Lab 4

ESSE3630

Professor Benjamin Brunson

Submitted by Jared Yen

Date of Submission: 11/03/2021

Contents

Part One: Pre-Adjustment Quality Assurance

Task 1

The unknown parameters are the 3D cartesian coordinates of all points in Figure 1 except control point A.

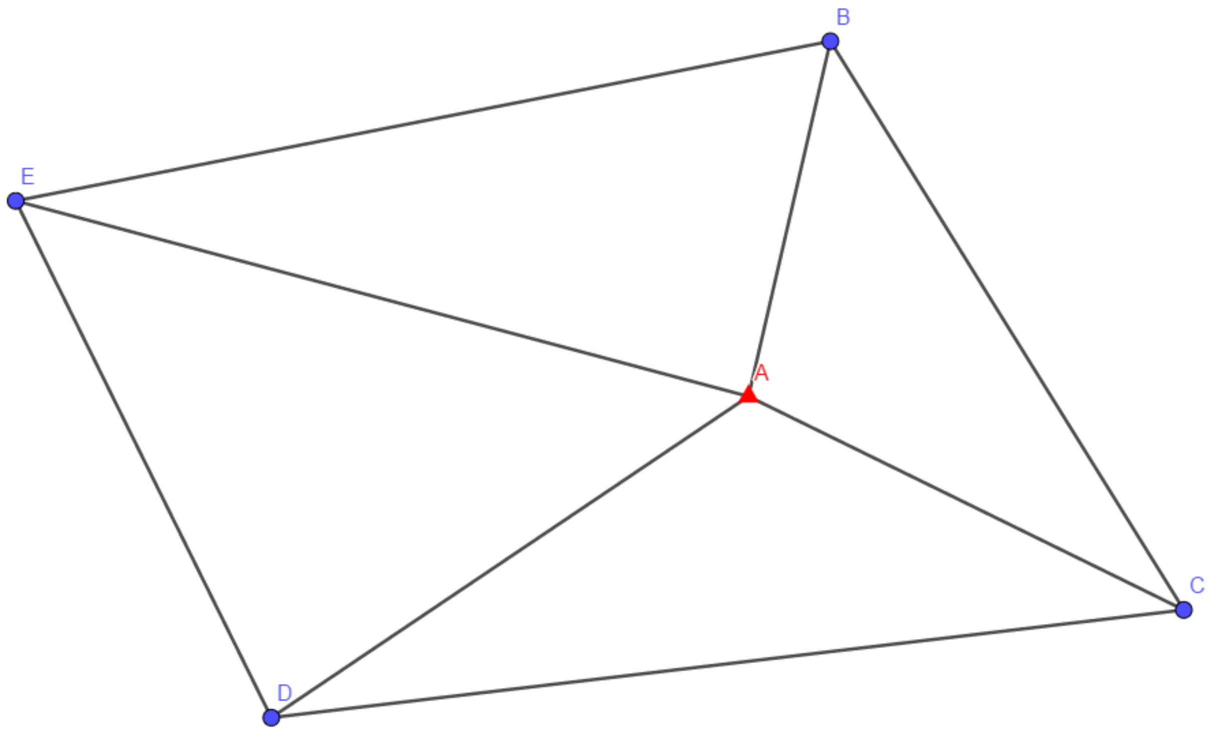


Figure 1 - 3D GNSS Baseline network with control point A and the rest unknown

For unknowns x

$$x = \begin{bmatrix} \vec{X}_B \\ \vec{X}_C \\ \vec{X}_D \\ \vec{X}_E \end{bmatrix}$$

$$\overrightarrow{X_i} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

Task 2

The observation vector l is defined as

$$l = \begin{bmatrix} \overrightarrow{DX}_{AB} \\ \overrightarrow{DX}_{BA} \\ \vdots \\ \overrightarrow{DX}_{BE} \end{bmatrix}$$

Where the GNSS baseline measurement $\overrightarrow{DX}_{ij}$ measures the change in coordinates from point i to j. The measurements appear in the row-order they appear in the input tables in Appendix A.1.

The weight matrix $P = D_{LL}^{-1}$ and the covariance matrix is defined with a diagonal matrix consisting of covariance sub-matrices

$$D_{LL} = \begin{bmatrix} D_{\overrightarrow{DX}_{AB}} & 0 & 0 & 0 \\ 0 & D_{\overrightarrow{DX}_{BA}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & D_{\overrightarrow{DX}_{BE}} \end{bmatrix}$$

Where each zero is a 3x3 zero matrix and $D_{\overrightarrow{DX}_{ij}}$ is the covariance matrix made from the variances and covariances of the change in cartesian coordinates $DX_{ij}, DY_{ij}, DZ_{ij}$ listed in A.1, full code rearranging the input data table into D_{LL} in A.4.

```
P = 48×48
   4.0009e+06        51049       -33182            0            0            0  ...
        51049   6.2526e+06       -64219            0            0            0
       -33182       -64219   2.0417e+06            0            0            0
            0            0            0   4.0009e+06        51049       -33182
            0            0            0        51049   6.2526e+06       -64219
            0            0            0       -33182       -64219   2.0417e+06
            0            0            0            0            0            0
            0            0            0            0            0            0
            0            0            0            0            0            0
            0            0            0            0            0            0
       ⋮
```

The first design matrix A was calculated by the partial derivatives of the observation equation

$$\overrightarrow{DX}_{ij} = \overrightarrow{X_j} - \overrightarrow{X_i}$$

$$\frac{\partial \overrightarrow{DX}_{ij}}{\overrightarrow{X_i}} = -I_3$$

$$\frac{\partial \overrightarrow{DX}_{ij}}{\overrightarrow{X_j}} = I_3$$

$$A = \frac{\partial l}{\partial x} = \begin{bmatrix} \frac{\partial \overrightarrow{DX}_{AB}}{\overrightarrow{X_B}} & 0 & 0 & 0 \\ \frac{\partial \overrightarrow{DX}_{BA}}{\overrightarrow{X_B}} & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ \frac{\partial \overrightarrow{DX}_{BE}}{\overrightarrow{X_B}} & 0 & 0 & \frac{\partial \overrightarrow{DX}_{BE}}{\overrightarrow{X_E}} \end{bmatrix} = \begin{bmatrix} I_3 & 0 & 0 & 0 \\ -I_3 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ -I_3 & 0 & 0 & I_3 \end{bmatrix}$$

The rest of the partial derivatives are evaluated to zero when the unknown is not in the observation equation.

The A matrix was computed in MATLAB by first importing the table from A.1, converting the letters to numbers for indexing purposes

```
% converts letters in From and To columns to numbers A=1,B=2...
letter2number = @(c) 1+lower(c)-'a';
T{:,[1 2]}
```

```
ans = 16×2 cell

'A'            'B'
'B'            'A'
'A'            'C'
'C'            'A'
       ⋮
```

```
FromTo=cellfun(letter2number,T{:,[1 2]})
```

```
FromTo = 16×2

     1     2
     2     1
     1     3
     3     1
      ⋮
```

Since point A is a control point, it is treated as a constant and evaluated to zero in the A matrix. Every 3 lines of the A matrix representing a single 3D GNSS baseline $\overrightarrow{DX}_{ij}$ measurement was calculated using a helper function `calcA`. As point A = 1, nothing was done to the initialized A zero matrix. The indexing of the identity matrices I_3 from the definition of A into the correct 3x3 block into the 3x12 block representation of $\overrightarrow{DX}_{ij}$ was done by using the numerical representation of the points B through E inputted as `from` $= i$ and `to` $= j$.

```
function A = calcA(from,to)
% assuming coordinate 1 is the control point
u=4;
A=zeros(3,3*u);
if from~=1
    A(:,((from-1)*3-2):(from-1)*3)=-eye(3);
end
if to~=1
    A(:,((to-1)*3-2):(to-1)*3)=eye(3);
end
```

Task 3

To see if the accuracy of the unknown points are at least at an accuracy of 1cm, the covariance matrix of the adjusted unknown points $D_{\hat{x}\hat{x}} = (A^T PA)^{-1}$ then the square root of the diagonals were calculated to receive the standard deviations or accuracies for each observation in the order they appear in the definition for l.

```
Dxhat=inv(A'*P*A)
sigmaxhat=sqrt(diag(Dxhat))
```

```
sigmaxhat = 12×1
0.000241522945769824

0.000193218356615859
```

```
0.000338132124077754

0.000241522945769824

0.000193218356615859

0.000338132124077754

0.000241522945769824

0.000193218356615859

0.000338132124077754

0.000241522945769824

0.000193218356615859

0.000338132124077754
```

These error units are meters and none of the adjusted unknown errors exceed 1cm or 0.01m therefore the observations in the network will achieve the accuracy of 1cm.

Task 4

The geometric conditions I chose were the double-run measurements so 8 for the 16 total measurements and 1 loop condition of loop ABC where the condition was $\overrightarrow{DX}_{AB} + \overrightarrow{DX}_{BC} + \overrightarrow{DX}_{CA} = 0$. The double run measurement conditions were the first 8 conditions defined as

$$w_i = \overrightarrow{DX}_{ij} + \overrightarrow{DX}_{ji} = 0$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_8 \\ w_9 \end{bmatrix} = \begin{bmatrix} \overrightarrow{DX}_{AB} + \overrightarrow{DX}_{BA} \\ \overrightarrow{DX}_{AC} + \overrightarrow{DX}_{CA} \\ \vdots \\ \overrightarrow{DX}_{EB} + \overrightarrow{DX}_{BE} \\ \overrightarrow{DX}_{AB} + \overrightarrow{DX}_{BC} + \overrightarrow{DX}_{CA} \end{bmatrix}$$

To evaluate the conditions and check for outliers, the second design matrix B is needed to calculate the misclosure variance $D_{ww} = BD_{ww}B^T$

$$B = \frac{\partial l}{\partial x} = \begin{bmatrix} \frac{\partial w_1}{\overrightarrow{DX}_{AB}} & \frac{\partial w_1}{\overrightarrow{DX}_{BA}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial w_2}{\overrightarrow{DX}_{AC}} & \frac{\partial w_2}{\overrightarrow{DX}_{CA}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial w_8}{\overrightarrow{DX}_{EB}} & \frac{\partial w_8}{\overrightarrow{DX}_{BE}} \\ \frac{\partial w_9}{\overrightarrow{DX}_{AB}} & 0 & 0 & \frac{\partial w_9}{\overrightarrow{DX}_{CA}} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} I_3 & I_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_3 & I_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_3 & I_3 \\ I_3 & 0 & 0 & I_3 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

In MATLAB, I calculated the misclosures as per the following.

```
w=zeros(9,3);
% since double run measurements are stacked in adjacent row pairs,
for i=1:size(data,1)/2
    w(i,:)=data(2*i-1,3:5)+data(2*i,3:5);
end


% included loop ABC in cond.s w9=AB+BC+CA
ftLoop= [1 2; 2 3; 3 1]; % ft=[from to;...]
for i=1:size(ftLoop,1)
    % compares from and to columns in input data and if both equal the from and
    % to in ftLoop(i,:) then return ind
    ind=find(sum(data(:,1:2)==ftLoop(i,:),2)==2);
    w(9,:)=w(9,:)+data(ind,3:5);
end
w
```

```
w = 9×3

      -0.016         0.012         0.004
      -0.002         0.005        -0.048
      -0.018        -0.001         0.038
      -0.029        -0.035        -0.078
      -0.015         0.017        -0.008
      -0.004         0.001         0.023
      -0.003        -0.051         -0.02
       0.013         0.004        -0.047
      -0.013         0.027        -0.033
```

The covariance of the misclosures were then calculated by calculating B then the accuracies of the misclosures extracted.

```
B=zeros(size(w,1)*3,n);
for i=1:size(data,1)/2
    B(3*i-2:3*i,6*i-5:6*i)=[eye(3),eye(3)];
end

for i=1:size(ftLoop,1)
    ind=find(sum(data(:,1:2)==ftLoop(i,:),2)==2);
    B(3*9-2:3*9,3*ind-2:3*ind)=eye(3);
end
Dww=B*DLL*B';
sigmaw=sqrt(diag(Dww));
xyzsigmaw=zeros(size(w));
% reshape sigmaw into x,y,z
for i=1:numel(sigmaw)/3
    xyzsigmaw(i,:)=[sigmaw(3*i-2) sigmaw(3*i-1) sigmaw(3*i)];
end
```

```
xyzsigmaw
```

```
xyzsigmaw = 9×3

   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00070711   0.00056569   0.00098995
   0.00086603   0.00069282    0.0012124
```

The above representation of σ_w and all future representations in x, y, z array form were outputted for better visualization and not used for calculation.

To check for outliers, I assumed the $\frac{w_i}{\sigma_{w_i}} \sim N(0,1)$ and using a significance level of $\alpha = 0.01$ a test for outliers was made where if $\left|\frac{w_i}{\sigma_{w_i}}\right| \leq |z_{0.01/2}| = 2.575$ then the observation passed and was likely not an outlier.

```
normTest=w./xyzsigmaw
a=0.01;
```

```
notOutlier=abs(normTest)<=abs(norminv(a/2)) % two-tail
```

```
notOutlier = 9×3 logical array
      0   0   0
      0   0   0
      0   1   0
      0   0   0
      0   0   0
      0   1   0
      0   0   0
      0   0   0
      0   0   0
```

Observation	Test Stat			Pass/Fail	True=pass	False=fail
w	DX	DY	DZ	DX	DY	DZ
1	-22.6274	21.2132	4.04061	FALSE	FALSE	FALSE
2	-2.82843	8.838835	-48.4873	FALSE	FALSE	FALSE
3	-25.4558	-1.76777	38.3858	FALSE	TRUE	FALSE
4	-41.0122	-61.8718	-78.7919	FALSE	FALSE	FALSE
5	-21.2132	30.05204	-8.08122	FALSE	FALSE	FALSE
6	-5.65685	1.767767	23.23351	FALSE	TRUE	FALSE
7	-4.24264	-90.1561	-20.2031	FALSE	FALSE	FALSE
8	18.38478	7.071068	-47.4772	FALSE	FALSE	FALSE
9	-15.0111	38.97114	-27.2179	FALSE	FALSE	FALSE

All except 2 misclosures and their corresponding double-run measurements seem to be outliers which may be due to the nature of the small errors of the measurements in D_{LL} on the order of magnitude of 0.1mm affecting the calculations for $D_{ww} = BD_{LL}B^T$ making the threshold for a non-outlier very slim.

Task 5

A priori variance factor $\sigma_0^2 = \frac{w^T D_{ww}^{-1} w}{number\ of\ conditions} = \frac{w^T D_{ww}^{-1} w}{27}$

```
w=reshape(w',[numel(w),1])
```

```
w = 27×1
      -0.016
       0.012
       0.004
      -0.002
⋮
```

```
priorVar=w'*inv(Dww)*w/numel(w)
```

```
priorVar =        1140.1
```

To verify what conclusions σ_0^2 is making on the quality of the observations, a test of variance is carried out. Ideally $\sigma_0^2 = 1$,

$$\begin{cases} H_0: \sigma_0^2 = 1 \\ H_A: \sigma_0^2 \neq 1 \end{cases}$$

$$(number\ of\ conditions)\sigma_0^2 \sim \chi^2(number\ of\ conditions)$$

$$(27)\sigma_0^2 = 30783$$

Using significance level $\alpha = 0.01$

```
statTest=27*priorVar
```

```
statTest =         30783
```

```
a=0.01;
lims=chi2inv([a/2 1-a/2],27) % lower and upper confidence interval limits
```

```
lims = 1×2
      11.808       49.645
```

The test statistic $(number\ of\ conditions)\sigma_0^2$ is not in the confidence interval using $\alpha = 0.01$ and the null hypothesis is not accepted therefore the initial estimate of the observation errors D_{LL} are of low quality and unrealistic.

Task 6

After scaling the observation covariance by σ_0^2, the 3x3 covariance that repeatedly appeared on D_{LL}'s diagonal scaled to.

2.9E-04	-2.3E-06	4.6E-06
-2.3E-06	1.8E-04	5.7E-06
4.6E-06	5.7E-06	5.6E-04

To see if the scale factor affected outlier testing, statistical testing using the T-distribution is necessary since scaling the previous test statistic $\frac{w_i}{\sigma_{w_i}}$ by $\frac{1}{\sigma_0}$ (since $D_{ww} = \sigma_0^2 D_{ww}^0$) turns the test statistic into a T-distributed variable i.e. $\frac{w_i}{\sigma_0 \sigma_{w_i}} \sim T(number\ of\ conditions)$. Using the same $\alpha = 0.01$

```
tTestVar=normTest/sqrt(priorVar);
isNotOutlier=abs(tTestVar)<=abs(tinv([a/2],27)) % two-tailed test for outliers, t-Dist is symmetric
isNotOutlier = 9×3 logical array
   1   1   1
   1   1   1
⋮
```

Observation	Test Stat			Pass/Fail	True=pass	False=fail
w	DX	DY	DZ	DX	DY	DZ
1	-0.67013	0.628248	0.119666	TRUE	TRUE	TRUE
2	-0.08377	0.26177	-1.436	TRUE	TRUE	TRUE
3	-0.7539	-0.05235	1.13683	TRUE	TRUE	TRUE
4	-1.21461	-1.83239	-2.33349	TRUE	TRUE	TRUE
5	-0.62825	0.890018	-0.23933	TRUE	TRUE	TRUE
6	-0.16753	0.052354	0.688081	TRUE	TRUE	TRUE
7	-0.12565	-2.67005	-0.59833	TRUE	TRUE	TRUE
8	0.544482	0.209416	-1.40608	TRUE	TRUE	TRUE
9	-0.44457	1.154165	-0.80608	TRUE	TRUE	TRUE

The a priori variance factor scale had an impact on which measurements were considered outliers since the previous non-scaled normal distribution test concluded that all except 2 of the measurements were qualified as outliers and the test statistic scaled by the a priori variance factor concluded that no measurements were outliers.

Part Two

Task 1

To perform the linear parametric adjustment, the only missing defined variable was the constant matrix C for the equation for $\hat{x}$

$$\hat{x} = (A^T PA)^{-1} A^T P(l - C)$$

Where

$$C = \begin{bmatrix} -\overrightarrow{X_A} \\ \overrightarrow{X_A} \\ \vdots \\ 0 \end{bmatrix}$$

Iteratively, this was done by recording when the control point A was a To or From point in the measurement network and the sign of vector $\overrightarrow{X_A}$ in the general observation equation was assigned to `Csign`. For efficiency considerations, the calcA function was updated since it also went through each observation checking the From and To points.

$$\overrightarrow{DX_{ij}} = \overrightarrow{X_j} - \overrightarrow{X_i}$$

```
function [A,Csign] = calcA(from,to)
% assuming coordinate 1 is the control point
u=4;
A=zeros(3,3*u);
Csign=0;
from
to
if from~=1
    A(:,((from-1)*3-2):(from-1)*3)=-eye(3);
else
    Csign=-1;
end
if to~=1
    A(:,((to-1)*3-2):(to-1)*3)=eye(3);
else
    Csign=1;
end
```

The following lines were updated to the main body of code

```
C=zeros(n,1);
XA=[1000;1000;1000];
for i=1:size(data,1)
    [A(i*3-2:i*3,:),Csign]=calcA(FromTo(i,1),FromTo(i,2));
    C(i*3-2:i*3)=Csign*XA;
end
```

Observation vector l was also defined in code as in Part 1 Task 2 and $D_{LL} = \sigma_0^2 D_{LL}^0$

```
priorDLL=DLL*priorVar;
l=reshape(data(:,3:5)',[numel(data(:,3:5)) 1]);
P=inv(priorDLL);
xhat=inv(A'*P*A)*A'*P*(l-C);
xhatxyz=col2xyz(xhat);
```

```
xhatxyz = 4×3
1192.43920000000      1250.66613333333      1121.11790000000
1250.32980000000      936.776033333333      1005.60376666667
854.494700000000      789.561466666666      894.768900000000
763.353800000000      1058.85986666667      967.231433333333
```

The rows represent x, y, z coordinates of B, C, D, E respectively.

Task 2

To calculate residuals, the following relationship was used

$$\hat{v} = \hat{l} - l$$

$$\hat{l} = A\hat{x} + C$$

```
lhat=A*xhat+C;
v=lhat-l;
lhatxyz=col2xyz(lhat);
vxyz=col2xyz(v)
```

```
vxyz = 16×3
      0.0112   -0.0068667       0.0029
      0.0048   -0.0051333      -0.0069
      0.0078   -0.0029667     0.025767
     -0.0058   -0.0020333     0.022233
      0.0107    0.0064667      -0.0161
      0.0073   -0.0054667      -0.0219
      0.0028     0.012867     0.029433
      0.0262     0.022133     0.048567
      0.0076      -0.0181    0.0078667
      0.0074       0.0011   0.00013333
        ⋮
```

To identify outliers a normal distribution test using a significance level $\alpha = 0.01$ for each residual was performed similarly to the misclosures in Part One.

$$\begin{cases} H_0: v_i = 0 \\ H_A: v_i \neq 0 \end{cases}, i = 1,2, \ldots, 48$$

$$\frac{v_i}{\sigma_{v_i}} \sim N(0,1)$$

The standard deviations of the residuals σ_{v_i} were the square root of the diagonals of the covariance matrix $D_{\hat{v}\hat{v}} = D_{LL} - A(A^T PA)^{-1}A$

```
Dvhat=priorDLL-A*inv(A'*inv(priorDLL)*A)*A';
sigmav=sqrt(diag(Dvhat));
xyzsigmav=col2xyz(sigmav);
a=0.01;
vNotOutlier=abs(vxyz./xyzsigmav)<=abs(norminv([a/2]))
```

```
vNotOutlier = 16×3 logical array
true   true   true
true   true   true
true   true   true
true   true   true
⋮
```

Observation		Test Stat			Pass/Fail	True=pass	False=fail
From	To	DX	DY	DZ	DX	DY	DZ
'A'	'B'	0.75765162	-0.58064	0.140127	TRUE	TRUE	TRUE
'B'	'A'	0.32470784	-0.43407	-0.33341	TRUE	TRUE	TRUE
'A'	'C'	0.52765024	-0.25086	1.245036	TRUE	TRUE	TRUE
'C'	'A'	-0.3923553	-0.17194	1.074306	TRUE	TRUE	TRUE
'A'	'D'	0.72382789	0.546817	-0.77795	TRUE	TRUE	TRUE
'D'	'A'	0.4938265	-0.46226	-1.0582	TRUE	TRUE	TRUE
'A'	'E'	0.18941291	1.087997	1.422207	TRUE	TRUE	TRUE
'E'	'A'	1.77236362	1.87158	2.346723	TRUE	TRUE	TRUE
'B'	'C'	0.52567546	-1.56492	0.388657	TRUE	TRUE	TRUE
'C'	'B'	0.5118419	0.095106	0.006587	TRUE	TRUE	TRUE
'C'	'D'	0.61559363	-0.91359	-0.28985	TRUE	TRUE	TRUE
'D'	'C'	-0.3389223	0.827132	-0.84648	TRUE	TRUE	TRUE
'D'	'E'	0.69859502	1.850239	0.91565	TRUE	TRUE	TRUE
'E'	'D'	-0.4910916	2.559209	0.072462	TRUE	TRUE	TRUE
'E'	'B'	-0.6640111	-0.928	1.109979	TRUE	TRUE	TRUE
'B'	'E'	-0.2351706	0.582163	1.212084	TRUE	TRUE	TRUE

None of the residuals were tested statistically to be outliers.

Task 3

The a posteriori variance factor $\hat{\sigma}_0^2 = \frac{\hat{v}^T P \hat{v}}{dof} = \frac{\hat{v}^T P \hat{v}}{48-12} = \frac{\hat{v}^T P \hat{v}}{36}$

```
postVar = v'*P*v/(numel(l)-numel(xhat))
```

```
postVar =
      0.93012
```

To make conclusions on what $\hat{\sigma}_0^2$ says about the observation quality, a chi-square distribution test statistic was used similarly to Part One Task 5 using the same significance level used in the other statistical tests $\alpha = 0.01$. A two-tail or right-tail test can be performed, a right-tail test was performed.

$$\begin{cases} H_0: \hat{\sigma}_0^2 = 1 \\ H_A: \hat{\sigma}_0^2 \neq 1 \end{cases}$$

$$(dof)\hat{\sigma}_0^2 \sim \chi^2(dof)$$

```
postChiStat=v'*P*v
```

```
postChiStat =
       33.484
```

```
chi2inv(a,dof)
```

```
ans =
       19.233
```

```
postIsOutlier=postChiStat>chi2inv(a,dof)
```

```
postIsOutlier = logical
   1
```

The $\hat{\sigma}_0^2$ fails the null hypothesis, $\hat{\sigma}_0^2 \neq 1$ and means a problem with the dataset or model exists such that the solution to the system is affected.

Task 4

Scaling the covariance matrices of the observations and residuals by $\hat{\sigma}_0^2$

```
postDLL=postVar*priorDLL;
postDvhat=postVar*Dvhat
```

```
postDvhat = 48×48
      0.23173   -0.0018539    0.0037077     0.070527 ...
   -0.0018539      0.14831    0.0046346  -0.00056422
    0.0037077    0.0046346      0.45419    0.0011284
     0.070527  -0.00056422    0.0011284      0.23173
  -0.00056422     0.045137    0.0014105   -0.0018539
    0.0011284    0.0014105      0.13823    0.0037077
    -0.030226   0.00024181  -0.00048361     0.030226
   0.00024181    -0.019345  -0.00060452  -0.00024181
  -0.00048361  -0.00060452    -0.059243   0.00048361
     0.030226  -0.00024181   0.00048361    -0.030226
        :
```

The impact of the $\hat{\sigma}_0^2$ on the observations was tested by performing a test using the t-distributed test statistic, $\alpha = 0.01$ and two-tailed test.

$$\frac{\hat{v}_i}{\sigma_{\hat{v}_i}} \sim T(dof), i = 1, \ldots, 48$$

$$\left|\frac{\hat{v}_i}{\sigma_{\hat{v}_i}}\right| \leq |t_{0.005,dof}|$$

Where $\sigma_{\hat{v}_i}$ were the square root of the diagonals of $D_{\hat{v}\hat{v}}$

```
sigmavPost=sqrt(diag(postDvhat));
xyzsigmavPost=col2xyz(sigmavPost);
tTestPost=vxyz./xyzsigmavPost
```

```
tTestPost = 16×3
    0.023266     -0.01783    0.0043031
   0.0099712     -0.01333    -0.010238
    0.016203   -0.0077035     0.038233
   -0.012049   -0.0052799      0.03299
    0.022228     0.016792    -0.023889
    0.015165    -0.014195    -0.032496
   0.0058165     0.033411     0.043674
    0.054426     0.057473     0.072064
    0.016143    -0.048056     0.011935
    0.015718    0.0029205   0.00020229
```

```
tNotOutlier=abs(tTestPost)<abs(tinv(a/2,dof))
```

Observation		Test Stat			Pass/Fail	True=pass	False=fail
From	To	DX	DY	DZ	DX	DY	DZ
'A'	'B'	0.02326619	-0.01783	0.004303	TRUE	TRUE	TRUE

'B'	'A'	0.00997122	-0.01333	-0.01024	TRUE	TRUE	TRUE
'A'	'C'	0.01620324	-0.0077	0.038233	TRUE	TRUE	TRUE
'C'	'A'	-0.0120486	-0.00528	0.03299	TRUE	TRUE	TRUE
'A'	'D'	0.02222752	0.016792	-0.02389	TRUE	TRUE	TRUE
'D'	'A'	0.01516457	-0.0142	-0.0325	TRUE	TRUE	TRUE
'A'	'E'	0.00581655	0.033411	0.043674	TRUE	TRUE	TRUE
'E'	'A'	0.05442626	0.057473	0.072064	TRUE	TRUE	TRUE
'B'	'C'	0.0161426	-0.04806	0.011935	TRUE	TRUE	TRUE
'C'	'B'	0.01571779	0.002921	0.000202	TRUE	TRUE	TRUE
'C'	'D'	0.01890383	-0.02805	-0.0089	TRUE	TRUE	TRUE
'D'	'C'	-0.0104077	0.0254	-0.02599	TRUE	TRUE	TRUE
'D'	'E'	0.02145266	0.056818	0.028118	TRUE	TRUE	TRUE
'E'	'D'	-0.0150806	0.078589	0.002225	TRUE	TRUE	TRUE
'E'	'B'	-0.0203906	-0.0285	0.034086	TRUE	TRUE	TRUE
'B'	'E'	-0.0072217	0.017877	0.037221	TRUE	TRUE	TRUE

Again none of the measurements were statistically tested to be outliers so scaling the covariance matrices by $\hat{\sigma}_0^2$ did not change which measurements were seen as outliers.

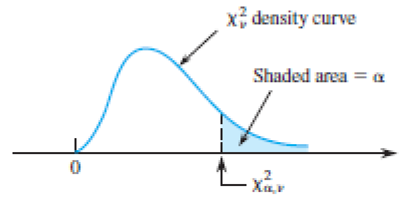
Appendix

A.1 – Data table

From	To	DX (m)	DY (m)	DZ (m)	XX	XY	XZ	YY	YZ	ZZ
A	B	192.428	250.673	121.115	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
B	A	-192.444	-250.661	-121.111	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
A	C	250.322	-63.221	5.578	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
C	A	-250.324	63.226	-5.626	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
A	D	-145.516	-210.445	-105.215	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
D	A	145.498	210.444	105.253	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
A	E	-236.649	58.847	-32.798	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
E	A	236.62	-58.882	32.72	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
B	C	57.883	-313.872	-115.522	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
C	B	-57.898	313.889	115.514	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
C	D	-395.844	-147.204	-110.829	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
D	C	395.84	147.205	110.852	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
D	E	-91.151	269.277	72.444	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
E	D	91.148	-269.328	-72.464	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
E	B	429.095	191.817	153.864	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07
B	E	-429.082	-191.813	-153.911	2.5E-07	-2E-09	4E-09	1.6E-07	5E-09	4.9E-07

A.2 Chi-Square Distribution Critical Values

Table A.7 Critical Values for Chi-Squared Distributions

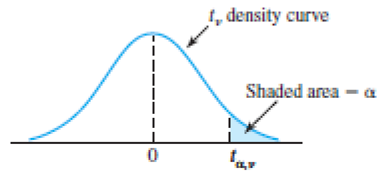


	α									
ν	.995	.99	.975	.95	.90	.10	.05	.025	.01	.005
1	0.000	0.000	0.001	0.004	0.016	2.706	3.843	5.025	6.637	7.882
2	0.010	0.020	0.051	0.103	0.211	4.605	5.992	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.344	12.837
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.832	15.085	16.748
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.440	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.012	18.474	20.276
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.534	20.090	21.954
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.022	21.665	23.587
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.724	26.755
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.041	19.812	22.362	24.735	27.687	29.817
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.600	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.577	32.799
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.407	7.564	8.682	10.085	24.769	27.587	30.190	33.408	35.716
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.843	7.632	8.906	10.117	11.651	27.203	30.143	32.852	36.190	38.580
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.033	8.897	10.283	11.591	13.240	29.615	32.670	35.478	38.930	41.399
22	8.643	9.542	10.982	12.338	14.042	30.813	33.924	36.781	40.289	42.796
23	9.260	10.195	11.688	13.090	14.848	32.007	35.172	38.075	41.637	44.179
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.558
25	10.519	11.523	13.120	14.611	16.473	34.381	37.652	40.646	44.313	46.925
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.807	12.878	14.573	16.151	18.114	36.741	40.113	43.194	46.962	49.642
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.120	14.256	16.147	17.708	19.768	39.087	42.557	45.772	49.586	52.333
30	13.787	14.954	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
31	14.457	15.655	17.538	19.280	21.433	41.422	44.985	48.231	52.190	55.000
32	15.134	16.362	18.291	20.072	22.271	42.585	46.194	49.480	53.486	56.328
33	15.814	17.073	19.046	20.866	23.110	43.745	47.400	50.724	54.774	57.646
34	16.501	17.789	19.806	21.664	23.952	44.903	48.602	51.966	56.061	58.964
35	17.191	18.508	20.569	22.465	24.796	46.059	49.802	53.203	57.340	60.272
36	17.887	19.233	21.336	23.269	25.643	47.212	50.998	54.437	58.619	61.581
37	18.584	19.960	22.105	24.075	26.492	48.363	52.192	55.667	59.891	62.880
38	19.289	20.691	22.878	24.884	27.343	49.513	53.384	56.896	61.162	64.181
39	19.994	21.425	23.654	25.695	28.196	50.660	54.572	58.119	62.426	65.473
40	20.706	22.164	24.433	26.509	29.050	51.805	55.758	59.342	63.691	66.766

For $\nu > 40$, $\chi^2_{\alpha,\nu} \approx \nu\left(1 - \frac{2}{9\nu} + z_\alpha\sqrt{\frac{2}{9\nu}}\right)^3$

A.3 – T-Distribution Critical Values

Table A.5 Critical Values for t Distributions



	α						
ν	.10	.05	.025	.01	.005	.001	.0005
1	3.078	6.314	12.706	31.821	63.657	318.31	636.62
2	1.886	2.920	4.303	6.965	9.925	22.326	31.598
3	1.638	2.353	3.182	4.541	5.841	10.213	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	1.319	1.714	2.069	2.500	2.807	3.485	3.767
24	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
32	1.309	1.694	2.037	2.449	2.738	3.365	3.622
34	1.307	1.691	2.032	2.441	2.728	3.348	3.601
36	1.306	1.688	2.028	2.434	2.719	3.333	3.582
38	1.304	1.686	2.024	2.429	2.712	3.319	3.566
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
50	1.299	1.676	2.009	2.403	2.678	3.262	3.496
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
120	1.289	1.658	1.980	2.358	2.617	3.160	3.373
∞	1.282	1.645	1.960	2.326	2.576	3.090	3.291

A.4 Rearrange Data into D_{LL}

```
% Creates weight matrix


% cell array of functions reshaping a row/line of variances/covariances
% representing a GNSS baseline obs. into a covariance matrix
covar={ @(line)line(6),@(line)line(7),@(line)line(8);
    @(line)line(7),@(line)line(9),@(line)line(10);
    @(line)line(8),@(line)line(10),@(line)line(11)};


u=4; % # of unknown 3D coordinates
n=3*size(data,1); % # of measurements
DLL=zeros(n);
covars=zeros(3,3,size(data,1));
for k = 1:size(data,1)
    for i = 1:3
        for j=1:3
            covars(i,j,k)=covar{i,j}(data(k,:));
        end
    end
    DLL(3*k-2:3*k,3*k-2:3*k)=covars(:,:,k);
end
P=inv(DLL);
```