APG3013F

PARAMETRIC LEAST SQUARES ADJUSTMENT OF A TRAVERSE ANALYSIS

Lydia John

JUNE 2017

Contents

1 Introduction 3

1.1 About the assignment 3

1.1.1 Structure of the report 3

1.1.2 Aim of the assignment 3

1.1.3 Duration of the assignment 3

1.2 The setting. 3

1.3 The problem 3

1.4 Steps to solving the problem 4

2 Traverse. 5

2.1 Background. 5

2.2 Pre-analysis. 5

2.3 Arranging data. 6

3 Parametric Adjustment. 7

3.1 Setting up network. 7

3.2 Network details. 7

3.3 Functional model. 8

3.4 Least squares criteria. 9

3.5 Observation equations. 9

3.6 Matrices. 9

4 Conclusion 12

4.1 Overview 12

4.2 Evaluation 12

Chapter 1

Introduction

1.1 About the assignment

The assignment is based on the adjustment by parametric least squares and includes the adjusted problem as well as a thorough analysis of the results and the quality of the adjustment.

1.1.1 Structure of the report

This chapter introduces the assignment at hand and the problems that have to be solved. The following chapters will demonstrate how the actual problem is solved with each chapter dealing with the analysis of different problems. Conclusions are then drawn about the outcomes of the assignment and additional problems encountered throughout the duration.

1.1.2 Aim of the assignment

The aim was to understand the effect of observations and understanding how to set up and analyze an adjustment by coding for a parametric least squares adjustment of a traverse.

1.1.3 Duration of the assignment

8 May 2017 – 25 June 2017

1.2 The setting

The assignment is to code for the adjustment of a traverse of varying complexity, and to include orientation corrections at each set up.

1.3 The problem

The traverse route varies depending on the observations and orientations. The reliability of a traverse calculation is very weak due to few redundant measurements and gross errors that are not easily detectable.

1.4 Steps to solving the problem

Based on my knowledge and based on the requirements of the assignment, the steps to solving the problem by code are as follows:

1. Read a point and observation file into the code.
2. Appropriately restructure the data by appending it to a network
3. Adjust the traverse by least squares
4. Fit results to ellipses
5. Analyze results and perform statistical and mathematical tests to come to a conclusion on the quality of the traverse and whether gross errors have been accounted for or not.

Chapter 2

Traverse

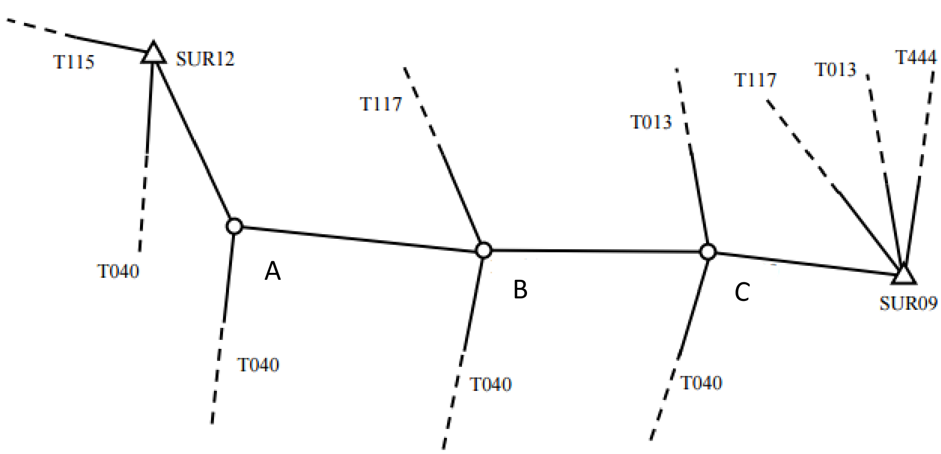
2.1 Background

Traverses are carried out between two known points which fix the traverse in space. The traverse is essentially made up of polars to unknown points using distance and direction observations. Additional orienting rays at each set up are used to increase redundancy and improve the accuracy of the traverse. The adjustment of the traverse relies heavily on the calculation of the initial coordinates of the unknown points therefore errors in this calculation will affect the results of the adjustment.

2.2 Pre-analysis

For the traverse the pre-analysis was necessary to distinguish orientation rays from the actual traverse route observations. All the observations have to be entangled so that they are equally adjusted. Orientation observations are only angles to trigs or other fixed points. The orientation rays need to be added to each set up in the traverse before calculating the polar to get initial coordinates to correct for the direction misclosure and ensure the best possible estimate. The adjusted equations are essentially distance and direction equations

The traverse network used for this example is as follows



2.3 Arranging data

The data is given in two CSV files, namely the point file and the observation file, which are read into the code. The observation file contains observations stored by the source, target, the distance observed, the direction observed in degrees, minutes and seconds. The point file contains the point name, the y coordinate, the x coordinate and a code to say whether a point is fixed,1, or free,0. Free points are the points for which we are calculating the coordinates and fixed points are the trig beacons.

Chapter 3

Parametric Adjustment

3.1 Setting up the network

I chose to arrange the traverse data in a network using Networkx in Python using a DiGraph. We were initially advised to use a MultiDiGraph since the traverse can go in either direction, but I encountered numerous problems as with it as I was unfamiliar with the functions of the graph and found that using the DiGraph worked just fine for me.

3.2 Network details

The network G is populated with nodes, the points in the traverse and orientation points, and edges between nodes. The nodes contain point information from the point file and edges are populated with observations from the observation file.

1. Known points: SUR09, SUR12, T117, T13, T40, T444 – y,x coordinates for each
2. Unknowns: Ay, Ax, By, Bx, Cy, Cx, OSUR09, OA, OB, OC, OSUR12
3. Observations:

SOURCE	**TARGET**	**DISTANCE**	**DEGREES**	**MINUTES**	**SECONDS**
SUR09	A	229.598	14	25	28
SUR09	T117	0	88	53	18
SUR09	T13	0	111	2	5
SUR09	T444	0	127	57	40
A	SUR09	0	194	26	16
A	T40	0	316	31	21
A	B	159.918	7	14	50
A	T13	0	121	23	48
B	A	0	187	14	58
B	T40	0	310	20	8
B	C	180.52	14	38	56
B	T117	0	99	33	15
C	B	0	194	38	53
C	T40	0	301	40	59
C	SUR12	247.223	51	33	15
SUR12	C	0	231	33	22
SUR12	T40	0	290	24	23
SUR12	T115	0	47	44	48

4. Provisional Coordinates:
   Provisional coordinates are calculated in InitialCoords.py
   I first created a list that contained the points for the traverse route. To create this route I excluded orientation rays by searching for points that had an out_degree > 0, then from the remaining points created the start of the traverse by checking if the remaining points were fixed or not.

3.3 Functional model

In parametric least squares adjustment there needs to be an equation for every observation. Since we are observing distances and directions, the following are the equations for the observations that make up the functional model of the adjustment:

Angles:
$$\arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) = \theta_{12}$$

Distances:
$$\left[(y_2 - y_1)^2 + (x_2 - x_1)^2\right]^{1/2} = d_{12}$$

These equations are non-linear and are linearized as follows, using Taylor series expansion:
$$F(Y) = F(Y^o) + \frac{\partial F(Y^o)}{\partial Y} X + \ldots$$

Differentiating with respect to y2 as the unknown in each we get:

Angle:
$$\frac{\partial F}{\partial y_2} = \frac{1}{1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2} \cdot \frac{1}{x_2 - x_1} \quad \ldots$$

$$\frac{\partial F}{\partial y_2} = \frac{x_2 - x_1}{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \frac{x_2 - x_1}{d_{12}^2}$$

Distance:
$$\frac{\partial F}{\partial y_2} = \frac{1}{2} \frac{1}{\left[(y_2 - y_1)^2 + (x_2 - x_1)^2\right]^{1/2}} 2(y_2 - y_1) = \frac{y_2 - y_1}{d_{12}}$$

3.4 Least squares criteria

$$V^T PV = min$$

3.5 Observation equations

$$V = Ax - l$$

Where:

- V is the vector of residuals
- A is the design matrix
- x is the vector of unknowns
- l is the vector of observations
- P is the weight matrix

Angles:

$$\frac{x_2^o - x_1^o}{d_{12}^2}.(\delta y_2 - \delta y_1) - \frac{y_2^o - y_1^o}{d_{12}^2}.(\delta x_2 - \delta x_1) - z_1 + \arctan\left(\frac{y_2^o - y_1^o}{x_2^o - x_1^o}\right) - \theta_{12} = v_{12}$$

Distances:

$$\frac{y_2^o - y_1^o}{d_{12}^o}(\delta y_2 - \delta y_1) + \frac{x_2^o - x_1^o}{d_{12}^o}(\delta x_2 - \delta x_1) + d_{12}^o - d_{12} = v_{d_{12}}$$

3.6 Matrices

To generate the matrices below I used the least squares adjustment code provided by Dr. Sithole during the practical session and input my data into the relevant class. My data is sorted using dataCheck.py. The weight matrix was created separately in weight.py and was also imported in the sitholeLeastSquares.py program where the A, A^T, P, X, V and l matrices are all calculated.

The dimensions of the matrices are:

- A : 22 x 11
- X : 11 x 1
- l : 22 x 1
- V : 22 x 1

The A-matrix(design matrix) for all observations is given below:

$$
\left[\begin{array}{lllllllllll}
-7.72E-13 & 1.44E-13 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
-7.72E-13 & 1.44E-13 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
-3.27E-13 & 3.52E-14 & 3.27E-13 & -3.52E-14 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
-7.72E-13 & 1.44E-13 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
-7.72E-13 & 1.44E-13 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
-3.27E-13 & 3.52E-14 & 3.27E-13 & -3.52E-14 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 1.21E-12 & 6.10E-13 & -1.21E-12 & -6.10E-13 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 5.63E-13 & -2.91E-14 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 5.63E-13 & -2.91E-14 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 7.85E-13 & -2.97E-13 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 1.21E-12 & 6.10E-13 & -1.21E-12 & -6.10E-13 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 7.85E-13 & -2.97E-13 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 7.85E-13 & -2.97E-13 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
-9.83E-01 & 1.84E-01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-9.94E-01 & 1.07E-01 & 9.94E-01 & -1.07E-01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 8.94E-01 & 4.49E-01 & -8.94E-01 & -4.49E-01 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 9.35E-01 & -3.54E-01 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
$$

$$
X = \begin{bmatrix} Ax \\ Ay \\ Bx \\ By \\ Cx \\ Cy \\ OSUR09 \\ OA \\ OB \\ OC \\ OSUR12 \end{bmatrix}
$$

$$
l = \begin{bmatrix} \alpha1-\acute{\alpha}1 \\ \alpha2-\acute{\alpha}2 \\ \alpha3-\acute{\alpha}3 \\ \alpha4-\acute{\alpha}4 \\ \alpha5-\acute{\alpha}5 \\ \alpha6-\acute{\alpha}6 \\ \alpha7-\acute{\alpha}7 \\ \alpha8-\acute{\alpha}8 \\ \alpha9-\acute{\alpha}9 \\ \alpha10-\acute{\alpha}10 \\ \alpha11-\acute{\alpha}11 \\ \alpha12-\acute{\alpha}12 \\ \alpha13-\acute{\alpha}13 \\ \alpha14-\acute{\alpha}14 \\ \alpha15-\acute{\alpha}15 \\ \alpha16-\acute{\alpha}16 \\ \alpha17-\acute{\alpha}17 \\ \alpha18-\acute{\alpha}18 \\ s1-\acute{s}1 \\ s2-\acute{s}2 \\ s3-\acute{s}3 \\ s4-\acute{s}4 \end{bmatrix}
$$

The solutions to the unknowns are found using the following equation:

$$x = (A^T PA)^{-1} A^T Pl$$

Chapter 4

Conclusion

4.1 Overview

By inspecting my results after iterating 10 times I found them to have very large residuals. Despite applying corrections to my code and restructuring my inputs numerous times over, I still was unable to find the error in the code. This error is also evident in the A matrix displayed above.

With values as incorrect as mine are, it is impossible to plot error ellipses or test the adjustment since it is obviously wrong. My final adjusted coordinates are extremely incorrect due to this.

Final adjusted coordinates

	y	x
A	1.66E+99	3.67E+99
B	2.36E+99	-5.19E+99
C	1.68E+99	3.66E+99

The gross error is also obvious from the value of the posteriori variance factor being so large where:

$$\hat{\sigma}_0^2 = \frac{V^T P V}{n - u} = 8.05962673e+203$$

The precision of the points are calculated using this value:

$$\sum{}_x = \hat{\sigma}_0^2 \left(A^T P A\right)^{-1}$$

4.2 Evaluation

The erroneous can definitely be attributed to my own logical error and lack of knowledge and skill in coding for such a problem, but in my own opinion I felt like I had a fairly firm understanding of the problem and how to undertake it using a logical approach as taught in lectures.