Least Squares Program Report

Craig Ferguson

This report relates to a program developed to minimize the residuals in a given system, through the use of Parametric least squares:

“

The Least Squares Principle states that the solution to an overdetermined problem should be found in such a manner that the weighted sum of the squares of the discrepancies between the observed values and most probable values should be a minimum.”

(UCT Geomatics 2 Notes)

The goal of this exercise, was to develop a single application that could solve any surveying related calculation task, whether resection, intersection, traverse, network etc. And minimize the error associated with the provisional results through use of the least squares principal. This was successfully achieved and this report will briefly explore this program.

PROGRAM Backend development:

- Used network X for ordering observations
- Created point class as base
- Had Points() , SurveyData() dictionaries
- SurveyData() was dictionary of station(), station() was dictionary of target()
- Target() has type, distance or direction, then variables to hold these values.
- Control Points and Porvisional points were kept in separate dictionaries, and points keys could be searched in each for creating the A matrix
- Used Numpy matix and operations for all matrix/vector data handling
- Used networkX to view geometry of observations
- Provisional coordinates for all types automatically calculated

Testing

- Tested with 7 data sets:
    - Resection
    - Intersection
    - Traverse
    - Combined
    - Triangulation
    - Complex Traverse
    - Simple Traverse

All tests past using results obtained from Geomatics 2 Course notes(Solutions to all data sets using the same code, regardless of observation type)

Designing the Interface

PyQT was used for Gui development.

Ref : http://www.riverbankcomputing.co.uk/software/pyqt/intro

A pyqt interface named qtdesigner was used to increase the efficiency of the GUI development through a drag and drop user interface designed with GUI creation in mind. This gui could then be saved as a .ui file, converted to .py using pyuic4.bat file shipped with the pyqt library. One need only run the pyuic4.bat file from within the command line, and navigate to the .ui file, and specify and output name to convert to .py.

This could be handled automatically by creating a .bat windows file, which was programmed to run on the call to main from within the eclipse development environment. This made it possible to update the interface in qtdesigner, and simply run the program in eclipse to convert and utilize the updated gui.

All connections, and code can be found with the program .py files which are also attached.

Applications

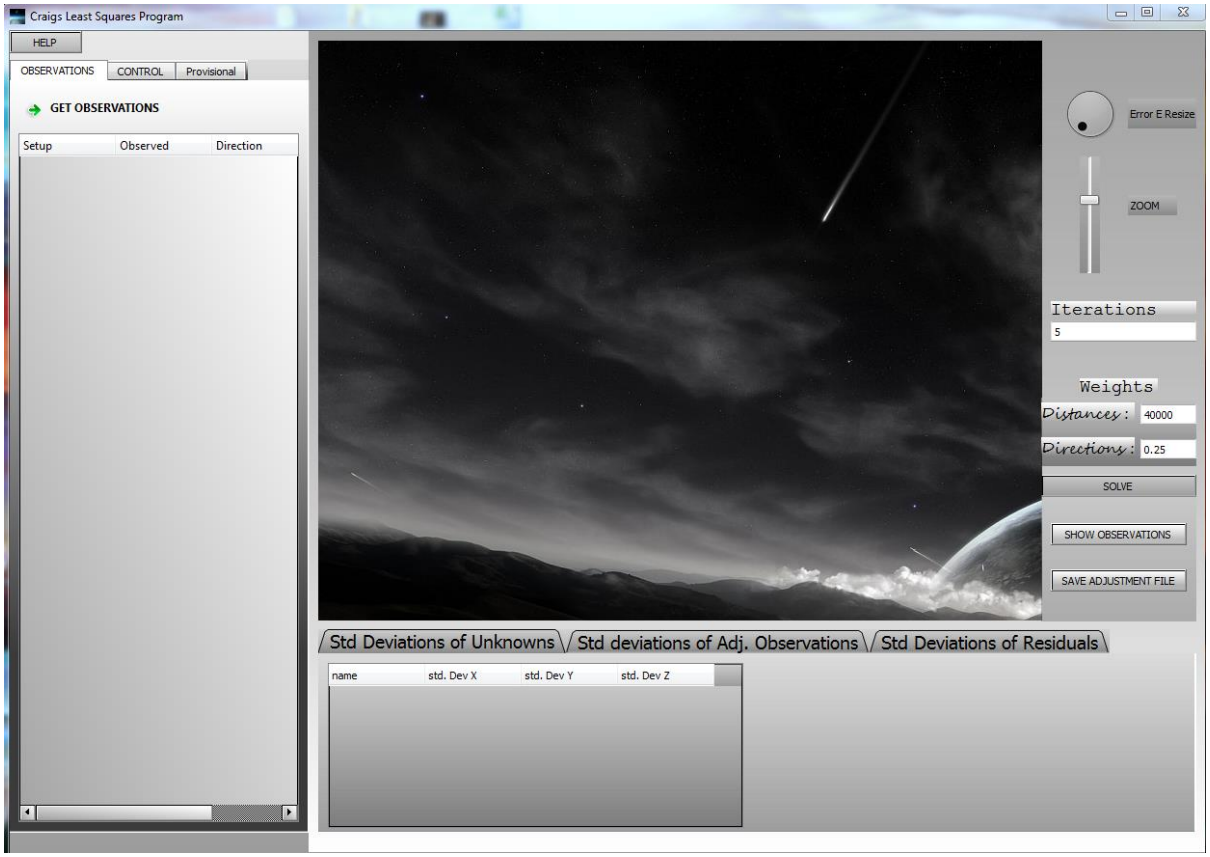
Along with the solution of least squares, this program features very useful functionality…

- Printout
  printout final coordinates to excel or text file
  printout full calculation process, with matrices, and residuals per iteration to text file
  printout a data traverse bowditch adjustment from provisional coordinates, where weighting is of bowditch distribution is based on distance of the leg
- Visualize
  See updates to observation exclusion in real time on the graphic interface
  Observe error ellipses, and zoom in and out of environment for a better understanding of context
  Increase the size of error ellipses to better view the shape
- Error
  Observe covariance matrix of unknowns, precisions of unknowns, precisions of observations , and the std. deviation of adjusted observations.
- Checks
  Inherent global checks, and numerical checks, to give early warning signs when there is a problem with the data.
  VtPV value updated in realtime

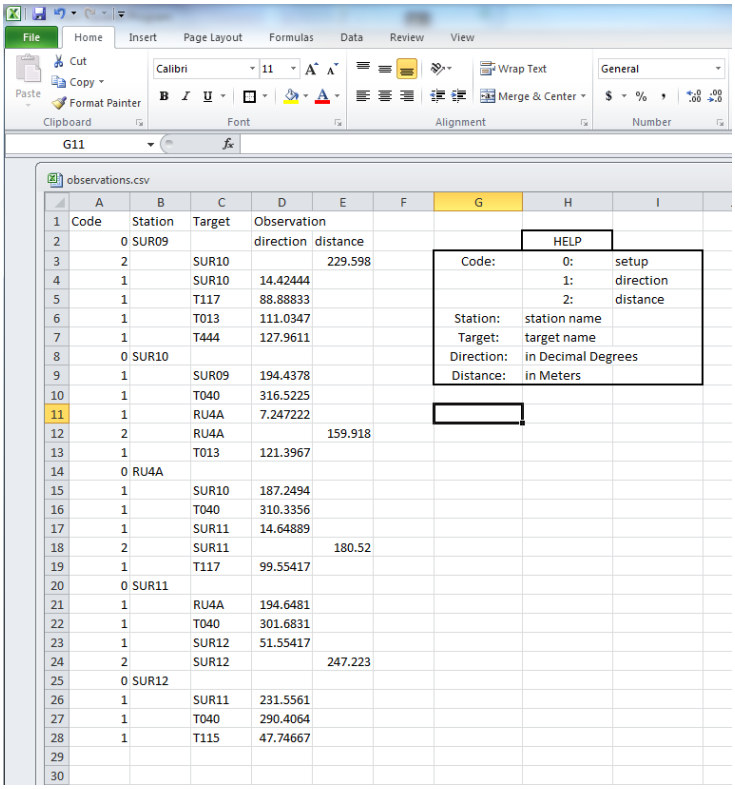
Usage

In order of procedure:

1) Welcome to the interface



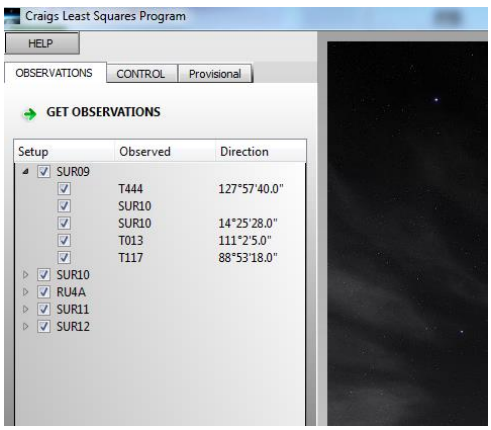
2) Open Observations file (make sure it's in the format specified)



observations.csv

	A	B	C	D	E	F	G	H	I
1	Code	Station	Target	Observation					
2		0	SUR09		direction	distance		HELP	
3		2		SUR10		229.598	Code:	0:	setup
4		1		SUR10	14.42444			1:	direction
5		1		T117	88.88833			2:	distance
6		1		T013	111.0347		Station:	station name	
7		1		T444	127.9611		Target:	target name	
8		0	SUR10				Direction:	in Decimal Degrees	
9		1		SUR09	194.4378		Distance:	in Meters	
10		1		T040	316.5225				
11		1		RU4A	7.247222				
12		2		RU4A		159.918			
13		1		T013	121.3967				
14		0	RU4A						
15		1		SUR10	187.2494				
16		1		T040	310.3356				
17		1		SUR11	14.64889				
18		2		SUR11		180.52			
19		1		T117	99.55417				
20		0	SUR11						
21		1		RU4A	194.6481				
22		1		T040	301.6831				
23		1		SUR12	51.55417				
24		2		SUR12		247.223			
25		0	SUR12						
26		1		SUR11	231.5561				
27		1		T040	290.4064				
28		1		T115	47.74667				
29									
30									

You should see a tree of Data:

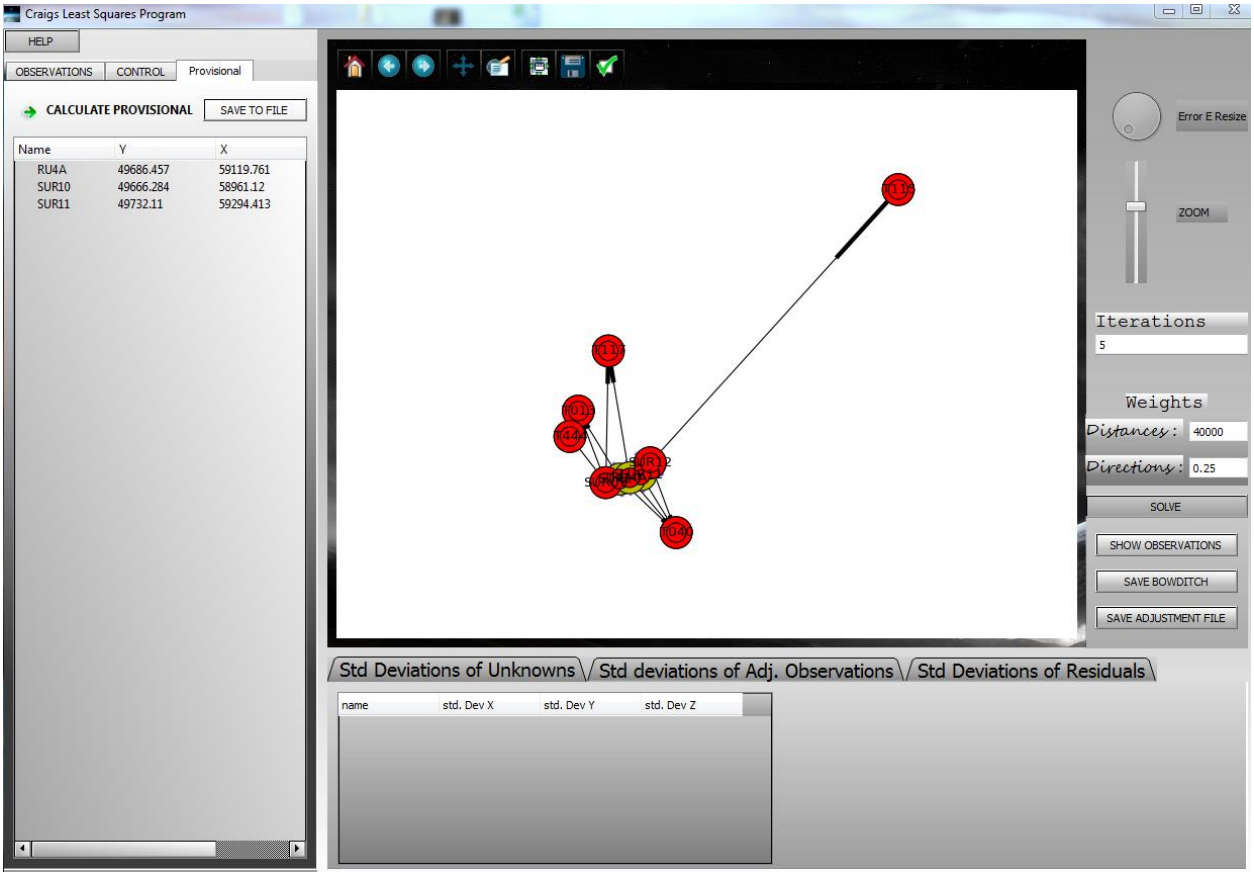


3) Do the same for control file (take note of the format below, and the column of “knowns” = 1)

control.csv

	A	B	C	D	E	F	G
1	Name	Y	X	H		known	
2	T117	51711.35	58779.23	0		1	
3	T013	50750.11	58299.8	0		1	
4	T115	54285.72	63409.75	0		1	
5	T040	48811.38	59862.58	0		1	
6	T444	50345.4	58164.13	0		1	
7	SUR09	49609.09	58738.76	0		1	
8	SUR12	49925.71	59448.14	0		1	
9	T119	52951.68	60807.08	0		1	
10	T381	49125.53	58438.66	0		1	
11							

4) You can now click Provisional tab and click “calculate the provisional”.
This will bring up the preview screen, where you have a visual representation of the data files you selected.



5) Weights:

Change weights to Taste (remember: using the inverse square of the precisions).

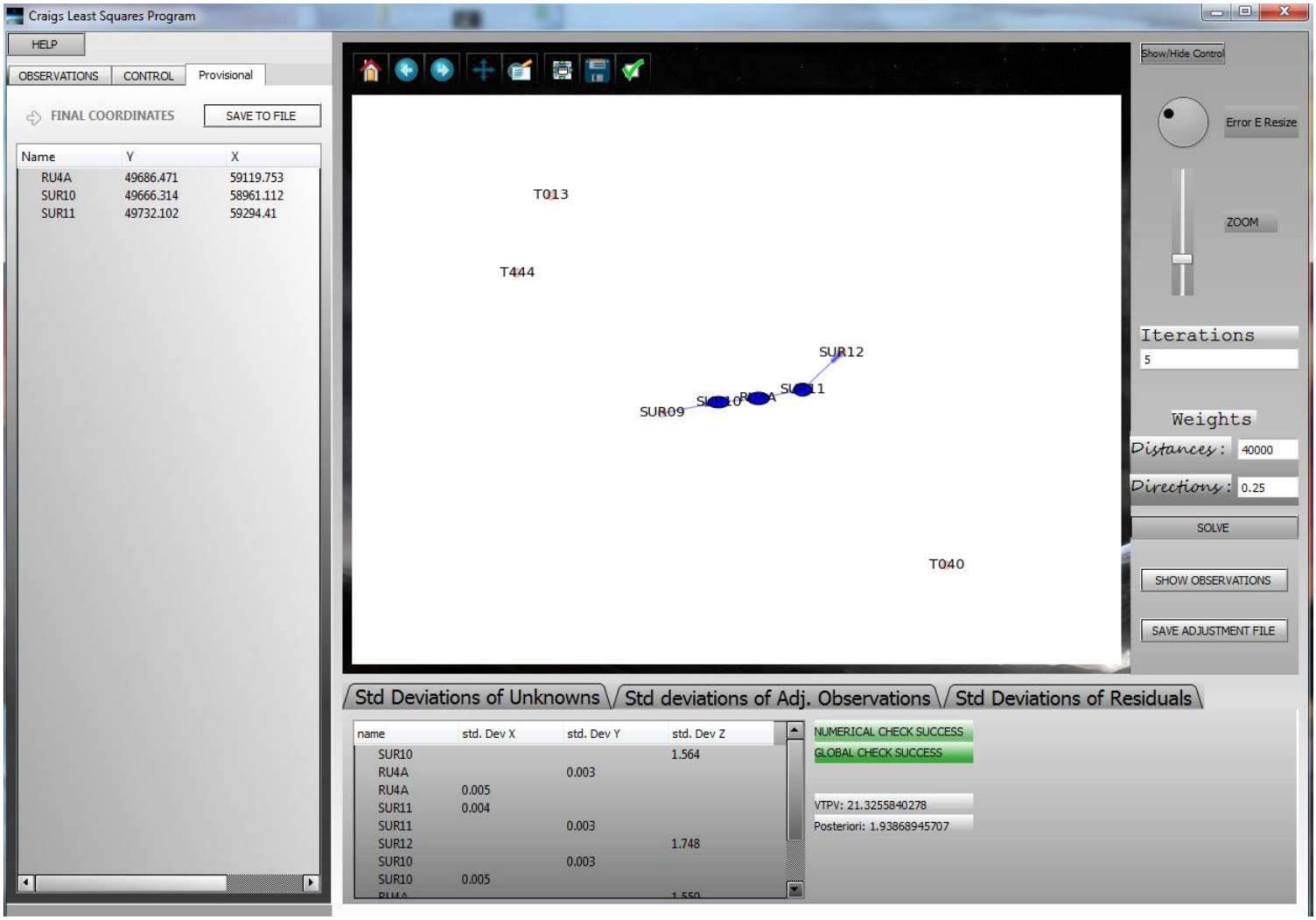
Defaults values are set to use the estimated precision for the orienting directions as 2"(The corresponding weights are 0.25) and for the distances, the estimated precision is 5mm, which corresponds to a weight of 40000.

Iterations:

Set as much iteration as you like, and be sure to watch the VTPV value for the specific iteration you choose. This will be handled effectively in a future update.

Solve:

Click solve and you are presented with your unknowns positions, and the error ellipse corresponding to the std. dev in x and y



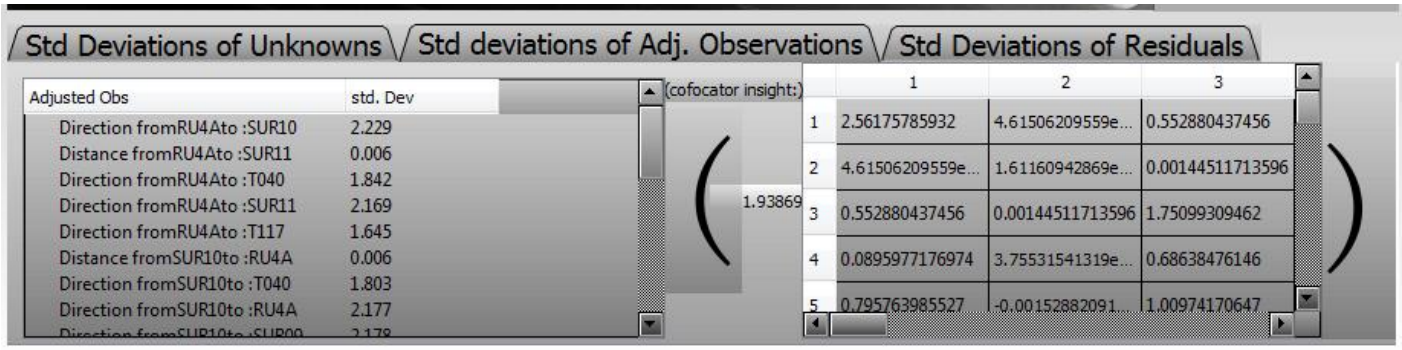
6) Std. Deviations Tabs

These tabs give insight into precisions of unknowns, std. dev. of observations, and precisions of observations respectively.

Distances are in (m) units

Directions are in seconds (“)

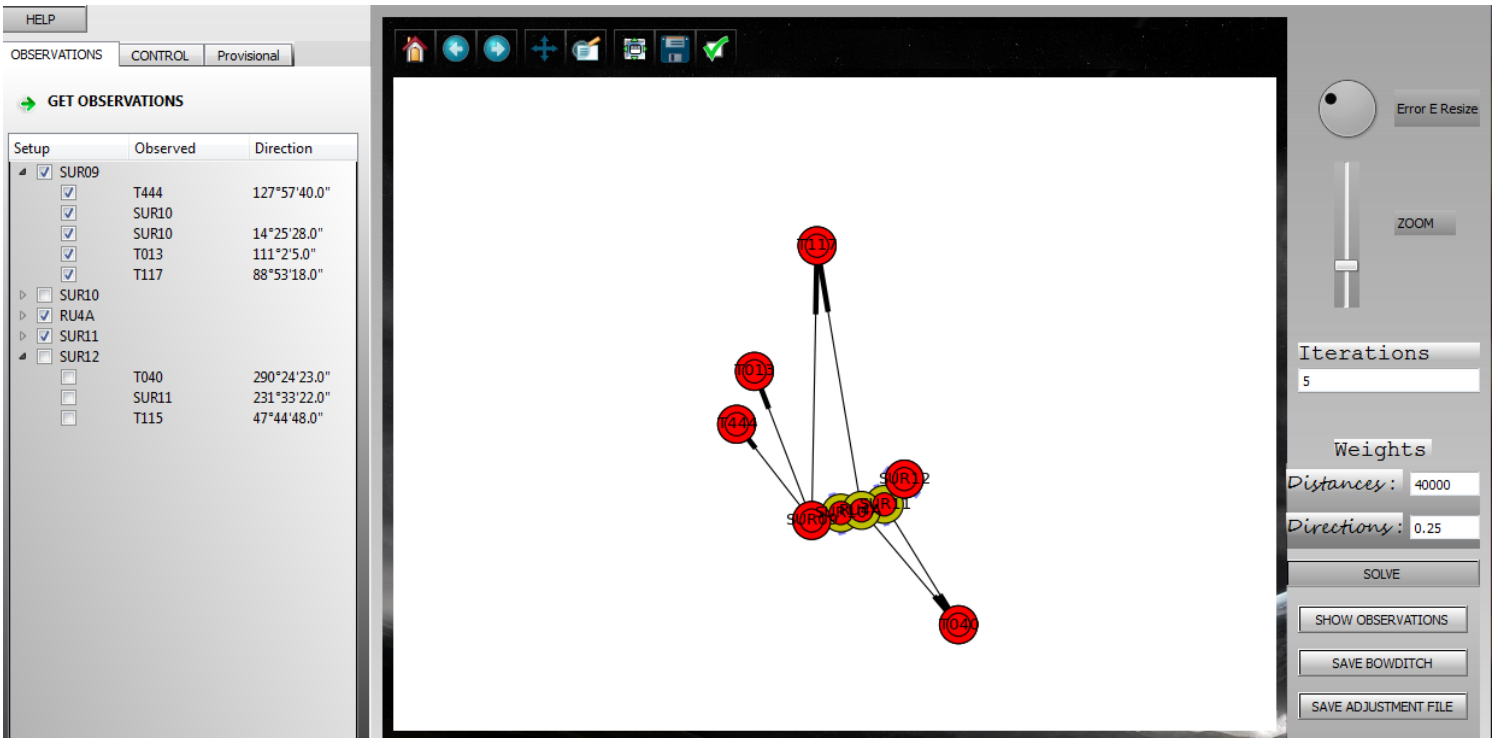
The Cofactor Matrix is presented on the right, with the posteriori outside the matrix as a constant.



7) Click “Show Observations”

You are now free to deselect observations as they are automatically updated in the provisional view tab.

Once you are happy with the chosen observations, click “solve” again, and the adjustment will be recalculated using only the new chosen observations.



Conclusion

As an exercise, I have thoroughly enjoyed the building of this application, and have learnt invaluable techniques that will certainly help me achieve similar results in less time in the future. The principals used to develop, design and manage this application can be applied to many areas in the profession of Geomatic's.

I will continue to improve on this program, and perhaps make it available as a tool for others to use and enjoy sometime in the future.

Craig Ferguson

Author, Designer, Developer