.
4. The calculator asks for an input for X, the x co-ordinate in the old system ( $x_1$ ). Enter this. Press R/S. If you want to end the program, enter  $-99$  as the input. If you have an actual value of  $-99$  to be entered, use  $-99.00001$  instead. If you entered  $-99$ , the calculator goes to step 11.
5. The calculator asks for an input for Y, the y co-ordinate in the old system ( $x_2$ ). Enter this. Press R/S.
6. The calculator asks for an input for P, the x co-ordinate in the new system ( $y_1$ ). Enter this. Press R/S.
7. The calculator asks for an input for Q, the y co-ordinate in the new system ( $y_2$ ). Enter this. Press R/S.
8. The calculator returns a value for C, the  $c_1$  translation parameter. Press R/S.
9. The calculator returns a value for D, the  $c_2$  translation parameter. Press R/S.
10. The calculator returns to step 4.
11. The calculator displays a value for E, the mean of all values computed for the  $c_1$  parameter. Press R/S.
12. The calculator displays a value for F, the mean of all values computed for the  $c_2$  parameter. Press C to exit the program.

**Helmert Co-ordinate Transformation**

HP-33S Calculator Program

Helmert

2

**Co-ordinate Transformation Using 2-D Helmert Transformation**

Programmer: Dr. Bill Hazelton

Date: May, 2005.

Line	Instruction	Display	User Instructions
R0001	LBL R		
R0002	INPUT T		Enter rotation value ( $\theta$ ) in D.MMSS. Press R/S
R0003	INPUT L		Enter scale factor ( $\lambda$ ). Press R/S
R0004	INPUT C		Enter $c_1$ value. Press R/S
R0005	INPUT D		Enter $c_2$ value. Press R/S
R0006	RCL T		
R0007	$\rightarrow$ HR		
R0008	STO T		
R0009	COS		
R0010	RCL L		
R0011	$\times$		
R0012	STO A		
R0013	RCL T		
R0014	SIN		
R0015	RCL L		
R0016	$\times$		
R0017	STO B		
F0001	LBL F		
F0002	INPUT X		Enter old X co-ordinate. Press R/S
F0003	INPUT Y		Enter old Y co-ordinate. Press R/S
F0004	RCL X		
F0005	RCL A		
F0006	$\times$		
F0007	RCL B		
F0008	RCL Y		
F0009	$\times$		
F0010	+		
F0011	RCL C		
F0012	+		
F0013	STO P		
F0014	RCL D		
F0015	RCL B		
F0016	RCL X		
F0017	$\times$		
F0018	-		
F0019	RCL A		
F0020	RCL Y		
F0021	$\times$		

**Helmert Co-ordinate Transformation**

Line	Instruction	Display	User Instructions
F0022	+		New X co-ordinate displayed. Press R/S New Y co-ordinate displayed. Press R/S
F0023	STO Q		
F0024	VIEW P		
F0025	VIEW Q		
F0026	GTO F		

**Notes**

Program is used to compute the new co-ordinates for a simple 2-D Helmert transformation, assuming that the translation parameters ( $c_1$ ,  $c_2$ ), the scale factor ( $\lambda$ ) and the rotation ( $\theta$ ) for the transformation are known.

Rotation value ( $\theta$ ) is entered in DDD.MMSS (HP notation) and converted to decimal degrees by the program. If you run the program from the start a second time with the same data, you will have to re-enter the rotation value, or else it will be re-converted and produce incorrect results.

For an example of this program's use, see the accompanying discussion on the Helmert Transformation in these notes.

**Operation**

1. Start the program by pressing XEQ R.
2. The calculator asks for an input for T, the rotation parameter ( $\theta$ ). Enter this in DDD.MMSS format (HP notation). Press R/S.
3. The calculator asks for an input for L, the scale factor ( $\lambda$ ). Enter this value. Press R/S.
4. The calculator asks for an input value for C, the  $c_1$  translation parameter. Enter this value. Press R/S.
5. The calculator asks for an input value for D, the  $c_2$  translation parameter. Enter this value. Press R/S.
6. The calculator asks for an input value for X, the x co-ordinate in the old system ( $x_1$ ). Enter this value. Press R/S.
7. The calculator asks for an input value for Y, the y co-ordinate in the old system ( $x_2$ ). Enter this value. Press R/S.
8. The calculator displays the value for P, the x co-ordinate in the new system ( $y_1$ ). Press R/S.
9. The calculator displays the value for Q, the y co-ordinate in the new system ( $y_2$ ). Press R/S.
10. The calculator returns to step 6 and asks for the next set of input co-ordinates. To end the program, press C.

**Helmert Co-ordinate Transformation****Labels Used**

Label **R**            Length = 51            Checksum = CC2F

Label **F**            Length = 78            Checksum = 3B5B

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

**Storage Registers Used**

**A**       $\lambda \cos \theta$

**B**       $\lambda \sin \theta$

**C**       $c_1$

**D**       $c_2$

**L**      Scale factor ( $\lambda$ )

**P**      New co-ordinate system x value

**Q**      New co-ordinate system y value

**T**      Rotation parameter ( $\theta$ )

**X**      Old co-ordinate system x value

**Y**      Old co-ordinate system y value

**Co-ordinate ‘Inverse’  
or Bearing and Distance from Co-ordinates**

Programmer: Dr. Bill Hazelton

Date: March, 2005.

This particular key sequence is so simple that it doesn’t deserve a program to itself. Most of the keystrokes you have to do yourself to enter the data, so you might as well just know them and do it. If you feel that it must be a program, then add a LBL to the top and a RTN to the end, as well as suitable INPUT or STOP lines, and you’ll have it. But I use it all the time without a program.

You want to get the bearing and distance between two points, whose co-ordinates are known. We want the bearing from the near point (1) to the far point (2). The co-ordinates of the far point are: (E2, N2), and the co-ordinates of the near point are (E1, N1).

Keystrokes	Data Entry	Display	User Instructions
 CLEAR $\Sigma(4)$			Clear the statistical registers
	E2		Enter the far point’s Easting
ENTER			Press Enter
	N2		Enter the far point’s Northing
$\Sigma+$		1.0000	Press the summation + key
	E1		Enter the near point’s Easting
ENTER			Press Enter
	N2		Enter the near point’s Northing
 $\Sigma-$		0.0000	Press the summation – key
 SUMS $\Sigma y$			Recall the Y sum to the stack
 SUMS $\Sigma x$			Recall the X sum to the stack
 $\rightarrow \theta, r$			Convert from rectangular to polar ***
x <> y			Swap the X and Y registers
 $\rightarrow HMS$			Convert bearing to degrees, minutes, seconds

**Notes**

- (1) General keystroke sequence to obtain bearing and distance between two points, in the order discussed above.
- (2) Go through the keystroke sequence. At the point marked \*\*\* (after the conversion to polar), the stack will contain:

Stack Register	Contents
T	
Z	
Y	Bearing of line in decimal degrees
X	Distance of line

**Helmert Co-ordinate Transformation**

If you need the bearing in degrees, minutes and seconds, swap the X and Y registers and convert the bearing using  $\rightarrow$  HMS. (as shown in the last two steps)

- (3) If the bearing is negative, swap it into the X register, add 360 to it, and convert to degrees, minutes and seconds if desired.

**Sample Computation**

Clear the statistical registers: Press  CLEAR  $\Sigma$  (4)

Enter far point's co-ordinates: E2 205 123.456 Press Enter

N2 123 456.789 Press  $\Sigma+$

Enter near point's co-ordinates: E1 206 654.321 Press Enter

E1 132 654.987 Press   $\Sigma-$

Recall  $\Sigma y$  to the stack  SUMS  $\Sigma y$

Recall  $\Sigma x$  to the stack  SUMS  $\Sigma x$

Stack will now contain: -1,530.865 in the Y register

-9,198.198 in the X register

Convert to polar   $\rightarrow \theta, r$

Stack will now contain: -170.5508178 in the Y register (bearing, dec. degrees)

9,324.719518 in the X register (distance)

You now have the length of the line (9324.720). To get the bearing in DMS, swap the X and Y registers using the  $x \leftrightarrow y$  key. As the bearing is negative, key in 360, press +. The bearing is now 189.4491822 in decimal degrees.

To convert to degrees, minutes and seconds, press   $\rightarrow$ HMS .

The bearing now reads 189.2657056, which is 189° 26' 57.056" (if you need it to that level of precision!)

**Storage Registers Used**

None, except:

Statistical Registers:  $\Sigma x = Y$  or N co-ordinates, or  $\Delta Y$  or  $\Delta N$

$\Sigma y = X$  or E co-ordinates, or  $\Delta X$  or  $\Delta E$

**Labels Used**

None

**Helmert Co-ordinate Transformation****HP-33S Calculator Program (Appendix 2)**

Utility 2

**HMS-**

Programmer: Dr. Bill Hazelton

Date: March, 2005.

Line	Instruction	Display	User Instructions
E0001	LBL E		Enter first angle. Press ENTER.
E0002	→HR		
E0003	x < > y		Enter second angle. Press XEQ E.
E0004	→HR		
E0005	x < > y		(Angles in DDD.MMSS format)
E0006	-		
E0007	→HMS		
E0008	RTN		Angle sum displayed (in HP notation)

**Notes**

- (1) General program to get the difference between two angles, bearings 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
X	Second angle in DMS

Press XEQ E. 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**

$$321^{\circ} 54' 32'' - 123^{\circ} 45' 56'' = 198^{\circ} 08' 36''$$

**Storage Registers Used**

None

**Labels Used**

Label E Length = 24 Checksum = 7DAF

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.

**Helmert Co-ordinate Transformation**