Survey network adjustments

by the L_1 method

Bruce R. Harvey

School of Surveying and Spatial Information Systems, University of NSW

Abstract

The application of the L_1 method to adjust survey networks of directions and distances while minimising the sum of the absolute values of the weighted residuals, $\sum \left| \dfrac{v}{s} \right|$, is investigated in this paper. The L_1 norm algorithms are robust in the presence of gross observation errors but are different from iteratively reweighted least squares. Comparisons are made between the results of the L_1 method and the results of conventional least squares (LS) adjustment. The L_1 method can easily be programmed to read the same input file as a conventional least squares adjustment program. It can be used to generate approximate coordinates, locate gross error observations, and adjust the network despite rank defects in the network. It is particularly useful for analysing erroneous input data files which cause conventional least squares programs to crash or not converge. It is suggested that the L_1 method be considered as an additional tool that is sometimes useful to use as well as, but not as a replacement for, least squares.

1. Introduction and Historical Development

According to Dodge (1987) a form of the L_1 method was used by Galilei in 1632 to do astronomical positioning. Least squares has been well used since the time of Gauss and Legendre (late 1790s), but gross errors or outliers can have an unfairly large influence on least squares results. Robust methods have been created by modifying least squares to reduce the effect of outliers. The L_1 method (also called the least absolute values method or L_1 norm method) is a robust alternative to least squares (eg Huber (1964), Fuchs (1981), Hahn & Bill (1984), Kampmann (1989)).

Rousseeuw (in Dodge, 1987) presents some examples of people's reactions to the promotion of the L_1 method. The L_1 method has not been used as much as least squares because of its computational difficulties (there is no direct formulae like the normal equations of least squares, just a 'searching algorithm') and because it does not use much of the data, redundant data is virtually ignored. Also, statistical analysis of the results of an L_1 adjustment was not available in the early years. Rousseeuw and Leroy (1987) concentrate on the use of robust techniques such as L_1 for outlier detection especially applied to regression problems and gives modified methods that aim to combine the best features of L_1 and least squares.

Huber (in Dodge, 1987) refers to the L_1 method as "<u>the</u> starting point for estimates that have to be calculated by iterative procedures." This paper advocates the use of the L_1 method, in some circumstances, as an addition to, but not a replacement for, a conventional least squares adjustment (eg Harvey, 1991). The L_1 method could be used to generate approximate values of parameters (coordinates), and help solve or clarify difficulties that a least squares solution may have with rank defects and data outliers. Any problems could be rectified and then least squares and associated statistical analysis could be used to obtain the best results.

This paper will cover how to obtain a minimum $\sum \left| \dfrac{v}{s} \right|$ solution, its features, whether it can be used to generate approximate coordinates, how it treats gross observation errors, and whether it can help solve problems in input files when a least squares solution does not converge.

Much has been written about the analysis of survey data containing outliers, particularly detection techniques, robust estimation and iteratively reweighted least squares (eg Baarda, 1968, Pope, 1976, Kubik et al, 1985, Jorgensen et al, 1985, and Caspary, 1987). **Perhaps more emphasis needs to be placed on the avoidance or prevention of such errors rather than their cure or detection.** Moreover, apart from their initial attempts with new equipment, surveyors and survey students rarely make large errors in observations, especially when using modern electronic instruments with data recording and electronic transfer to computer files.

However there are other sources of errors in input files for least squares adjustments regardless of the software package used. For example, the approximate (starting) coordinates of points may be in incorrect relative positions, or errors in the format or layout of the data, or incorrect point numbers or set/arc numbers for observations. If a least squares solution of the data does not converge beginners can experience considerable difficulty.

2. The L$_1$ Method

The L$_1$ method minimises the sum of the absolute values of the residuals (v). It is a robust method that can cope with errors in input files. It is not the same as iteratively reweighted least squares.

The procedure is to set up equations (one per observation), as in parametric least squares, of the form Ax = b. Where A is the matrix of partial derivatives $\partial F/\partial x$, x is a vector of parameters, and b is a vector representing the difference between observations and values calculated from estimates of the parameters. If there are n observations and u unknown parameters then solve n simultaneous equations (Ax = b) directly instead of solving u normal equations (eg A^{T}Ax = A^{T}b) as in least squares. [Least squares problems can also be solved without forming normal equations see, for example, Lawson and Hanson (1974).]

The first step in an L$_1$ solution is to decide which equations to use. One way is to try every possible combination of equations, solve for x with each (uxu) combination, and select the one with the smallest $\sum|v|$. This simple approach is very inefficient. A better method is to use the algorithm of Barrodale and Roberts (1974) based on Linear Programming concepts. Branham (1990) gives a modified version of the Fortran source code. Other algorithms have also been developed (see Branham (1990), Dodge (1987)).

One of the theorems of linear programming is that given a region of possible solutions the best solution is either at a corner or on an edge (boundary line) of the region. In the case of a corner there is one single best solution. In the edge case any point on the line including the corners at the end of the line are equally suitable solutions, thus there are multiple solutions that all give the same $\sum|v|$. The Barrodale and Roberts algorithm (see Branham (1990) for details) selects one combination of u equations (a 'corner') as a basis then using a process similar to the pivot row operations of Gaussian elimination replaces one of the basis equations with one of the other equations, solves the equations and sees whether the $\sum|v|$ is smaller or not. The choice of which equations to swap is determined systematically by the algorithm in an efficient manner.

Thus one feature of the L$_1$ method is that u of the n equations will be satisfied exactly giving zero residuals for the corresponding observations (see Branham, 1990 and his references for proof). Many (n-u) of the observation equations are virtually not used in the solution for x. Their only use is in deciding which of the equations are used. This is a concern to many people and is one of the reasons Gauss preferred least squares to L$_1$ (Branham, 1990). This paper suggests the L$_1$ method be used as a tool to find the causes of problems with a data set, then use least squares on the 'cleaned' data.

Given A and b the Barrodale and Roberts algorithm solves for x, even if the A matrix is rank deficient (eg not enough parameters held fixed). It calculates the rank of the matrix and indicates if there is more than one solution with the same $\sum|v|$. There are two types of iteration in an L$_1$ solution of a nonlinear survey adjustment. Firstly, the

common least squares iteration: approximate coordinates, A and b, solution, improved coordinates, new A and b, another solution, etc. Secondly, an iteration within each solution of Ax=b where the algorithm iterates to find which observation equations to use to determine the values of x.

The computation time of the L_1 method is usually larger than the computation time of a least squares solution of the same data set. This is because the L_1 method solves a nxu set of equations and least squares solves uxu with symmetric normal equations (n is usually considerably larger than u). For small data sets the difference is insignificant. However, if the data set is so large that virtual memory has to be used the L_1 method can be very slow. With 'clean' data least squares is therefore preferable. However if there are many problems with the input data that cause least squares not to converge (eg rank deficient, very poor approximate values of the parameters, large gross errors in the data), then the L_1 method may be useful.

Branham (1990) describes how to use the L_1 method to determine the quality of the parameters. However this author prefers not to do those calculations and prefers to use least squares to determine standard deviations (etc) of the parameters.

3. Modification of the L_1 Method

To account for the fact that survey networks usually have observations with different units (eg directions and distances) and observations of varying qualities the following simple modification of the L_1 method is suggested. Huber (in Dodge, 1987) discusses some aspects of other choices of weights in the L_1 method. The L_1 method solves:

$$Ax = b \qquad \text{so that } \sum|v| \text{ is a minimum.}$$

The modified method solves:

$$P^{1/2} Ax = P^{1/2} b \qquad \text{so that } \sum\left|\frac{v}{s}\right| \text{ is a minimum,}$$

$$\text{where} \quad P^{1/2} = \begin{pmatrix} \frac{1}{s_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{s_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{s_n} \end{pmatrix},$$

s_i is the standard deviation of the ith observation and the observations are assumed uncorrelated.

This is easily achieved by dividing each row in the A matrix (one row per observation) and each term in the b vector, by the standard deviation of the

corresponding observation. Then use the standard L_1 algorithm. For comparison purposes, least squares solves: $(A^TPA)x = A^TPb$, and the modified L_1 method solves: $P^{1/2} Ax = P^{1/2} b$

4. Other considerations

A conventional least squares program to adjust a survey network could be changed to use an L_1 algorithm instead of solving normal equations, or the L_1 algorithm could be added as an option. However the A matrix and b vector need to be formed and stored rather than the common practice of adding each observation's contribution to the normal equations directly. Some care also needs to be taken with fixed parameters when forming the A matrix.

Since the L_1 method calculates coordinates from non redundant subsets of the data it would appear to be a possible method for calculating approximate coordinates for use in a least squares adjustment. The examples which follow will investigate this. Some alternative methods of finding approximate coordinates for least squares can be time consuming and error prone. If coordinates are calculated by software that uses the observations and known coordinates then gross errors may yield subsets of the points with grossly incorrect coordinates.

5. Mean and median example

The following hypothetical example is used to highlight the differences between a simple L_1 adjustment, the median, and a least squares adjustment, the mean, in the presence of an artificially introduced gross error. The remaining examples in this paper are all real data sets.

Suppose 5 measurements of a distance are: 96, 98, 100, 102, 140. The last observation should have been 104 not 140. Table 1 shows the results of L_1 and least squares solutions. The gross error in the last observation has less effect on the median than on the mean. Least squares has spread the effect into all residuals L_1 has not.

```
        Table 1.  Comparison of L₁ and LS with artificial data

              median (L₁) = 100              mean (LS)   = 107.2
measurement      v      |v|      v²           v      |v|        v²
    96           4       4      16          11.2    11.2      125.44
    98           2       2       4           9.2     9.2       84.64
   100           0       0       0           7.2     7.2       51.84
   102          -2       2       4           5.2     5.2       27.04
   140         -40      40    1600         -32.8    32.8     1075.84
                Σ       48    1624                  65.6     1364.8
```


6. Distance resection example

The example data in figure 1 was used to determine the E and N coordinates of a point P by distances observed to four control marks A, B, C, and D. A, B, C, and D have known fixed coordinates, which are considered error-free. The distances are have equal precision and are uncorrelated. Since all observations are in the same units and equal precision, this example will minimise $\Sigma|v|$ rather than $\Sigma|v/s|$.
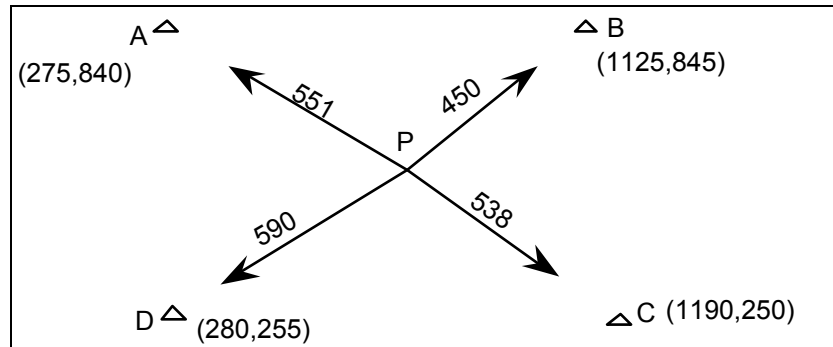

Figure 1. Distance resection data, coordinates are in the order (E,N).

A least squares solution gave: P = (764.77, 582.70) and residuals, v = (2.2, -4.4, 1.9, -4.9)^T and $\Sigma|v|$ = 13.4.

Several 'intersection by distances' solutions were calculated by choosing pairs of lines and then calculating P from two points. The distances from P to the other two points were calculated and thus residuals. This was repeated for all possible combinations (six for this data) of pairs of lines.   The solution with the smallest $\Sigma|v|$ used the lines from B and C with P = (763.15, 577.48) and $\Sigma|v|$ = 12.4.

The L_1 method yielded a single solution with a rank of 2, which equals the number of parameters and therefore no rank defect. The coordinates of P, after iteration, were (763.15, 577.48). The residuals, v = (3.3, 0,0,-9.1)^T and $\Sigma|v|$ = 12.4.

Both L_1 and least squares solutions give the largest residual to the distance to D, but it was most pronounced in the L_1 solution. The coordinates differ by 5.5 which is not large compared to the least squares solution error ellipse.   As expected, the least squares solution had the smaller Σv^2 and the L_1 solution had the smaller $\Sigma|v|$.

A semigraphic solution of this data is shown in figure 2. There is an arc (line when magnified) for each distance. The least squares solution is near the centre of the error figure formed by the 4 arcs. The L_1 solution is at one of the corners of the error figure, ie the intersection of 2 arcs. So it uses only two distances, but the algorithm has to decide which of the possible corners gives the smallest $\Sigma|v|$.
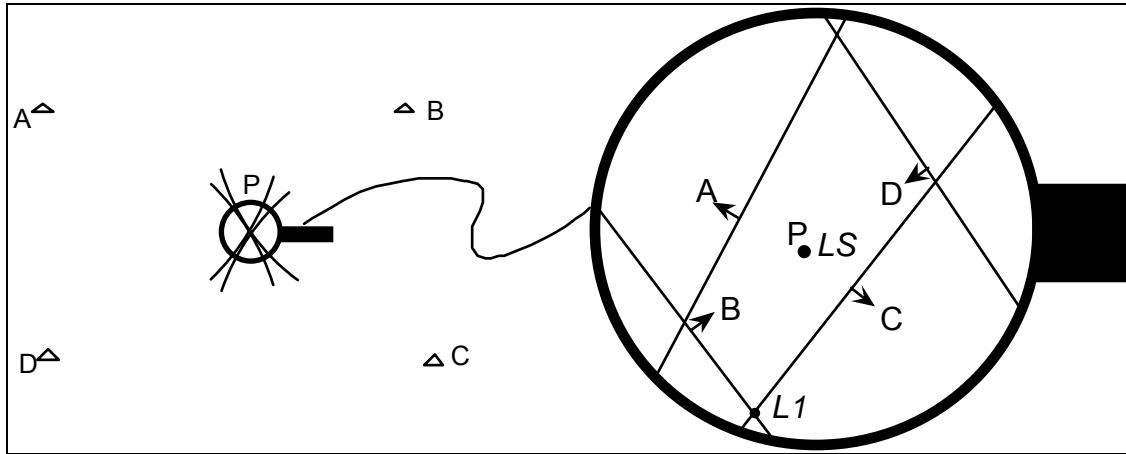
Figure 2. Semigraphic LS and L_1 solution of resection by distances.

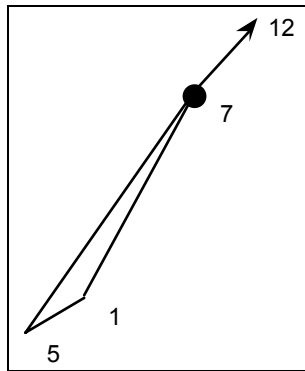## 7. Simple triangle traverse example

Figure 3. Simple triangle traverse

Figure 3 shows a plan of a simple traverse where points 7 and 12 have known coordinates held fixed. The measurements and the residuals from two solutions are shown in table 2.

Both L_1 and least squares required 3 iterations for convergence. The L_1 method gave a non unique optimal solution with matrix rank 7 and 7 parameters. The adjusted coordinates from the two methods were compared. The differences are of the order 2-3mm, the least squares error ellipses are about 1 cm.

```
        Table 2. Results of triangle traverse adjustments.

  Obs Type   From  To   Measurement      s       v (LS)      v (L₁)
  DIRECTION    1    5     40 47 30       1"      -1.2"       -0.9"
  DIRECTION    1    7    187 43 19       1"       1.2"        0.0"
  DIRECTION    5    7    179 59 52       1"      -2.8"        0.0"
  DIRECTION    5    1    204 57 35       1"       2.8"        6.8"
  DIRECTION    7   12    139 17 49       1"       0.0"        0.0"
  DIRECTION    7    1    308 28 22       1"       0.5"        0.0"
  DIRECTION    7    5    316 34 43       1"      -0.5"       -0.7"
   HOR DIST    1    5       552.968      1mm     -0.48mm      0.0mm
   HOR DIST    1    7      1655.179      1mm     -0.72mm      0.0mm
   HOR DIST    7    5      2139.950      1mm      0.68mm      0.0mm


Final L₁ coordinates:
   Point       E m            N m
     1      9279.747       5154.883
     5      8794.460       4889.798
     7     10064.072       6612.433     Fixed
    12     22099.220      21416.713     Fixed
```

Another L₁ solution was calculated but nothing was held fixed. There were thus 11 parameters and 10 observations. The L₁ solution noted the rank of the matrix was 7 and there were 11 parameters. It yielded a non unique optimal solution after 3 iterations and did not require any special techniques to overcome the rank deficiency.   In least squares programs special steps are needed to overcome such a rank deficiency.

The final coordinates and residuals from the L₁ solution with no points held fixed are given in Table 3. Note that the coordinates are not the same as when 7 and 12 were held fixed, but they are accurate enough to use as starting points in a least squares analysis. However the residuals are very similar (but different) to those for the L₁ solution with points 7 and 12 fixed.

```
        Table 3.  Results from an L₁ solution with no points held fixed.

     OBS    FROM  TO         v       Point       E m            N m
     DIR     1    5        0.0"        1      9279.285       5154.002
     DIR     1    7        0.9"        5      8794.000       4888.912
     DIR     5    7        0.0"        7     10063.596       6611.559
     DIR     5    1        6.8"       12     22099.220      21416.713
     DIR     7   12        0.0"
     DIR     7    1        0.0"
     DIR     7    5       -0.7"
     DIS     1    5        0.0mm
     DIS     1    7        0.0mm
     DIS     7    5        0.0mm
```

Several other related solutions were calculated. Solutions using zero as approximate values for coordinates for points 1 and 5, but with points 7 and 12 held fixed, were calculated. Both least squares and L_1 methods converged to correct answers at a similar rate. In another solution a 2m radiation from point 1 was added. The approximate coordinates for this point were placed in the reverse bearing. Again both least squares and L_1 methods converged to correct answers at a similar rate.

Another solution used included a direction but left out the distance for a radiation from point 1. Least squares programs usually 'crash' unless special precautions are taken. The L_1 method did not 'crash'. It calculated the rank of the matrix to be one less than the number of parameters (thus indicating a rank defect of one). The residuals of all observations were the same as for an L_1 solution that included the distance observation, and the coordinates of all points (except the radiated point) were correct. So in this case the L_1 method is far more robust and stable in the presence of rank defects (and therefore more 'crash' proof).

8. Traverse example with gross error.

Figure 4 shows a comparison of L_1, least squares and a Bowditch adjustment for a traverse. The position of L_1 (r) and Bowditch (b) coordinates are shown with respect to the least squares solution at the same scale as the least squares standard error ellipse. This particular data gave a variance factor of about 5, the error ellipses shown here are not scaled by this factor. The difference vectors show that L_1 coordinates are not significantly worse than Bowditch coordinates.

In the least squares adjustment the largest scaled residual |v/s| was the distance from 2 to 3, it had a v of +14mm (about 2 times standard deviation,s). In the L_1 solution the same observation also had the largest |v/s|, here v was +29mm (about 5s), and was thus more prominent. A check of field notes and reductions subsequently showed a 12mm error in this distance. Note that the residual from the L_1 solution includes this gross error and absorbs normal measurement errors from some other observations (they were given v = 0).
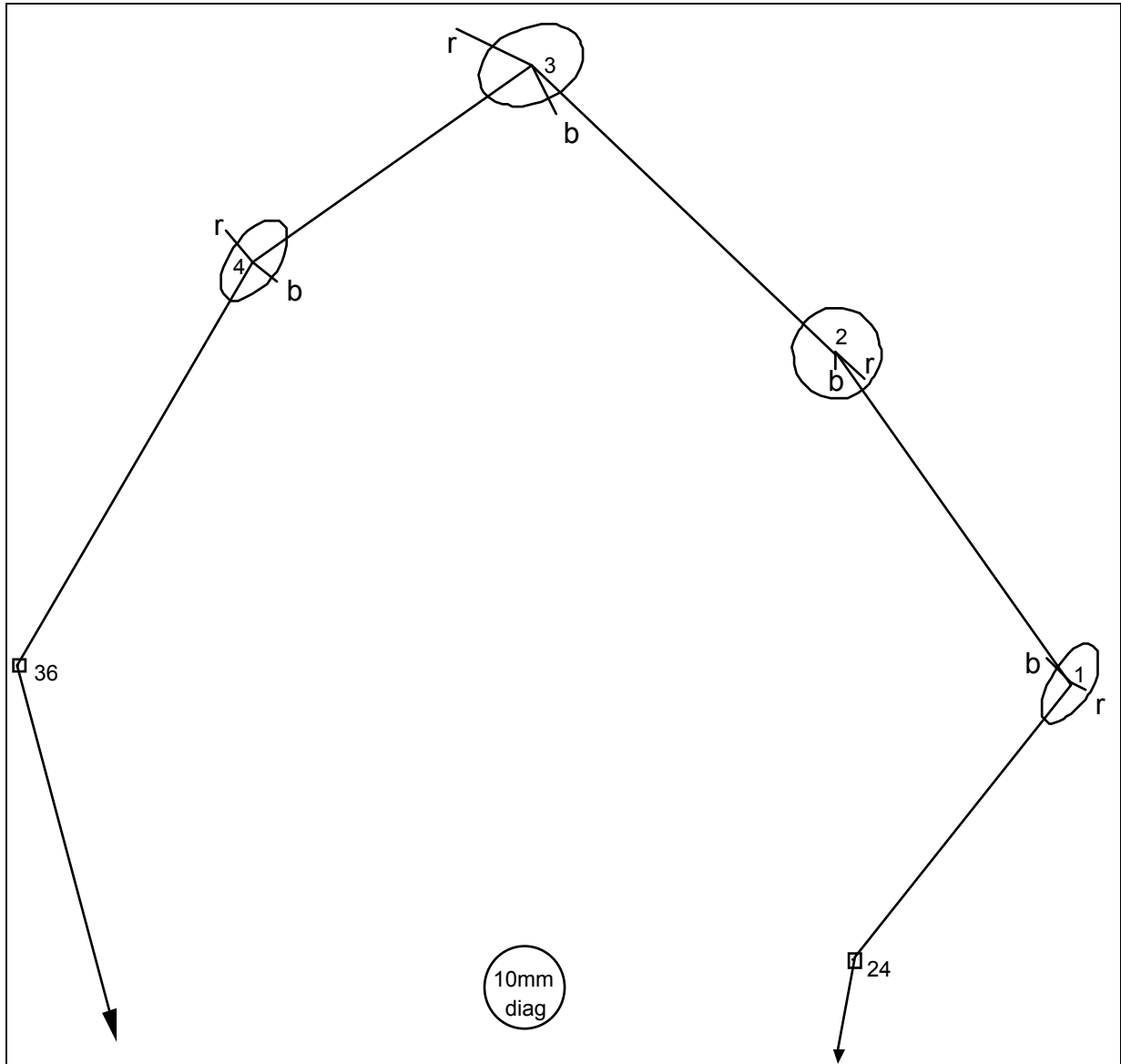
Figure 4. Comparison of coordinates obtained by $L_1(r)$, LS and Bowditch (b) adjustment for a traverse.

Despite this example, in general least squares tends to hide gross errors by spreading error to the residuals of other observations, L_1 does the opposite. L_1 magnifies gross errors by absorbing errors from other observations. Neither case, in general, gives residuals that indicate true errors.

When no approximate coordinates were entered for the unknown points in this traverse both L_1 and least squares converged to correct solutions after about the same number of iterations. So for this data set there is no advantage in using the L_1 method to generate approximate coordinates for a least squares solution. In this traverse it would be more efficient to use the mean of the fixed points' coordinates as the approximate coordinates for all other points rather than zero coordinates and L_1 calculations. An alternative method described by Gruendig (1985) is far more efficient.

9. Network example

The network in figure 5 was designed, observed, preprocessed and computed by students at a survey camp. It contains about 1500 observations and 1000 parameters. Students had a lot of difficulty processing this data - they couldn't get a converging least squares solution or find the problems. There were enormous corrections to the approximate coordinates. This data set was too large to run on a portable computer with the author's L_1 program.
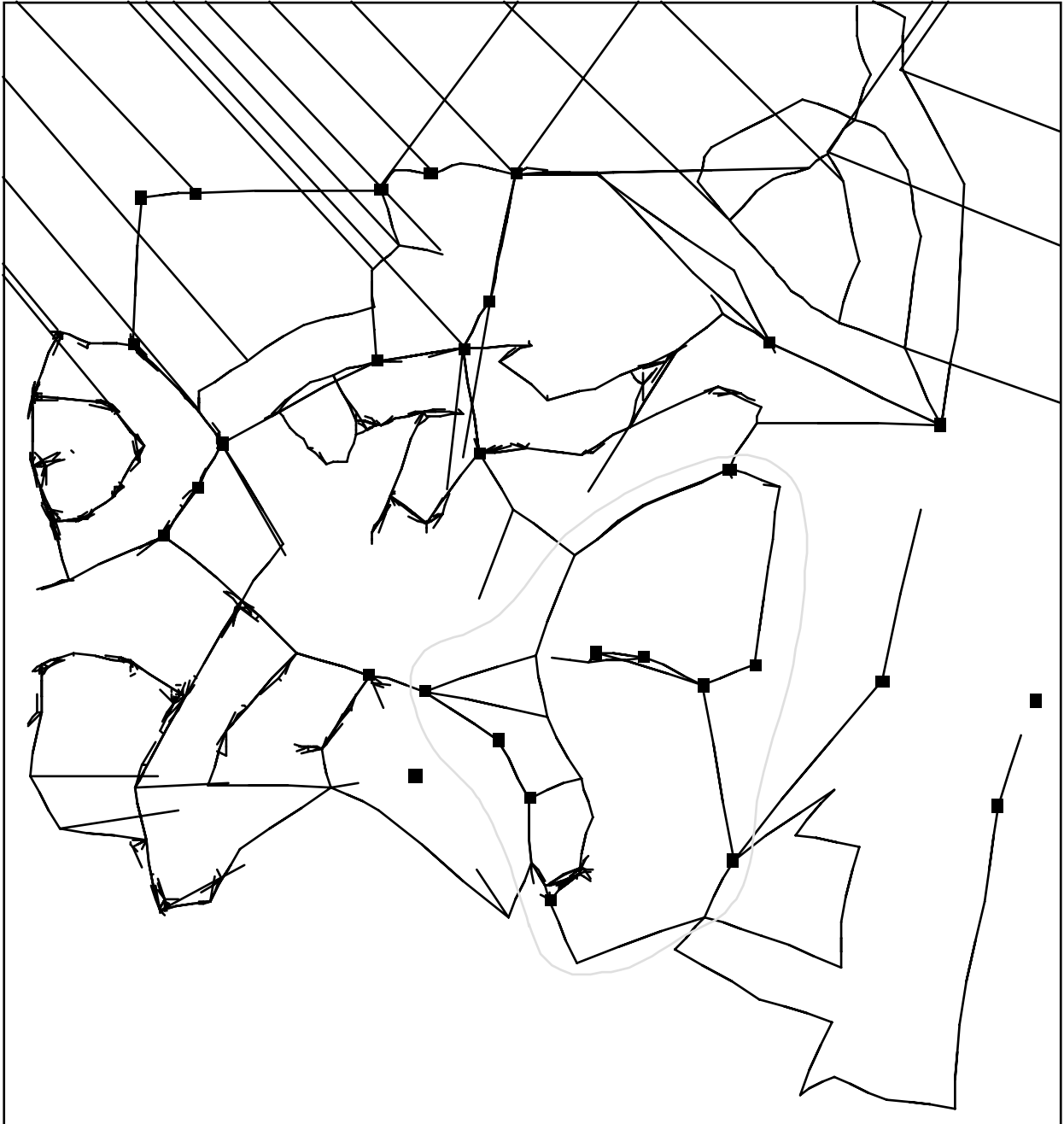

Figure 5. Survey camp network plan.

One strategy was to carefully check the input file for transcription, format and other errors. Unfortunately the students involved were not able to detect any errors, probably because checking over a thousand lines of input is a very demanding task. Other strategies were then used to locate the problems, including investigating 'prefit' residuals (O-C terms) ie before any adjustment, plotting the network (by reading the same input file into a network drawing program), and dividing the total area into sections. Using this last strategy, all sections except one were then able to be 'cleaned'. The remaining section is enclosed by a dotted curved line in figure 5.

A least squares adjustment of the troublesome section did not converge. There were corrections to coordinates of the order 40 m at every iteration. A solution by the weighted L_1 method for this data converged quickly. There were 86 parameters and 162 observations. Before any adjustment the residuals (O-C terms) for most observations including all distances were acceptable. However the residuals for 9 directions from one point were enormous (about 130°). The L_1 solution calculated the rank of the A matrix to be 85 but there were 86 parameters! The correction to approximate coordinates was less than 0.2m for all points. The L_1 solution gave only one large residual and together with the rank defect information led to the discovery of a direction input with an incorrect arc/set number.

Certainly, more experienced users or more sophisticated least squares software may have detected and rectified this error, but the L_1 solution did it without the expertise. The errors were corrected and a least squares solution was then performed.

10. Conclusions

This paper has shown how to calculate an L_1 solution for a survey network. Its features have been described and compared with conventional least squares, and examples given. L_1 uses a minimum subset of the data, so many observations have zero corrections and the adjusted coordinates are determined directly from non-redundant data. It has been shown that it can do adjustments of data without approximate coordinates but does not do this any better than least squares. Perhaps a combination of Gruendig's method (Gruendig, 1985) and the L_1 method could be considered for approximate coordinate generation for survey networks. The L_1 method can highlight gross observation errors, but the residuals are usually larger than the true errors (compare this with least squares where residuals of gross error observations are usually smaller than the true errors). However it is also robust against rank deficient data sets and thus perhaps more likely to be 'crash proof'. L_1 algorithms require considerably more computer time and space than least squares methods for networks so large that they need virtual computer memory.

This paper has not suggested L_1 be used instead of least squares, but it may be a useful program to use as well as a least squares program or as an option in a least squares program. It may help to resolve difficulties with some troublesome data sets.

11. Acknowledgment

J.M. Rüeger kindly commented on a draft on this paper.

12. References

Baarda, W.   (1968) A testing procedure for use in Geodetic Networks, Net. Geod. Comm. Vol 2 No 5.

Barrodale, I. and F.D.K. Roberts (1974)   Algorithm 478: Solution of an Overdetermined System of Equations in the L1 norm, Comm. ACM, 17, p.319.

Branham, R.L. Jr. (1990) Scientific Data Analysis: An introduction to overdetermined systems, Springer-Verlag New York Inc., 237p.

Caspary, W.F. (1987) Concepts of Network and Deformation Analysis, Monograph 11, School of Surveying, University of New South Wales, Australia.

Dodge, Y. Editor (1987) Statistical Data Analysis Based on the L_1-norm and related methods, North-Holland.

Fuchs, H. (1981) Adjustment by minimizing the sum of absolute residuals, International Symposium on Geodetic Networks and Computations, Munich.

Gruendig, L. (1985) Special Adjustment Tools for Surveying Networks, p. 19-33, in Papers for the Precise Engineering and Deformation Surveys Workshop, Ed. by W.F. Teskey and L. Gruendig, University of Calgary, Publ.60004.

Hahn, M. and R. Bill (1984) Ein Vergleich der L1- und L2-Norm am Beispiel Helmerttransformation, AVN 11-12, p. 440+.

Harvey, B.R. (1991) Practical Least Squares and Statistics for Surveyors, Monograph 13, School of Surveying, University of NSW. 229 pp.

Huber, P.J. (1964) Robust estimation of a location parameter, Annalen der Mathematishen Statistik, Vol. 35, pp. 73-101.

Jorgensen, P.C., K. Kubik, P. Frederiksen, and W. Weng (1985) Ah, Robust Estimation, AJGPS, No. 42, pp. 19-32.

Kampmann, G. (1989) Zur Ausgleichung freier Netze mit der L1-Norm-Methode, AVN 3, p110+.

Kubik, K., W. Weng, and P. Frederiksen (1985) Oh, Gross Erorors!, AJGPS, No. 42, pp. 1-18.

Lawson, G.L. and R.J. Hanson (1974) Solving Least Squares Problems, Prentice-

Hall, 340p.

Pope, A.J. (1976) The Statistics of Residuals and the Detection of Outliers, NOAA
Tech. Rpt. NOS65NGS1.

Rousseeuw P.J. and A.M. Leroy (1987) Robust regression and Outlier Detection,
John   Wiley & Sons.