TECHNICAL UNIVERSITY BERLIN

GEODESY AND GEOINFORMATION SCIENCE

Prof. Dr.-Ing. Frank Neitzel

SELECTED SECTIONS OF ADJUSTMENT CALUCLATION

HOMEWORK II

[Combined Horizontal Network]

SUMMER SEMESTER – 2017

Author: Arghadeep Mazumder

Matriculation Number: 378554

e-mail: arghadeep.mazumder@campus.tu-berlin.de

Contents

List of Tables

TASK I

(Free Network Adjustment)

Objective: The main objective of this task is to perform free net adjustment of a combined horizontal network which has been provided. In this task, all the points contribute to the datum definition. Apart from that, the parameters of the internal and external reliability of the network have to be calculated. After calculation, global test has to be done to evaluate the mathematical and stochastic model. In the calculation, the blunder(s) have to be removed. After removing the blunder(s), the results should be presented in tables so that the network can be determined in terms of reliability and accuracy.

1.1 *Free Network Adjustment*

A horizontal network has been given where the observations are horizontal distances and directions. Along with that control points and new points has been given. The unknows are the x, y coordinates of all control and new points as well as zero directions for each point. Calculation of the coordinates of the points of the network requires three constraints are added to the solution. The network can be arbitrarily shifted in x, y directions and rotated.

1.1.1 Observation Vector and Initial Unknowns

The observation vector, denoted by L, is the vector with 18 horizontal distances [m] and 18 directions [gon] which is provided. There is total 36 number of observations out of which 21 unknowns are there and 3 constraints. So the redundancy will be 36 – 21 +3 = 18

$$L = \begin{bmatrix} s_{1000,100} \\ s_{1000,102} \\ s_{2000,103} \\ s_{2000,101} \\ s_{3000,100} \\ s_{3000,103} \\ s_{100,1000} \\ s_{100,102} \\ s_{100,2000} \\ s_{100,3000} \\ s_{101,103} \\ s_{101,2000} \\ s_{102,1000} \\ s_{102,100} \\ s_{102,2000} \\ s_{103,3000} \\ s_{103,2000} \\ s_{103,101} \\ r_{1000,100} \\ r_{1000,102} \\ r_{2000,103} \\ r_{2000,101} \\ r_{3000,100} \\ r_{3000,103} \\ r_{100,1000} \\ r_{100,102} \\ r_{100,2000} \\ r_{100,3000} \\ r_{101,103} \\ r_{101,2000} \\ r_{102,1000} \\ r_{102,100} \\ r_{102,2000} \\ r_{103,3000} \\ r_{103,2000} \\ r_{103,101} \end{bmatrix} = \begin{bmatrix} 201.941 \\ 175.940 \\ 106.177 \\ 175.288 \\ 93.728 \\ 122.506 \\ 201.941 \\ 121.468 \\ 207.826 \\ 93.727 \\ 222.323 \\ 175.287 \\ 175.942 \\ 121.466 \\ 200.334 \\ 122.507 \\ 106.185 \\ 222.325 \\ 269.6980 \\ 310.4634 \\ 207.9866 \\ 320.7754 \\ 69.2435 \\ 311.3932 \\ 69.8977 \\ 3.3149 \\ 326.2594 \\ 269.1427 \\ 151.8806 \\ 120.8758 \\ 110.3651 \\ 203.0060 \\ 287.5533 \\ 11.2898 \\ 7.8869 \\ 351.6806 \end{bmatrix}$$

$$X^0 = \begin{bmatrix} y_{1000} \\ x_{1000} \\ y_{2000} \\ x_{2000} \\ y_{3000} \\ x_{3000} \\ y_{100} \\ x_{100} \\ y_{101} \\ x_{101} \\ y_{102} \\ x_{102} \\ y_{103} \\ x_{103} \\ \omega_{1000} \\ \omega_{2000} \\ \omega_{3000} \\ \omega_{100} \\ \omega_{101} \\ \omega_{102} \\ \omega_{103} \end{bmatrix}$$

1.1.2 Functional Model

The functional model for distance measurement is given by:

$$s_{ij} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \ldots \tag{1}$$

The functional model for direction is given by:

$$r_{ij} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \ldots \tag{2}$$

where, ω_i is the zero directions for each point.

1.1.3 Observation Equation

The observation equations can be written as:

$$s_{ij} + v_{s_{ij}} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \ldots \tag{3}$$

and for directions the equation can be written as:

$$r_{ij} + v_{r_{ij}} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \ldots \tag{4}$$

where, $v_{s_{ij}}$ and $v_{r_{ij}}$ are the residuals.

1.1.4 Datum Definition

We need to perform an adjustment of the combined horizontal network while all control points are contributing to the datum definition. In this task, we need 3 constraints, they are :

$$\sum_{i=1}^{p} \widehat{x_i} = 0 \ldots \tag{5}$$

$$\sum_{i=1}^{p} \widehat{y_i} = 0 \quad \text{... (6)}$$

$$\sum_{i=1}^{p} (y_i'\widehat{x_i} - x_i'\widehat{y_i}) = 0 \quad \text{... (7)}$$

In the problem sheet, it is said that we have to use all control points for datum definition. So, the datum will be

$$datum = [1\ 1\ 1\ 0\ 0\ 0\ 0] \quad \text{... (8)}$$

1.1.5 Stochastic Model

It is mentioned that the standard deviation for the distance measurement is 1mm and for direction is 1mgon. So,

Distance Measurement [σ_s] = 0.001 m

Direction Measurement [σ_r] = 0.001 $\times \frac{\pi}{200} = 1.570796 \times 10^{-5}$ rad

Again, it is mentioned that $\sigma_0 = 1$. So, the Stochastic model is a 36×36 matrix which can be written as:

$$\sum_{LL} = \begin{bmatrix} \sigma^2_{s_{1000,100}} & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2_{s_{1000,102}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{r_{103,2000}} & 0 \\ 0 & 0 & \cdots & 0 & \sigma^2_{s_{103,101}} \end{bmatrix} \quad \text{... (9)}$$

$$Q_{ll} = \frac{1}{\sigma_0^2} \sum_{LL} \quad \text{... (10)}$$

$$P = Q_{ll}^{-1} \quad \text{... (11)}$$

1.1.6 Unknown coordinates with respect to centroid

The initial coordinates of the unknowns have to be reduced to the centroid. So, the centroid has to be calculated. It can be written as:

$$x_{centroid} = \frac{1}{n} \sum_{i=1}^{n} x_i^o \quad \text{... (12)}$$

$$y_{centroid} = \frac{1}{n} \sum_{i=1}^{n} y_i^o \quad \text{... (13)}$$

where, n is the given number of points. In this case, the n is equal to 3 because the number of control points given is 3. x_i^o and y_i^o are the initial value of the coordinates of the unknown points. Now, it is required to perform the reduction of the coordinates of the unknown points.

$$x_i^{new} = x_i^o - x_{centroid} \quad \text{... (14)}$$

$$y_i^{new} = y_i^o - y_{centroid} \quad \text{... (15)}$$

Now, x_i^{new} and y_i^{new} can be used for adjustment.

1.1.7 Constraints

As it is mentioned previously, that the network can be shifted arbitrarily in x, y directions and can also be rotated arbitrarily. So, taking some constraints into account.

For shifting in x-direction

$$\sum_{i=1}^{n} \hat{x_i} = 0 \quad ...(16)$$

For shifting in y-direction

$$\sum_{i=1}^{n} \hat{y_i} = 0 ... \quad (17)$$

and for rotation

$$\sum_{i=1}^{p} (y_i^{new} \hat{x_i} - x_i^{new} \hat{y_i}) = 0 \quad ...(18)$$

1.1.8 B-Matrix

B matrix consist of partial derivatives of the constraints with respect to the unknowns. So, B matrix is a 3×21 matrix as shown below:

$$B^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 \\ -x_{1000}^{new} & y_{1000}^{new} & -x_{2000}^{new} & y_{2000}^{new} & -x_{3000}^{new} & y_{3000}^{new} & 0 & \cdots & 0 \end{bmatrix} \quad ...(19)$$

1.1.9 Design Matrix

The design matrix consists of partial derivatives of the equation with respect to the unknowns.

$$A_{36\times21} = \begin{bmatrix} \frac{\partial_{s_{1000,100}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{s_{1000,100}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{r_{103,101}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{r_{103,101}}}{\partial_{\omega_{103}}} \end{bmatrix} \quad ...(20)$$

1.1.10 Adjusted Unknowns

To compute the adjusted unknowns, the following procedure has been used.

The normal equation system is given by:

$$N = A^T P A \quad ...(21)$$

Now, from the B matrix, the new matrix N_{ext} in matrix notation is

$$N_{ext} = \begin{bmatrix} N & \vdots & B^t \\ \ldots & \ldots & \ldots \\ B & \vdots & 0 \end{bmatrix} \quad ...(22)$$

and the vector of absolute values:

$$n = A^T Pl \qquad \text{... (23)}$$

where, l is the observation vector.

Now, due to changes in the N matrix it is required to change the n matrix also.

$$n_{ext} = [n\ 0\ 0\ 0]^T \qquad \text{... (24)}$$

The corrections $\hat{x}$ of the unknowns are calculated equation:

$$\hat{x} = Q_{ext} n_{ext} \qquad \text{... (25)}$$

where,

$$Q_{ext} = N_{ext}^{-1} \qquad \text{... (26)}$$

The unknowns are given by the equation:

$$X_{final} = X_0 + \hat{x} \qquad \text{... (27)}$$

where, X_0 are the adjusted unknowns.

1.2 Calculation of parameters for Internal and External Reliability (EV, NV, GF, GRZW, EGK and EP)

1.2.1 Internal Reliability

To compute the internal reliability, the assumptions should be taken as

$$r_i = diag(Q_{vv} P) \qquad \text{... (28)}$$

where, Q_{vv} is the co-factor matrix of the residuals.

According to this, the redundancy is calculated by

$$EV = 100 r_i \qquad \text{... (29)}$$

$0\% \leq EV < 1\%$: Observation not controlled,

$1\% \leq EV < 10\%$: Observation poorly controlled,

$10\% \leq EV < 30\%$: Observation sufficiently controlled,

$30\% \leq EV < 70\%$: Observation sufficiently controlled,

$70\% \leq EV < 100\%$: Observation can be dismissed without loss of reliability

The standardized residuals can be calculated by the equation

$$NV = \frac{|v|}{\sigma_v} \qquad \text{... (30)}$$

where, $\sigma_v = \sigma_0 \sqrt{q_{v_i v_i}}$

The following scale can be taken as a basis for the evaluation of the standardized residuals:

$NV < 2.5$: No blunder distinguishable,

$2.5 \leq NV < 4.0$: Blunder possible,

$4.0 \leq NV$: Blunder is highly probable

The potential magnitude of gross error is calculated as shown below:

$$GF_i = \frac{-v_i}{r_i} \qquad \text{... (31)}$$

Since, $\delta_0 = 4.13$, the boundary value for the blunders is

$$GRZW_i = \frac{\sigma_0 \delta_0}{\sqrt{r_i P_i}} \qquad \text{... (32)}$$

1.2.2 External Reliability

It describes the influence of observation errors on the parameters which have to be determined. In this case, the parameters are coordinates and zero directions.

The impact of the boundary value for the blunders on the coordinates of the corresponding points is described by the value

$$EGK_i = (1 - r_i)GRZW\ _i \qquad \text{... (33)}$$

The impact of a potential magnitude of the blunder GF_i on a point corresponding to the measurement is described by the value:

$$EP_i = (1 - r_i)GF_i \qquad \text{... (34)}$$

1.3 Functional and Stochastic Model Evaluation using Global Test

The functional and stochastic model have to be evaluated, according to an appropriate statistical test. The evaluation will be performed according to χ^2 test. For approaching, a null and alternative hypothesis are constructed.

Null Hypothesis:

$$H_0: E\{s_0^2\} = \sigma_0^2 \qquad \text{... (35)}$$

Alternative Hypothesis:

$$H_A: E\{s_0^2\} < \sigma_0^2 \qquad \text{... (36)}$$

In the case that the null hypothesis is accepted, it means that the stochastic model is considered to be chosen correctly a priori, since σ_0^2 is almost equal to the posteriori value s_0^2. However, if null hypothesis cannot be accepted which means that the observation contain blunders or the functional model is maybe incomplete/inappropriate or the stochastic model is inappropriate.

The test is performed according to the equation

$$T_{\chi^2} = \frac{f s_0^2}{\sigma_0^2} \qquad \text{... (37)}$$

Now for further approach, the following considerations should be considered.

Confidence level $S = 95\%$ i.e. $\alpha = 0.05$

Degree of freedom f = 18

Threshold taken from the table $\chi^2_{f,\alpha} = 28.869$

Now, as it is defined that $T_{\chi^2} < \chi^2_{f,\alpha}$ then the null hypothesis can be accepted. Otherwise the null hypothesis fails to be accepted and alternative hypothesis should be considered.

For the performed adjustment the test statistic value is $T_{\chi^2} = 102.1625$ which is greater than $\chi^2_{f,\alpha}$. With the probability of 95% the null hypothesis is incorrect and the alternative hypothesis has to be accepted. So, it can be said that the functional model or stochastic models are inappropriate or the observation contain blunders.

1.4 Blunder Detection

Now as previously mentioned that the null hypothesis is incorrect and alternative hypothesis has be taken into account, means there is error in functional model or stochastic model or observation matrix contains blunder. Before checking the functional model or stochastic model it will be appropriate to go through observation and search blunder in that.

To detect the blunder from the observation it is required to process the data, for that the following steps are required.

Calculation of free network.

Performing Global test

If the Null hypothesis is accepted, then there are no blunders and the final result is calculated.

If the Null hypothesis is not accepted, then the observation with the largest NV_i and also with $NV_i > 4$ should be assumed as a blunder.

If the observation contains any blunder it should be removed from the table and the process should be repeated until and unless the observation table is error free. In this case, after performing the first calculation it was found that the blunder is in $r_{100,102}$. This blunder was removed from the table and performed the test again to detect any more possible blunder(s). The design matrix has been changed and again performed $T_{\chi^2} = 47.210 > \chi^2_{f,\alpha} = 27.58$ So, performing the test it is clear that the observation vector still contains blunder. After removing the blunder, the test has been done again to detect possible blunder(s). After this iteration the global test showed that $T_{\chi^2} = 14.348 < \chi^2_{f,\alpha} = 26.29$. This means that null hypothesis can be accepted and can be calculated for final results.

NV 1st Iteration	**NV 2nd Iteration**	**NV 3rd Iteration**
0.514793991	0.232926088	0.205090219
0.664273722	1.635290579	1.589154707
5.958806358	5.732582538	1.126502369
0.86351705	0.835508542	0.465444938
0.568994506	0.44987149	1.06516031
1.252450279	1.206197926	0.205090219
0.514793991	0.232926088	1.250230252
1.520129478	1.30319112	1.852209194
4.334867226	2.339290514	0.93101828
0.827282663	0.94658556	1.828044799
1.458755289	1.428172718	0.192535917
0.453823921	0.481841608	1.10620316

2.008426215	1.059979718	1.455227141
1.184047548	1.402150655	0.862832235
3.958454168	0.980568129	0.306305144
0.118569658	0.164848585	0.63628666
4.820795655	5.052164738	0.800170955
1.163167616	1.193772113	0.880966861
0.778017443	0.872380106	0.880966861
0.778017443	0.872380106	0.792008439
0.24039581	0.281120483	0.792008439
0.24039581	0.281120483	1.174575195
0.303300802	0.940066721	1.174575195
0.303300802	0.940066721	0.021644708
3.368907733	0.265503267	0.166980824
7.412963313	0.662950609	0.169428453
1.922216668	1.023044617	0.631200242
2.622332119	0.241925434	0.631200242
0.201609352	0.241925434	0.96310317
0.201609352	0.870021058	0.134012736
0.95054077	0.068655457	1.123232262
0.806432974	0.958054676	1.563418557
0.092685801	2.110586665	0.471590138
2.247430138	0.774969429	0.782862124
0.832293312	0.920257146	
0.972847708		

Table 1. Comparison Table for Standardized Residuals

T_{χ^2} **after 1st Iteration**	T_{χ^2} **after 2nd Iteration**	T_{χ^2} **after 3rd Iteration**
102.1625353	47.21052239	14.34796356

Table 2. Comparison Table for T_{χ^2}

1.5 Observation Table

The following tables present data for the performed adjustments.

Obs (m)	Residuals (mm)	Adjusted Obs (m)	Std. Dev of Obs (mm)	Std. Dev of Adjusted Obs (mm)	EV %	GF (mm)	GRZW (mm)	EGK (mm)	EP (mm)
201.941	-0.153	201.941	0.708	0.629	55.903	0.274	5.524	2.436	0.121
175.940	1.179	175.941	0.703	0.635	55.059	-2.142	5.566	2.501	-0.962
175.288	-0.854	175.287	0.718	0.618	57.476	1.486	5.448	2.317	0.632
93.728	-0.333	93.728	0.678	0.661	51.279	0.650	5.767	2.810	0.317
122.506	0.777	122.507	0.690	0.648	53.166	-1.461	5.664	2.653	-0.684
201.941	-0.153	201.941	0.708	0.629	55.903	0.274	5.524	2.436	0.121
121.468	-0.924	121.467	0.700	0.638	54.649	1.691	5.587	2.534	0.767
207.826	-1.021	207.825	0.522	0.790	30.385	3.360	7.492	5.216	2.339
93.727	0.667	93.728	0.678	0.661	51.279	-1.300	5.767	2.810	-0.633
222.323	1.391	222.324	0.721	0.614	57.908	-2.402	5.427	2.284	-1.011
175.287	0.146	175.287	0.718	0.618	57.476	-0.254	5.448	2.317	-0.108
175.942	-0.821	175.941	0.703	0.635	55.059	1.491	5.566	2.501	0.670
121.466	1.076	121.467	0.700	0.638	54.649	-1.969	5.587	2.534	-0.893
200.334	0.363	200.334	0.398	0.859	17.698	-2.051	9.817	8.080	-1.688
122.507	-0.223	122.507	0.690	0.648	53.166	0.420	5.664	2.653	0.197
106.185	-0.272	106.185	0.405	0.856	18.264	1.489	9.664	7.899	1.217
222.325	-0.609	222.324	0.721	0.614	57.908	1.052	5.427	2.284	0.443
(gon)	**(mgon)**	**(gon)**	**(mgon)**	**(mgon)**	**%**	**(mgon)**	**(mgon)**	**(mgon)**	**(mgon)**
269.698	-0.611	269.697	0.657	0.682	48.168	1.269	5.951	22.015	4.696
310.463	0.611	310.464	0.657	0.682	48.168	-1.269	5.951	19.181	-4.091
207.987	0.488	207.987	0.583	0.746	37.919	-1.286	6.707	99.260	-19.035
320.775	-0.488	320.775	0.583	0.746	37.919	1.286	6.707	142.025	27.236
69.244	0.584	69.244	0.471	0.822	24.735	-2.362	8.304	196.645	-55.926
311.393	-0.584	311.393	0.471	0.822	24.735	2.362	8.304	257.022	73.097
69.898	0.015	69.898	0.635	0.703	44.907	-0.032	6.163	270.812	-1.419
326.259	-0.126	326.259	0.713	0.623	56.693	0.222	5.485	113.694	4.597
269.143	0.111	269.143	0.622	0.714	43.094	-0.258	6.291	138.999	-5.702
151.881	0.433	151.881	0.649	0.689	46.997	-0.921	6.024	40.218	-6.147
120.876	-0.433	120.875	0.649	0.689	46.997	0.921	6.024	31.709	4.846
110.365	-0.701	110.364	0.689	0.650	52.948	1.324	5.676	136.994	31.947
203.006	-0.103	203.006	0.729	0.605	59.218	0.174	5.367	48.559	1.576
287.553	0.804	287.554	0.678	0.661	51.227	-1.569	5.770	178.478	-48.540
111.290	0.907	111.291	0.549	0.771	33.630	-2.696	7.122	288.236	-109.112
7.887	-0.348	7.887	0.699	0.638	54.557	0.638	5.591	71.895	8.209
351.681	-0.558	351.680	0.675	0.664	50.861	1.098	5.791	203.495	38.574

Table 3. Observation Table for Distance and Direction

1.6 Unknowns and Standard Deviation

ID	Easting [m]	Std. Dev of Easting [mm]	Northing [m]	Std. Dev of Northing [mm]
1000	4590337.392	0.4870	5820823.663	0.2805
2000	4589967.53	0.5489	5820806.109	0.4980
3000	4590078.015	0.5244	5820681.744	0.6708
100	4590159.873	0.4950	5820727.398	0.8132
101	4589800.108	1.3560	5820858.029	2.3253
102	4590163.259	0.5931	5820848.818	0.9099
103	4589956.945	0.7625	5820700.453	0.8270

Table 4. Estimated Co-ordinates

ID	Reference Angle [gon]	Std. Dev of Ref Angle [mgon]
1000	398.669	0.7364
2000	398.369	0.9840
3000	398.368	0.7804
100	398.469	0.5749
101	398.269	1.0622
102	398.769	0.5668
103	398.470	0.7531

Table 5. Estimated Reference Angle

TASK II

(Adjustment with Stochastic Datum)

Objective: The main objective of this task is to perform adjustment with stochastic datum of a combined horizontal network which has been provided. In this task, the observations are used from the previous task while removing the blunders. The parameters of the internal and external reliability of the network have to be calculated. After calculation, global test has to be done to evaluate the mathematical and stochastic model. Apart from that, the control points in this case have to be checked to identify if there is any blunder present or not. After removing the blunder(s), the results should be presented in tables so that the network can be determined in terms of reliability and accuracy.

2.1 Free Network Adjustment

2.1.1 Observation Vector and Initial Unknowns

In this task, from the observation $s_{2000,\mathbf{103}}$ and $r_{100,102}$ has been removed in order to make the observation blunder free.

$$L = \begin{bmatrix} s_{1000,100} \\ s_{1000,102} \\ s_{2000,103} \\ s_{2000,101} \\ s_{3000,100} \\ s_{3000,103} \\ s_{100,1000} \\ s_{100,102} \\ s_{100,2000} \\ s_{100,3000} \\ s_{101,103} \\ s_{101,2000} \\ s_{102,1000} \\ s_{102,100} \\ s_{102,2000} \\ s_{103,3000} \\ s_{103,2000} \\ s_{103,101} \\ r_{1000,100} \\ r_{1000,102} \\ r_{2000,103} \\ r_{2000,101} \\ r_{3000,100} \\ r_{3000,103} \\ r_{100,1000} \\ r_{100,102} \\ r_{100,2000} \\ r_{100,3000} \\ r_{101,103} \\ r_{101,2000} \\ r_{102,1000} \\ r_{102,100} \\ r_{102,2000} \\ r_{103,3000} \\ r_{103,2000} \\ r_{103,101} \\ y_{1000} \\ x_{1000} \\ y_{2000} \\ x_{2000} \\ y_{3000} \\ x_{3000} \end{bmatrix} = \begin{bmatrix} 201.941 \\ 175.940 \\ 106.177 \\ 175.288 \\ 93.728 \\ 122.506 \\ 201.941 \\ 121.468 \\ 207.826 \\ 93.727 \\ 222.323 \\ 175.287 \\ 175.942 \\ 121.466 \\ 200.334 \\ 122.507 \\ 106.185 \\ 222.325 \\ 269.6980 \\ 310.4634 \\ 207.9866 \\ 320.7754 \\ 69.2435 \\ 311.3932 \\ 69.8977 \\ 3.3149 \\ 326.2594 \\ 269.1427 \\ 151.8806 \\ 120.8758 \\ 110.3651 \\ 203.0060 \\ 287.5533 \\ 11.2898 \\ 7.8869 \\ 351.6806 \\ 4590337.39 \\ 5820823.642 \\ 4589967.526 \\ 5820806.067 \\ 4590078.021 \\ 5820681.807 \end{bmatrix}$$

Now the observation vector in this case

$$X^0 = \begin{bmatrix} y_{1000} \\ x_{1000} \\ y_{2000} \\ x_{2000} \\ y_{3000} \\ x_{3000} \\ y_{100} \\ x_{100} \\ y_{101} \\ x_{101} \\ y_{102} \\ x_{102} \\ y_{103} \\ x_{103} \\ \omega_{1000} \\ \omega_{2000} \\ \omega_{3000} \\ \omega_{100} \\ \omega_{101} \\ \omega_{102} \\ \omega_{103} \end{bmatrix}$$

2.1.2 Functional Model

The functional model for distance measurement is given by:

$$s_{ij} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \qquad \ldots (38)$$

The functional model for direction is given by:

$$r_{ij} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \qquad \ldots (39)$$

where, ω_i is the zero directions for each point.

For the co-ordinate of the 3 control points

$$x_i = x_i \qquad \ldots (40)$$

$$y_i = y_i \qquad \ldots (41)$$

2.1.3 Observation Equation

The observation equations can be written as:

$$s_{ij} + v_{s_{ij}} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \qquad \ldots (42)$$

and for directions the equation can be written as:

$$r_{ij} + v_{r_{ij}} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \qquad \ldots (43)$$

where, $v_{s_{ij}}$ and $v_{r_{ij}}$ are the residuals.

For the co-ordinate of the 3 control points

$$x_i + v_{x_i} = x_i \qquad \ldots (44)$$

$$y_i + v_{y_i} = y_i \qquad \ldots (45)$$

2.1.4 Stochastic Model

It is mentioned that the standard deviation for the distance measurement is 1mm and for direction is 1mgon. So,

Distance Measurement [σ_s] = 0.001 m

Direction Measurement [σ_r] = $0.001 \times \frac{\pi}{200} = 1.570796 \times 10^{-5}$ rad

Control Point Measurement [σ_{cp}] = 0.01 m

Again, it is mentioned that $\sigma_0 = 1$. So, the Stochastic model is a 36×36 matrix which can be written as:

$$\sum_{LL} = \begin{bmatrix} \sigma^2_{s_{1000,100}} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2_{s_{1000,102}} & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{r_{103,2000}} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & \sigma^2_{s_{103,101}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sigma^2_{y_{3000}} & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \sigma^2_{x_{3000}} \end{bmatrix} \qquad \ldots (46)$$

$$Q_{ll} = \frac{1}{\sigma_0^2} \sum_{LL} \qquad \ldots (47)$$

$$P = Q_{ll}^{-1} \qquad \ldots (48)$$

2.1.5 Design Matrix

The design matrix consists of partial derivatives of the equation with respect to the unknowns.

$$A = \begin{bmatrix} \frac{\partial_{s_{1000,100}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{s_{1000,100}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{s_{103,101}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{s_{103,101}}}{\partial_{\omega_{103}}} \\ \frac{\partial_{r_{1000,100}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{r_{1000,100}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{r_{103,101}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{r_{103,101}}}{\partial_{\omega_{103}}} \\ \frac{\partial_{y_{1000}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{y_{1000}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{x_{3000}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{x_{3000}}}{\partial_{\omega_{103}}} \end{bmatrix} \qquad \ldots (49)$$

2.1.6 Adjusted Unknowns

To compute the adjusted unknowns, the following procedure has been used.

The normal equation system is given by:

$$N = A^T PA \qquad \text{... (50)}$$

and the vector of absolute values:

$$n = A^T Pl \qquad \text{... (51)}$$

where, l is the observation vector.

The corrections $\hat{x}$ of the unknowns are calculated equation:

$$\hat{x} = Qn \qquad \text{... (52)}$$

where,

$$Q_{ext} = N^{-1} \qquad \text{... (53)}$$

The unknowns are given by the equation:

$$X_{final} = X_0 + \hat{x} \qquad \text{... (54)}$$

where, X_0 are the adjusted unknowns.

2.2 Calculation of parameters for Internal and External Reliability (EV, NV, GF, GRZW, EGK and EP)

1.2.1 Internal Reliability

To compute the internal reliability, the assumptions should be taken as

$$r_i = diag(Q_{vv}P) \qquad \text{... (55)}$$

where, Q_{vv} is the co-factor matrix of the residuals.

According to this, the redundancy is calculated by

$$EV = 100r_i \qquad \text{... (56)}$$

$0\% \leq EV < 1\%$: Observation not controlled,

$1\% \leq EV < 10\%$: Observation poorly controlled,

$10\% \leq EV < 30\%$: Observation sufficiently controlled,

$30\% \leq EV < 70\%$: Observation sufficiently controlled,

$70\% \leq EV < 100\%$: Observation can be dismissed without loss of reliability

The standardized residuals can be calculated by the equation

$$NV = \frac{|v|}{\sigma_v} \qquad \text{... (57)}$$

where, $\sigma_v = \sigma_0\sqrt{q_{v_i v_i}}$

The following scale can be taken as a basis for the evaluation of the standardized residuals:

$NV < 2.5$: No blunder distinguishable,

$2.5 \leq NV < 4.0$: Blunder possible,

$4.0 \leq NV$: Blunder is highly probable

The potential magnitude of gross error is calculated as shown below:

$$GF_i = \frac{-v_i}{r_i} \qquad \text{... (58)}$$

Since, $\delta_0 = 4.13$, the boundary value for the blunders is

$$GRZW_i = \frac{\sigma_0 \delta_0}{\sqrt{r_i P_i}} \qquad \text{... (59)}$$

1.2.2 External Reliability

It describes the influence of observation errors on the parameters which have to be determined. In this case, the parameters are coordinates and zero directions.

The impact of the boundary value for the blunders on the coordinates of the corresponding points is described by the value

$$EGK_i = (1 - r_i) GRZW_i \qquad \text{... (60)}$$

The impact of a potential magnitude of the blunder GF_i on a point corresponding to the measurement is described by the value:

$$EP_i = (1 - r_i) GF_i \qquad \text{... (61)}$$

2.2 Functional and Stochastic Model Evaluation using Global Test

The functional and stochastic model have to be evaluated, according to an appropriate statistical test. The evaluation will be performed according to χ^2 test. For approaching, a null and alternative hypothesis are constructed.

Null Hypothesis:

$$H_0: E\{s_0^2\} = \sigma_0^2 \qquad \text{... (62)}$$

Alternative Hypothesis:

$$H_A: E\{s_0^2\} < \sigma_0^2 \qquad \text{... (63)}$$

In the case that the null hypothesis is accepted, it means that the stochastic model is considered to be chosen correctly a priori, since σ_0^2 is almost equal to the posteriori value s_0^2. However, if null hypothesis cannot be accepted which means that the observation contains blunders or the functional model is maybe incomplete/inappropriate or the stochastic model is inappropriate.

The test is performed according to the equation

$$T_{\chi^2} = \frac{f s_0^2}{\sigma_0^2} \qquad \text{... (64)}$$

Now for further approach, the following considerations should be considered.

Confidence level $S = 95\%$ i.e. $\alpha = 0.05$

Degree of freedom f = 19

Threshold taken from the table $\chi^2_{f,\alpha} = 30.14$

Now, as it is defined that $T_{\chi^2} < \chi^2_{f,\alpha}$ then the null hypothesis can be accepted. Otherwise the null hypothesis fails to be accepted and alternative hypothesis should be considered.

For the performed adjustment the test statistic value is $T_{\chi^2} = 75.49170035$ which is greater than $\chi^2_{f,\alpha}$. With the probability of 95% the null hypothesis is incorrect and the alternative hypothesis has to be accepted. So, it can be said that the functional model or stochastic models are inappropriate or the observation contain blunders.

2.3 Blunder Detection

Now as previously mentioned that the null hypothesis is incorrect and alternative hypothesis has be taken into account, means there is error in functional model or stochastic model or observation matrix contains blunder. Before checking the functional model or stochastic model it will be appropriate to go through observation and search blunder in that.

To detect the blunder from the observation it is required to process the data, for that the following steps are required.

Calculation of free network.

Performing Global test

If the Null hypothesis is accepted, then there are no blunders and the final result is calculated.

If the Null hypothesis is not accepted, then the observation with the largest NV_i and also with $NV_i > 4$ should be assumed as a blunder.

If the observation contains any blunder it should be removed from the table and the process should be repeated until and unless the observation table is error free. In this case, after performing the first calculation it was found that the blunder is in control point 3000 . After removing the blunder, the test has been done again to detect possible blunder(s). After this iteration the global test showed that $T_{\chi^2} = 14.3784 < \chi^2_{f,\alpha} = 30.14$. This means that null hypothesis can be accepted and can be calculated for final results.

NV after 1st Iteration	NV after 2nd Iteration
1.719984492	1.593670154
1.080341818	1.126629663
0.68159147	0.465175859
0.980503527	1.065510093
0.372801652	0.201835906
1.349111667	1.252094153
2.020126496	1.842523725
0.712932894	0.931285707
1.781273288	1.828150211
0.238659298	0.192408271
0.973807511	1.100479521

1.35596059	1.453196078
1.327823643	0.876486237
0.390816203	0.305952575
1.099656483	0.636186625
0.846869695	0.80006506
0.875292749	0.880146769
0.875292749	0.880146769
0.754212402	0.791920854
0.754212402	0.791920854
1.673914721	1.17611226
1.673914721	1.17611226
0.122447902	0.024550857
0.433978419	0.169177189
0.372847782	0.168992527
0.585347883	0.631192757
0.585347883	0.631192757
0.810096432	0.962690652
0.121222512	0.133613362
0.953880364	1.122373385
1.242106586	1.564832199
0.335083981	0.472229602
0.664315953	0.783387446
0.277856085	0.17461909
5.457267354	0.17461909
0.436797219	0.17461909
6.93757976	0.17461909
0.758092536	
7.817090378	

Table 6. Comparison Table for Standardized Residuals

T_{χ^2} **after 1st Iteration**	T_{χ^2} **after 2nd Iteration**
75.49170035	14.37845538

Table 7. Comparison Table for T_{χ^2}

2. 5 Observation Table

The following tables present data for the performed adjustments.

Obs (m)	Residuals (mm)	Adjusted Obs (m)	Std. Dev of Obs (mm)	Std. Dev of Adjusted Obs (mm)	EV %	GF (mm)	GRZW (mm)	EGK (mm)	EP (mm)
201.941	-0.151	201.941	0.688	0.611	55.922	0.270	5.523	2.434	0.119
175.940	1.183	175.941	0.683	0.616	55.108	-2.147	5.563	2.498	-0.964
175.288	-0.854	175.287	0.697	0.600	57.476	1.486	5.448	2.317	0.632
93.728	-0.333	93.728	0.659	0.642	51.279	0.650	5.767	2.810	0.316
122.506	0.777	122.507	0.671	0.629	53.166	-1.461	5.664	2.653	-0.684
201.941	-0.151	201.941	0.688	0.611	55.922	0.270	5.523	2.434	0.119
121.468	-0.926	121.467	0.680	0.619	54.655	1.694	5.586	2.533	0.768
207.826	-1.017	207.825	0.507	0.767	30.446	3.339	7.485	5.206	2.323
93.727	0.667	93.728	0.659	0.642	51.279	-1.301	5.767	2.810	-0.634
222.323	1.391	222.324	0.700	0.597	57.908	-2.402	5.427	2.284	-1.011
175.287	0.146	175.287	0.697	0.600	57.476	-0.254	5.448	2.317	-0.108
175.942	-0.817	175.941	0.683	0.616	55.108	1.482	5.563	2.498	0.665
121.466	1.074	121.467	0.680	0.619	54.655	-1.966	5.586	2.533	-0.891
200.334	0.371	200.334	0.389	0.833	17.899	-2.072	9.762	8.014	-1.701
122.507	-0.223	122.507	0.671	0.629	53.166	0.420	5.664	2.653	0.197
106.185	-0.272	106.185	0.393	0.831	18.264	1.489	9.664	7.899	1.217
222.325	-0.609	222.324	0.700	0.597	57.908	1.051	5.427	2.284	0.443
(gon)	**(mgon)**	**(gon)**	**(mgon)**	**(mgon)**	**%**	**(mgon)**	**(mgon)**	**(mgon)**	**(mgon)**
269.698	-0.611	269.697	0.638	0.662	48.169	1.268	5.951	22.003	4.689
310.463	0.611	310.464	0.638	0.662	48.169	-1.268	5.951	19.170	-4.085
207.987	0.488	207.987	0.566	0.725	37.919	-1.286	6.707	99.260	-19.033
320.775	-0.488	320.775	0.566	0.725	37.919	1.286	6.707	142.025	27.233
69.244	0.585	69.244	0.457	0.798	24.737	-2.365	8.304	196.621	-55.992
311.393	-0.585	311.393	0.457	0.798	24.737	2.365	8.304	256.991	73.184
69.898	0.016	69.898	0.616	0.683	44.920	-0.037	6.162	270.619	-1.609
326.259	-0.127	326.259	0.693	0.605	56.702	0.225	5.485	113.581	4.653
269.143	0.111	269.143	0.604	0.694	43.094	-0.257	6.291	138.997	-5.688
151.881	0.433	151.881	0.630	0.670	46.997	-0.921	6.024	40.218	-6.147
120.876	-0.433	120.875	0.630	0.670	46.997	0.921	6.024	31.709	4.846
110.365	-0.701	110.364	0.669	0.631	52.948	1.323	5.676	136.991	31.932
203.006	-0.103	203.006	0.708	0.587	59.218	0.174	5.367	48.556	1.571
287.553	0.803	287.554	0.658	0.642	51.228	-1.568	5.770	178.462	-48.499
111.290	0.907	111.291	0.533	0.749	33.632	-2.698	7.122	288.205	-109.199
7.887	-0.349	7.887	0.679	0.620	54.558	0.639	5.591	71.890	8.220
351.681	-0.559	351.680	0.656	0.645	50.862	1.098	5.791	203.488	38.598

Table 8. Observation Table for Distance and Direction

2.6 Unknowns and Standard Deviation

ID	Easting [m]	Std. Dev of Easting [mm]	Northing [m]	Std. Dev of Northing [mm]
1000	4590337.389	6.5248	5820823.642	9.1915
2000	4589967.527	6.5248	5820806.067	9.1915
3000	4590078.02	8.0598	5820681.709	7.0838
100	4590159.875	7.2154	5820727.367	6.5765
101	4589800.103	6.7878	5820857.978	14.1874
102	4590163.254	6.6340	5820848.787	6.5894
103	4589956.949	7.7097	5820700.411	9.4966

Table 9. Estimated Co-ordinates

ID	Reference Angle [gon]	Std. Dev of Ref Angle [mgon]
1000	398.666	2.3492
2000	398.365	2.4443
3000	398.364	2.3642
100	398.465	2.3087
101	398.265	2.4744
102	398.765	2.3033
103	398.466	2.3586

Table 10. Estimated Reference Angles

TASK III

(Parametric Adjustment)

Objective: The main objective of this task is to perform parametric adjustment to calculate Gauss-Krueger coordinates of new points, while the coordinates of the control points (after removing the blunder) will remain fixed. Apart from that, the parameters of the internal and external reliability of the network have to be calculated. After calculation, global test has to be done to evaluate the mathematical and stochastic model. The results should be presented in tables so that the network can be determined in terms of reliability and accuracy.

3.1 *Parametric Adjustment*

3.1.1 *Functional Model*

The functional model for distance measurement is given by:

$$s_{ij} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \qquad \ldots(65)$$

The functional model for direction is given by:

$$r_{ij} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \qquad \ldots(66)$$

where, ω_i is the zero directions for each point.

3.1.2 *Observation Equation*

The observation equations can be written as:

$$s_{ij} + v_{s_{ij}} = \sqrt{(y_j - y_i)^2 + (x_j - x_i)^2} \qquad \ldots(67)$$

and for directions the equation can be written as:

$$r_{ij} + v_{r_{ij}} = tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \omega_i \qquad \ldots(68)$$

where, $v_{s_{ij}}$ and $v_{r_{ij}}$ are the residuals.

3.1.3 *Stochastic Model*

It is mentioned that the standard deviation for the distance measurement is 1mm and for direction is 1mgon. So,

Distance Measurement [σ_s] = 0.001 m

Direction Measurement [σ_r] = $0.001 \times \frac{\pi}{200} = 1.570796 \times 10^{-5}$ rad

Again, it is mentioned that $\sigma_0 = 1$. So, the Stochastic model is a 36×36 matrix which can be written as:

$$\sum_{LL} = \begin{bmatrix} \sigma^2_{s_{1000,100}} & 0 & \cdots & 0 & 0 \\ 0 & \sigma^2_{s_{1000,102}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{r_{103,2000}} & 0 \\ 0 & 0 & \cdots & 0 & \sigma^2_{s_{103,101}} \end{bmatrix} \qquad \ldots(69)$$

$$Q_{ll} = \frac{1}{\sigma_0^2} \sum_{LL} \quad \text{... (70)}$$

$$P = Q_{ll}^{-1} \quad \text{... (71)}$$

3.1.4 *Unknown Vector*

$$X^0 = \begin{bmatrix} y_{1000} \\ x_{1000} \\ y_{2000} \\ x_{2000} \\ y_{3000} \\ x_{3000} \\ y_{100} \\ x_{100} \\ y_{101} \\ x_{101} \\ y_{102} \\ x_{102} \\ y_{103} \\ x_{103} \\ \omega_{1000} \\ \omega_{2000} \\ \omega_{3000} \\ \omega_{100} \\ \omega_{101} \\ \omega_{102} \\ \omega_{103} \end{bmatrix} \quad \text{... (72)}$$

3.1.5 *Design Matrix*

The design matrix consists of partial derivatives of the equation with respect to the unknowns.

$$A = \begin{bmatrix} \frac{\partial_{s_{1000,100}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{s_{1000,100}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{s_{103,101}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{s_{103,101}}}{\partial_{\omega_{103}}} \\ \frac{\partial_{r_{1000,100}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{r_{1000,100}}}{\partial_{\omega_{103}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial_{r_{103,101}}}{\partial_{y_{1000}}} & \cdots & \frac{\partial_{r_{103,101}}}{\partial_{\omega_{103}}} \end{bmatrix} \quad \text{... (73)}$$

3.1.6 *Adjusted Unknowns*

To compute the adjusted unknowns, the following procedure has been used.

The normal equation system is given by:

$$N = A^T P A \quad \text{... (74)}$$

and the vector of absolute values:

$$n = A^T P l \quad \text{... (75)}$$

where, l is the observation vector.

The corrections $\hat{x}$ of the unknowns are calculated equation:

$$\hat{x} = Q_{ext} n_{ext} \quad ...(76)$$

where,

$$Q_{ext} = N_{ext}^{-1} \quad ...(77)$$

The unknowns are given by the equation:

$$X_{final} = X_0 + \hat{x} \quad ...(78)$$

where, X_0 are the adjusted unknowns.

3.2 *Calculation of parameters for Internal and External Reliability (EV, NV, GF, GRZW, EGK and EP)*

3.2.1 Internal Reliability

To compute the internal reliability, the assumptions should be taken as

$$r_i = diag(Q_{vv}P) \quad ...(79)$$

where, Q_{vv} is the co-factor matrix of the residuals.

According to this, the redundancy is calculated by

$$EV = 100 r_i \quad ...(80)$$

The following rating scale can be taken as a basis for the evaluation of the redundancy number:

$0\% \leq EV < 1\%$: Observation not controlled,

$1\% \leq EV < 10\%$: Observation poorly controlled,

$10\% \leq EV < 30\%$: Observation sufficiently controlled,

$30\% \leq EV < 70\%$: Observation sufficiently controlled,

$70\% \leq EV < 100\%$: Observation can be dismissed without loss of reliability

The standardized residuals can be calculated by the equation

$$NV = \frac{|v|}{\sigma_v} \quad ...(81)$$

where, $\sigma_v = \sigma_0 \sqrt{q_{v_i v_i}}$

The following scale can be taken as a basis for the evaluation of the standardized residuals:

$NV < 2.5$: No blunder distinguishable,

$2.5 \leq NV < 4.0$: Blunder possible,

$4.0 \leq NV$: Blunder is highly probable

The potential magnitude of gross error is calculated as shown below:

$$GF_i = \frac{-v_i}{r_i} \quad ...(82)$$

where

Since, $\delta_0 = 4.13$, the boundary value for the blunders is

$$GRZW_i = \frac{\sigma_0 \delta_0}{\sqrt{r_i P_i}} \quad \text{... (83)}$$

3.2.2 External Reliability

It describes the influence of observation errors on the parameters which have to be determined. In this case, the parameters are coordinates and zero directions.

The impact of the boundary value for the blunders on the coordinates of the corresponding points is described by the value

$$EGK_i = (1 - r_i) GRZW_i \quad \text{... (84)}$$

The impact of a potential magnitude of the blunder GF_i on a point corresponding to the measurement is described by the value:

$$EP_i = (1 - r_i) GF_i \quad \text{... (85)}$$

3.3 *Functional and Stochastic Model Evaluation using Global Test*

The functional and stochastic model have to be evaluated, according to an appropriate statistical test. The evaluation will be performed according to χ^2 test. For approaching, a null and alternative hypothesis are constructed.

Null Hypothesis:

$$H_0 : E\{s_0^2\} = \sigma_0^2 \quad \text{... (86)}$$

Alternative Hypothesis:

$$H_A : E\{s_0^2\} < \sigma_0^2 \quad \text{... (87)}$$

In the case that the null hypothesis is accepted, it means that the stochastic model is considered to be chosen correctly a priori, since σ_0^2 is almost equal to the posteriori value s_0^2. However, if null hypothesis cannot be accepted which means that the observation contains blunders or the functional model is maybe incomplete/inappropriate or the stochastic model is inappropriate.

The test is performed according to the equation

$$T_{\chi^2} = \frac{f s_0^2}{\sigma_0^2} \quad \text{... (88)}$$

Now for further approach, the following considerations should be considered.

Confidence level $S = 95\%$ i.e. $\alpha = 0.05$

Degree of freedom f = 17

Threshold taken from the table $\chi^2_{f,\alpha} = 27.59$

Now, as it is defined that $T_{\chi^2} < \chi^2_{f,\alpha}$ then the null hypothesis can be accepted. Otherwise the null hypothesis fails to be accepted and alternative hypothesis should be considered.

For the performed adjustment the test statistic value is $T_{\chi^2} = 21.17$ which is less than $\chi^2_{f,\alpha}$. With the probability of 95% the null hypothesis is correct. So, it can be said that the functional model or stochastic models are appropriate and the observation contain no blunders.

NV after 1st Iteration
0.496756408
2.518104969
1.154979956
0.405261021
1.143380789
0.496756408
1.661721446
0.082900054
0.990832787
1.851627616
0.163979543
0.05830943
1.007092491
2.671304042
0.22746174
0.613902924
0.776480283
0.697934518
0.697934518
0.772414493
0.772414493
1.51538657
1.51538657
0.652609135
0.649845177
0.071974061
0.629526069
0.629526069
0.870875913
0.044715588
0.931591573
1.877272328
0.61444044
0.900271219

Table 11. Observation Table for Standardized Residuals

T_{χ^2} after 1st Iteration
21.16969353

Table 12. Observation Table for T_{χ^2}

3.4 *Observation Table*

The following tables present data for the performed adjustments.

Obs (m)	Residuals (mm)	Adjusted Obs (m)	Std. Dev of Obs (mm)	Std. Dev of Adjusted Obs (mm)	EV %	GF (mm)	GRZW (mm)	EGK (mm)	EP (mm)
201.941	0.385	201.941	0.866	0.704	60.156	-0.640	5.325	2.122	-0.255
175.940	2.047	175.942	0.907	0.650	66.109	-3.097	5.079	1.721	-1.050
175.288	-0.876	175.287	0.846	0.728	57.483	1.523	5.447	2.316	0.648
93.728	-0.290	93.728	0.799	0.779	51.306	0.566	5.766	2.808	0.276
122.506	0.834	122.507	0.814	0.763	53.214	-1.567	5.662	2.649	-0.733
201.941	0.385	201.941	0.866	0.704	60.156	-0.640	5.325	2.122	-0.255
121.468	-1.245	121.467	0.836	0.739	56.160	2.217	5.511	2.416	0.972
207.826	-0.055	207.826	0.741	0.835	44.063	0.125	6.222	3.480	0.070
93.727	0.710	93.728	0.799	0.779	51.306	-1.383	5.766	2.808	-0.674
222.323	1.409	222.324	0.849	0.724	57.913	-2.433	5.427	2.284	-1.024
175.287	0.124	175.287	0.846	0.728	57.483	-0.216	5.447	2.316	-0.092
175.942	0.047	175.942	0.907	0.650	66.109	-0.072	5.079	1.721	-0.024
121.466	0.755	121.467	0.836	0.739	56.160	-1.344	5.511	2.416	-0.589
200.334	2.116	200.336	0.884	0.681	62.749	-3.372	5.214	1.942	-1.256
122.507	-0.166	122.507	0.814	0.763	53.214	0.312	5.662	2.649	0.146
106.185	-0.262	106.185	0.477	1.009	18.265	1.436	9.663	7.898	1.174
222.325	-0.591	222.324	0.849	0.724	57.913	1.020	5.427	2.284	0.429
(gon)	**(mgon)**	**(gon)**	**(mgon)**	**(mgon)**	**%**	**(mgon)**	**(mgon)**	**(mgon)**	**(mgon)**
269.698	-0.486	269.698	0.776	0.802	48.400	1.003	5.936	19.178	3.241
310.463	0.486	310.464	0.776	0.802	48.400	-1.003	5.936	16.709	-2.824
207.987	0.476	207.987	0.687	0.879	37.921	-1.254	6.707	99.240	-18.560
320.775	-0.476	320.775	0.687	0.879	37.921	1.254	6.707	141.996	26.557
69.244	0.761	69.244	0.560	0.965	25.191	-3.019	8.229	191.338	-70.206
311.393	-0.761	311.392	0.560	0.965	25.191	3.019	8.229	250.086	91.762
69.898	0.451	69.898	0.771	0.807	47.696	-0.945	5.980	229.093	-36.200
326.259	-0.498	326.259	0.855	0.717	58.724	0.848	5.389	88.957	13.997
269.143	0.047	269.143	0.733	0.841	43.154	-0.110	6.287	138.549	-2.415

151.881	0.432	151.881	0.765	0.812	46.997	-0.918	6.024	40.217	-6.130
120.876	-0.432	120.875	0.765	0.812	46.997	0.918	6.024	31.709	4.833
110.365	-0.634	110.364	0.813	0.765	53.013	1.196	5.672	136.258	28.732
203.006	-0.034	203.006	0.859	0.712	59.287	0.058	5.364	48.080	0.521
287.553	0.669	287.554	0.801	0.777	51.496	-1.298	5.755	174.911	-39.454
111.290	1.097	111.291	0.652	0.905	34.162	-3.212	7.066	281.372	-127.896
7.887	-0.455	7.886	0.825	0.751	54.723	0.831	5.583	70.807	10.534
351.681	-0.643	351.680	0.797	0.781	50.966	1.261	5.785	201.944	44.020

Table 13. Observation Table for Distance and Direction

3.5 *Unknown and Standard Deviation Table*

ID	Easting [m]	Std. Dev of Easting [mm]	Northing [m]	Std. Dev of Northing [mm]
3000	4590078.02	1.0797	5820681.709	1.2107
100	4590159.875	0.6914	5820727.368	1.0856
101	4589800.101	1.2614	5820857.976	2.7960
102	4590163.254	0.6392	5820848.788	1.1838
103	4589956.949	1.2076	5820700.41	1.0374

Table 14. Estimated Coordinates

ID	Reference Angle [gon]	Std. Dev of Ref Angle [mgon]
1000	398.665961	0.8667
2000	398.3647235	1.1534
3000	398.3638115	0.9228
100	398.4653248	0.6963
101	398.2642794	1.2457
102	398.7653807	0.6695
103	398.4653537	0.8654

Table 15. Estimated Reference Angles

3.6 *Evaluation of Network Accuracy and Reliability*

From Table 1. it is clear that the values of NV lie between $0.0216 < NV < 1.8522$. So, it can be said that there is no blunder in the network. More precisely the network is reliable and accurate. Again from Table 3. it is clear that the values of EV lies between $17.698 < EV < 59.218$. So, it can be said that the observations are well controlled. So, the network is quite accurate and reliable.

References

[1] Selected Sections of Adjustment Calculation Lecture Notes, Prof. Dr.-Ing. Frank Neitzel, 2017

[2] Datum Defect for different type of geodetic networks, Handout, 2017,

[3] Internal and External Reliability, Handout, 2017,

[4] Statistical Test, Handout, 2017,

[5] Chapter 16: "Adjustment of Horizontal Survey: Traverses and Network", Adjustment Computations Spatial Data Analysis, Charles D. Ghinali, 2006

Codes

Task 1 Part 1

```
%-----------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%-----------------------------------------------------------------------------

clc;
clear all;
close all;
format long g;

%Load all files
distances = load('Distances.txt');
directions = load('Directions.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');

L = [distances(:,3); directions(:,3)*pi/200];
%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);
% New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
x103 = new_point(4,3);
%Initial values for orientation unknowns
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;

%Initial values for unknowns
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%-----------------------------------------------------------------------------
%   Points for datum definition
%-----------------------------------------------------------------------------
xy = reshape(X_0(1:14),2,7);
```

```
%Case 1, all control points for datum definition
datum = diag([1 1 1 0 0 0 0]);

%Number of points
p = sum(sum(datum));
%Centroid
x_c = (1/p)*sum(datum*xy(2,:)');
y_c = (1/p)*sum(datum*xy(1,:)');

%Coordinates reduced to the centroid
y1000 = y1000-y_c;
y2000 = y2000-y_c;
y3000 = y3000-y_c;
y100 = y100-y_c;
y101 = y101-y_c;
y102 = y102-y_c;
y103 = y103-y_c;
x1000 = x1000-x_c;
x2000 = x2000-x_c;
x3000 = x3000-x_c;
x100 = x100-x_c;
x101 = x101-x_c;
x102 = x102-x_c;
x103 = x103-x_c;

%Initial values for unknowns after reduction to the centroid
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%---------------------------------------------------------------------------

%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u + 3; %3 constraint equations

%---------------------------------------------------------------------------
%  Stochastic model
%---------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                          %[m]
s_dir = 0.001*pi/200;                  %Convert to [rad]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1)];
S_LL = diag(s_LL);

%Theoretical standard deviation
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
```

```
P = inv(Q_LL);

%-------------------------------------------------------------------------
%  Adjustment
%-------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-9; %given accuracy in 0.001
delta = 10^-11;
max_x_hat = 10^Inf;

%Number of iterations
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y103,x103);
    L_0(4)=dis(y2000,x2000,y101,x101);
    L_0(5)=dis(y3000,x3000,y100,x100);
    L_0(6)=dis(y3000,x3000,y103,x103);
    L_0(7)=dis(y100,x100,y1000,x1000);
    L_0(8)=dis(y100,x100,y102,x102);
    L_0(9)=dis(y100,x100,y2000,x2000);
    L_0(10)=dis(y100,x100,y3000,x3000);
    L_0(11)=dis(y101,x101,y103,x103);
    L_0(12)=dis(y101,x101,y2000,x2000);
    L_0(13)=dis(y102,x102,y1000,x1000);
    L_0(14)=dis(y102,x102,y100,x100);
    L_0(15)=dis(y102,x102,y2000,x2000);
    L_0(16)=dis(y103,x103,y3000,x3000);
    L_0(17)=dis(y103,x103,y2000,x2000);
    L_0(18)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    L_0(19)=direction(y1000,x1000,y100,x100,w1000);
    L_0(20)=direction(y1000,x1000,y102,x102,w1000);
    L_0(21)=direction(y2000,x2000,y103,x103,w2000);
    L_0(22)=direction(y2000,x2000,y101,x101,w2000);
    L_0(23)=direction(y3000,x3000,y100,x100,w3000);
    L_0(24)=direction(y3000,x3000,y103,x103,w3000);
    L_0(25)=direction(y100,x100,y1000,x1000,w100);
    L_0(26)=direction(y100,x100,y102,x102,w100);
    L_0(27)=direction(y100,x100,y2000,x2000,w100);
    L_0(28)=direction(y100,x100,y3000,x3000,w100);
    L_0(29)=direction(y101,x101,y103,x103,w101);
    L_0(30)=direction(y101,x101,y2000,x2000,w101);
    L_0(31)=direction(y102,x102,y1000,x1000,w102);
    L_0(32)=direction(y102,x102,y100,x100,w102);
    L_0(33)=direction(y102,x102,y2000,x2000,w102);
    L_0(34)=direction(y103,x103,y3000,x3000,w103);
    L_0(35)=direction(y103,x103,y2000,x2000,w103);
    L_0(36)=direction(y103,x103,y101,x101,w103);

    l = L-L_0';
```

```
%Design matrix
A(1,1)=ds_dy_from(y1000,x1000,y100,x100);
A(1,2)=ds_dx_from(y1000,x1000,y100,x100);
A(1,7)=ds_dy_to(y1000,x1000,y100,x100);
A(1,8)=ds_dx_to(y1000,x1000,y100,x100);

A(2,1)=ds_dy_from(y1000,x1000,y102,x102);
A(2,2)=ds_dx_from(y1000,x1000,y102,x102);
A(2,11)=ds_dy_to(y1000,x1000,y102,x102);
A(2,12)=ds_dx_to(y1000,x1000,y102,x102);

A(3,3)=ds_dy_from(y2000,x2000,y103,x103);
A(3,4)=ds_dx_from(y2000,x2000,y103,x103);
A(3,13)=ds_dy_to(y2000,x2000,y103,x103);
A(3,14)=ds_dx_to(y2000,x2000,y103,x103);

A(4,3)=ds_dy_from(y2000,x2000,y101,x101);
A(4,4)=ds_dx_from(y2000,x2000,y101,x101);
A(4,9)=ds_dy_to(y2000,x2000,y101,x101);
A(4,10)=ds_dx_to(y2000,x2000,y101,x101);

A(5,5)=ds_dy_from(y3000,x3000,y100,x100);
A(5,6)=ds_dx_from(y3000,x3000,y100,x100);
A(5,7)=ds_dy_to(y3000,x3000,y100,x100);
A(5,8)=ds_dx_to(y3000,x3000,y100,x100);

A(6,5)=ds_dy_from(y3000,x3000,y103,x103);
A(6,6)=ds_dx_from(y3000,x3000,y103,x103);
A(6,13)=ds_dy_to(y3000,x3000,y103,x103);
A(6,14)=ds_dx_to(y3000,x3000,y103,x103);

A(7,7)=ds_dy_from(y100,x100,y1000,x1000);
A(7,8)=ds_dx_from(y100,x100,y1000,x1000);
A(7,1)=ds_dy_to(y100,x100,y1000,x1000);
A(7,2)=ds_dx_to(y100,x100,y1000,x1000);

A(8,7)=ds_dy_from(y100,x100,y102,x102);
A(8,8)=ds_dx_from(y100,x100,y102,x102);
A(8,11)=ds_dy_to(y100,x100,y102,x102);
A(8,12)=ds_dx_to(y100,x100,y102,x102);

A(9,7)=ds_dy_from(y100,x100,y2000,x2000);
A(9,8)=ds_dx_from(y100,x100,y2000,x2000);
A(9,3)=ds_dy_to(y100,x100,y2000,x2000);
A(9,4)=ds_dx_to(y100,x100,y2000,x2000);

A(10,7)=ds_dy_from(y100,x100,y3000,x3000);
A(10,8)=ds_dx_from(y100,x100,y3000,x3000);
A(10,5)=ds_dy_to(y100,x100,y3000,x3000);
A(10,6)=ds_dx_to(y100,x100,y3000,x3000);

A(11,9)=ds_dy_from(y101,x101,y103,x103);
A(11,10)=ds_dx_from(y101,x101,y103,x103);
A(11,13)=ds_dy_to(y101,x101,y103,x103);
A(11,14)=ds_dx_to(y101,x101,y103,x103);

A(12,9)=ds_dy_from(y101,x101,y2000,x2000);
A(12,10)=ds_dx_from(y101,x101,y2000,x2000);
```

```
A(12,3)=ds_dy_to(y101,x101,y2000,x2000);
A(12,4)=ds_dx_to(y101,x101,y2000,x2000);

A(13,11)=ds_dy_from(y102,x102,y1000,x1000);
A(13,12)=ds_dx_from(y102,x102,y1000,x1000);
A(13,1)=ds_dy_to(y102,x102,y1000,x1000);
A(13,2)=ds_dx_to(y102,x102,y1000,x1000);

A(14,11)=ds_dy_from(y102,x102,y100,x100);
A(14,12)=ds_dx_from(y102,x102,y100,x100);
A(14,7)=ds_dy_to(y102,x102,y100,x100);
A(14,8)=ds_dx_to(y102,x102,y100,x100);

A(15,11)=ds_dy_from(y102,x102,y2000,x2000);
A(15,12)=ds_dx_from(y102,x102,y2000,x2000);
A(15,3)=ds_dy_to(y102,x102,y2000,x2000);
A(15,4)=ds_dx_to(y102,x102,y2000,x2000);

A(16,13)=ds_dy_from(y103,x103,y3000,x3000);
A(16,14)=ds_dx_from(y103,x103,y3000,x3000);
A(16,5)=ds_dy_to(y103,x103,y3000,x3000);
A(16,6)=ds_dx_to(y103,x103,y3000,x3000);

A(17,13)=ds_dy_from(y103,x103,y2000,x2000);
A(17,14)=ds_dx_from(y103,x103,y2000,x2000);
A(17,3)=ds_dy_to(y103,x103,y2000,x2000);
A(17,4)=ds_dx_to(y103,x103,y2000,x2000);

A(18,13)=ds_dy_from(y103,x103,y101,x101);
A(18,14)=ds_dx_from(y103,x103,y101,x101);
A(18,9)=ds_dy_to(y103,x103,y101,x101);
A(18,10)=ds_dx_to(y103,x103,y101,x101);


A(19,1)=dr_dy_from(y1000,x1000,y100,x100);
A(19,2)=dr_dx_from(y1000,x1000,y100,x100);
A(19,7)=dr_dy_to(y1000,x1000,y100,x100);
A(19,8)=dr_dx_to(y1000,x1000,y100,x100);
A(19,15)=-1;

A(20,1)=dr_dy_from(y1000,x1000,y102,x102);
A(20,2)=dr_dx_from(y1000,x1000,y102,x102);
A(20,11)=dr_dy_to(y1000,x1000,y102,x102);
A(20,12)=dr_dx_to(y1000,x1000,y102,x102);
A(20,15)=-1;

A(21,3)=dr_dy_from(y2000,x2000,y103,x103);
A(21,4)=dr_dx_from(y2000,x2000,y103,x103);
A(21,13)=dr_dy_to(y2000,x2000,y103,x103);
A(21,14)=dr_dx_to(y2000,x2000,y103,x103);
A(21,16)=-1;

A(22,3)=dr_dy_from(y2000,x2000,y101,x101);
A(22,4)=dr_dx_from(y2000,x2000,y101,x101);
A(22,9)=dr_dy_to(y2000,x2000,y101,x101);
A(22,10)=dr_dx_to(y2000,x2000,y101,x101);
A(22,16)=-1;

A(23,5)=dr_dy_from(y3000,x3000,y100,x100);
```

```
A(23,6)=dr_dx_from(y3000,x3000,y100,x100);
A(23,7)=dr_dy_to(y3000,x3000,y100,x100);
A(23,8)=dr_dx_to(y3000,x3000,y100,x100);
A(23,17)=-1;

A(24,5)=dr_dy_from(y3000,x3000,y103,x103);
A(24,6)=dr_dx_from(y3000,x3000,y103,x103);
A(24,13)=dr_dy_to(y3000,x3000,y103,x103);
A(24,14)=dr_dx_to(y3000,x3000,y103,x103);
A(24,17)=-1;

A(25,7)=dr_dy_from(y100,x100,y1000,x1000);
A(25,8)=dr_dx_from(y100,x100,y1000,x1000);
A(25,1)=dr_dy_to(y100,x100,y1000,x1000);
A(25,2)=dr_dx_to(y100,x100,y1000,x1000);
A(25,18)=-1;

A(26,7)=dr_dy_from(y100,x100,y102,x102);
A(26,8)=dr_dx_from(y100,x100,y102,x102);
A(26,11)=dr_dy_to(y100,x100,y102,x102);
A(26,12)=dr_dx_to(y100,x100,y102,x102);
A(26,18)=-1;

A(27,7)=dr_dy_from(y100,x100,y2000,x2000);
A(27,8)=dr_dx_from(y100,x100,y2000,x2000);
A(27,3)=dr_dy_to(y100,x100,y2000,x2000);
A(27,4)=dr_dx_to(y100,x100,y2000,x2000);
A(27,18)=-1;

A(28,7)=dr_dy_from(y100,x100,y3000,x3000);
A(28,8)=dr_dx_from(y100,x100,y3000,x3000);
A(28,5)=dr_dy_to(y100,x100,y3000,x3000);
A(28,6)=dr_dx_to(y100,x100,y3000,x3000);
A(28,18)=-1;

A(29,9)=dr_dy_from(y101,x101,y103,x103);
A(29,10)=dr_dx_from(y101,x101,y103,x103);
A(29,13)=dr_dy_to(y101,x101,y103,x103);
A(29,14)=dr_dx_to(y101,x101,y103,x103);
A(29,19)=-1;

A(30,9)=dr_dy_from(y101,x101,y2000,x2000);
A(30,10)=dr_dx_from(y101,x101,y2000,x2000);
A(30,3)=dr_dy_to(y101,x101,y2000,x2000);
A(30,4)=dr_dx_to(y101,x101,y2000,x2000);
A(30,19)=-1;

A(31,11)=dr_dy_from(y102,x102,y1000,x1000);
A(31,12)=dr_dx_from(y102,x102,y1000,x1000);
A(31,1)=dr_dy_to(y102,x102,y1000,x1000);
A(31,2)=dr_dx_to(y102,x102,y1000,x1000);
A(31,20)=-1;

A(32,11)=dr_dy_from(y102,x102,y100,x100);
A(32,12)=dr_dx_from(y102,x102,y100,x100);
A(32,7)=dr_dy_to(y102,x102,y100,x100);
A(32,8)=dr_dx_to(y102,x102,y100,x100);
A(32,20)=-1;
```

```
A(33,11)=dr_dy_from(y102,x102,y2000,x2000);
A(33,12)=dr_dx_from(y102,x102,y2000,x2000);
A(33,3)=dr_dy_to(y102,x102,y2000,x2000);
A(33,4)=dr_dx_to(y102,x102,y2000,x2000);
A(33,20)=-1;

A(34,13)=dr_dy_from(y103,x103,y3000,x3000);
A(34,14)=dr_dx_from(y103,x103,y3000,x3000);
A(34,5)=dr_dy_to(y103,x103,y3000,x3000);
A(34,6)=dr_dx_to(y103,x103,y3000,x3000);
A(34,21)=-1;

A(35,13)=dr_dy_from(y103,x103,y2000,x2000);
A(35,14)=dr_dx_from(y103,x103,y2000,x2000);
A(35,3)=dr_dy_to(y103,x103,y2000,x2000);
A(35,4)=dr_dx_to(y103,x103,y2000,x2000);
A(35,21)=-1;

A(36,13)=dr_dy_from(y103,x103,y101,x101);
A(36,14)=dr_dx_from(y103,x103,y101,x101);
A(36,9)=dr_dy_to(y103,x103,y101,x101);
A(36,10)=dr_dx_to(y103,x103,y101,x101);
A(36,21)=-1;


%Normal matrix
N = A'*P*A;


%Zero matrix
Zz = zeros(3,3);     %c constraints

%B matrix
B = [1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     -x1000 y1000 -x2000 y2000 -x3000 y3000 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 ];

%Extended normal matrix
%For case a, b, d
 N_ext = [N B'; B Zz];

%For case c
 %N_ext = [N G'; G Zz];

%Vector of right hand side of normal equations
 n = A'*P*l;

%Extended vector of right hand side of normal equations
 n_ext = [n;0;0;0];

%Inversion of normal matrix / Cofactor matrix of the unknowns
 Q_xx_ext = inv(N_ext);

%Reduced cofactor matrix of the unknowns
 Q_xx = Q_xx_ext(1:no_u,1:no_u);     %Q11

%Solution of normal equation
```

```
    x_hat = Q_xx_ext*n_ext;

    %Adjusted unknowns
     X_hat = X_0+x_hat(1:21);

    %Update
    X_0 = X_hat;

    y1000 = X_0(1);
    x1000 = X_0(2);
    y2000 = X_0(3);
    x2000 = X_0(4);
    y3000 = X_0(5);
    x3000 = X_0(6);
    y100 = X_0(7);
    x100 = X_0(8);
    y101 = X_0(9);
    x101 = X_0(10);
    y102 = X_0(11);
    x102 = X_0(12);
    y103 = X_0(13);
    x103 = X_0(14);
    w1000 = X_0(15);
    w2000 = X_0(16);
    w3000 = X_0(17);
    w100 = X_0(18);
    w101 = X_0(19);
    w102 = X_0(20);
    w103 = X_0(21);

    %Check 1
     max_x_hat = max(abs(x_hat));

    %Update number of iterations
    iteration=iteration+1;

end


%Vector of residuals
v = A*x_hat(1:21)-l;
v_gon = v(19:36,1)*200/pi;              %Convert to [gon]

%Objective function
vTPv = v'*P*v;

%Vector of adjusted observations
L_hat = L+v;
L_hat_gon = L_hat(19:36)*200/pi;      %Convert to [gon]


%Final check for the linearization

    %Vector of reduced distances
    Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
    Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
    Phi_X_hat(3)=dis(y2000,x2000,y103,x103);
    Phi_X_hat(4)=dis(y2000,x2000,y101,x101);
    Phi_X_hat(5)=dis(y3000,x3000,y100,x100);
```

```
    Phi_X_hat(6)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(7)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(8)=dis(y100,x100,y102,x102);
    Phi_X_hat(9)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(10)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(11)=dis(y101,x101,y103,x103);
    Phi_X_hat(12)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(13)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(14)=dis(y102,x102,y100,x100);
    Phi_X_hat(15)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(16)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(17)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(18)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(19)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(20)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(21)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(22)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(23)=direction(y3000,x3000,y100,x100,w3000);
    Phi_X_hat(24)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(25)=direction(y100,x100,y1000,x1000,w100);
    Phi_X_hat(26)=direction(y100,x100,y102,x102,w100);
    Phi_X_hat(27)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(28)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(29)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(30)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(31)=direction(y102,x102,y1000,x1000,w102);
    Phi_X_hat(32)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(33)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(34)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(35)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(36)=direction(y103,x103,y101,x101,w103);

% Final Check
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end

%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_gon = s_X(15:21,1)*200/pi;        %Convert to [gon]

%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';

%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;
```

```
%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_gon = s_L_hat(19:36)*200/pi;     %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_gon = s_v(19:36,1)*200/pi;        %Convert to [gon]

%De-reduced unknowns
    y1000 = X_0(1)+y_c;
    x1000 = X_0(2)+x_c;
    y2000 = X_0(3)+y_c;
    x2000 = X_0(4)+x_c;
    y3000 = X_0(5)+y_c;
    x3000 = X_0(6)+x_c;
    y100 = X_0(7)+y_c;
    x100 = X_0(8)+x_c;
    y101 = X_0(9)+y_c;
    x101 = X_0(10)+x_c;
    y102 = X_0(11)+y_c;
    x102 = X_0(12)+x_c;
    y103 = X_0(13)+y_c;
    x103 = X_0(14)+x_c;

X_final =  [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102
x102 y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%Convert to [gon] and check the quadrants
gon = X_final(15:21,1)*200/pi;    %always convert the rad to gon
gon = gon+400;
%gon(1) = gon(1)+400;
%gon(2) = gon(2)+400;
%gon(4) = gon(4)+400;


% Write to file: X_hat [m] for case a
% fid = fopen('X_hat_case_a', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);

% Write to file: X_hat [m] for case b
% fid = fopen('X_hat_case_b', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);

% Write to file: X_hat [m] for case c, d
% fid = fopen('X_hat_case_d', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);


%--------------------------------------------------------------------------
%  Global test
```

```
%---------------------------------------------------------------------------

%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
tx2l = chi2inv(0.025,r);  %a/2

if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%---------------------------------------------------------------------------
% Internal and external reliability parameters
%---------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_gon = GF(19:36,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end

GRZW_d= GRZW(1:18,1);
GRZW_gon = GRZW(19:36,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(18,1);

r_w(1)  = P_diag(19,1)/(P_diag(19,1)+P_diag(20,1));
r_w(2)  = P_diag(20,1)/(P_diag(19,1)+P_diag(20,1));
r_w(3)  = P_diag(21,1)/(P_diag(21,1)+P_diag(22,1));
r_w(4)  = P_diag(22,1)/(P_diag(21,1)+P_diag(22,1));
r_w(5)  = P_diag(23,1)/(P_diag(23,1)+P_diag(24,1));
r_w(6)  = P_diag(24,1)/(P_diag(23,1)+P_diag(24,1));
r_w(7)  =
P_diag(25,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1)+P_diag(28,1));
r_w(8)  =
P_diag(26,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1)+P_diag(28,1));
r_w(9)  =
P_diag(27,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1)+P_diag(28,1));
```

```
r_w(10) =
P_diag(28,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1)+P_diag(28,1));
r_w(11) = P_diag(29,1)/(P_diag(29,1)+P_diag(30,1));
r_w(12) = P_diag(30,1)/(P_diag(29,1)+P_diag(30,1));
r_w(13) = P_diag(31,1)/(P_diag(31,1)+P_diag(32,1)+P_diag(33,1));
r_w(14) = P_diag(32,1)/(P_diag(31,1)+P_diag(32,1)+P_diag(33,1));
r_w(15) = P_diag(33,1)/(P_diag(31,1)+P_diag(32,1)+P_diag(33,1));
r_w(16) = P_diag(34,1)/(P_diag(34,1)+P_diag(35,1)+P_diag(36,1));
r_w(17) = P_diag(35,1)/(P_diag(34,1)+P_diag(35,1)+P_diag(36,1));
r_w(18) = P_diag(36,1)/(P_diag(34,1)+P_diag(35,1)+P_diag(36,1));

dd = distances(:,3);

%dd(1)  = dist(1,3);
%dd(2)  = dist(2,3);
%dd(3)  = dist(3,3);
%dd(4)  = dist(4,3);
%dd(5)  = dist(5,3);
%dd(6)  = dist(6,3);
%dd(7)  = dist(7,3);
%dd(8)  = dist(8,3);
%dd(9)  = dist(9,3);
%dd(10)  = dist(10,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(36,1);

for i=1:18
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end

for i=19:36
  EGK(i) = (1-EV(i,1)-r_w(i-18,1))*GRZW(i,1)*dd(i-18);
end

EGK_gon = EGK(19:36,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
EP = ones(36,1);

for i=1:18
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=19:36
  EP(i,1) = (1-EV(i,1)-r_w(i-18,1))*GF(i,1)*dd(i-18);
end

EP_gon = EP(19:36,1)*200/pi;


%Display the result
results.L=[L(1:18,1); directions(:,3)];
results.v=[v(1:18,1); v_gon(:,1)];
results.L=[L_hat(1:18,1); L_hat_gon(:,1)];
results.s=[s_v(1:18,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:18,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:18,1); GF_gon(:,1)];
```

```
results.GRZW=GRZW(1:18,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:18,1); EGK_gon(:,1)];
results.EP=[EP(1:18,1); EP_gon(:,1)];

results= struct2table(results);
writetable(results, 'task1_1.xls');
```

Task 1 Part 2

```
%--------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%--------------------------------------------------------------------------

clc;
clear all;
close all;
format long g;

distances = load('Distances.txt');
directions = load('Directions_1.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');

L = [distances(:,3); directions(:,3)*pi/200];

%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);
%New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
x103 = new_point(4,3);

%Initial values for orientation unknowns
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;

%Initial values for unknowns
```

```
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%--------------------------------------------------------------------------
%   Points for datum definition
%--------------------------------------------------------------------------
xy = reshape(X_0(1:14),2,7);

%Case 1, all control points for datum definition
datum = diag([1 1 1 0 0 0 0]);

%Number of points
p = sum(sum(datum));
%Centroid
x_c = (1/p)*sum(datum*xy(2,:)');
y_c = (1/p)*sum(datum*xy(1,:)');

%Coordinates reduced to the centroid
y1000 = y1000-y_c;
y2000 = y2000-y_c;
y3000 = y3000-y_c;
y100 = y100-y_c;
y101 = y101-y_c;
y102 = y102-y_c;
y103 = y103-y_c;
x1000 = x1000-x_c;
x2000 = x2000-x_c;
x3000 = x3000-x_c;
x100 = x100-x_c;
x101 = x101-x_c;
x102 = x102-x_c;
x103 = x103-x_c;

%Initial values for unknowns after reduction to the centroid
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%--------------------------------------------------------------------------

%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u + 3; %3 constraint equations

%--------------------------------------------------------------------------
%  Stochastic model
%--------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                          %[m]
s_dir = 0.001*pi/200;                  %Convert to [rad]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1)];
S_LL = diag(s_LL);
```

```
%Theoretical standard deviation
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
P = inv(Q_LL);

%--------------------------------------------------------------------------
%  Adjustment
%--------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-7; %given accuracy in 0.001
delta = 10^-11;
max_x_hat = 10^Inf;

%Number of iterations
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y103,x103);
    L_0(4)=dis(y2000,x2000,y101,x101);
    L_0(5)=dis(y3000,x3000,y100,x100);
    L_0(6)=dis(y3000,x3000,y103,x103);
    L_0(7)=dis(y100,x100,y1000,x1000);
    L_0(8)=dis(y100,x100,y102,x102);
    L_0(9)=dis(y100,x100,y2000,x2000);
    L_0(10)=dis(y100,x100,y3000,x3000);
    L_0(11)=dis(y101,x101,y103,x103);
    L_0(12)=dis(y101,x101,y2000,x2000);
    L_0(13)=dis(y102,x102,y1000,x1000);
    L_0(14)=dis(y102,x102,y100,x100);
    L_0(15)=dis(y102,x102,y2000,x2000);
    L_0(16)=dis(y103,x103,y3000,x3000);
    L_0(17)=dis(y103,x103,y2000,x2000);
    L_0(18)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    L_0(19)=direction(y1000,x1000,y100,x100,w1000);
    L_0(20)=direction(y1000,x1000,y102,x102,w1000);
    L_0(21)=direction(y2000,x2000,y103,x103,w2000);
    L_0(22)=direction(y2000,x2000,y101,x101,w2000);
    L_0(23)=direction(y3000,x3000,y100,x100,w3000);
    L_0(24)=direction(y3000,x3000,y103,x103,w3000);
    L_0(25)=direction(y100,x100,y1000,x1000,w100);
    L_0(26)=direction(y100,x100,y2000,x2000,w100);
    L_0(27)=direction(y100,x100,y3000,x3000,w100);
    L_0(28)=direction(y101,x101,y103,x103,w101);
    L_0(29)=direction(y101,x101,y2000,x2000,w101);
    L_0(30)=direction(y102,x102,y1000,x1000,w102);
    L_0(31)=direction(y102,x102,y100,x100,w102);
```

```
L_0(32)=direction(y102,x102,y2000,x2000,w102);
L_0(33)=direction(y103,x103,y3000,x3000,w103);
L_0(34)=direction(y103,x103,y2000,x2000,w103);
L_0(35)=direction(y103,x103,y101,x101,w103);

l = L-L_0';

%Design matrix
A(1,1)=ds_dy_from(y1000,x1000,y100,x100);
A(1,2)=ds_dx_from(y1000,x1000,y100,x100);
A(1,7)=ds_dy_to(y1000,x1000,y100,x100);
A(1,8)=ds_dx_to(y1000,x1000,y100,x100);

A(2,1)=ds_dy_from(y1000,x1000,y102,x102);
A(2,2)=ds_dx_from(y1000,x1000,y102,x102);
A(2,11)=ds_dy_to(y1000,x1000,y102,x102);
A(2,12)=ds_dx_to(y1000,x1000,y102,x102);

A(3,3)=ds_dy_from(y2000,x2000,y103,x103);
A(3,4)=ds_dx_from(y2000,x2000,y103,x103);
A(3,13)=ds_dy_to(y2000,x2000,y103,x103);
A(3,14)=ds_dx_to(y2000,x2000,y103,x103);

A(4,3)=ds_dy_from(y2000,x2000,y101,x101);
A(4,4)=ds_dx_from(y2000,x2000,y101,x101);
A(4,9)=ds_dy_to(y2000,x2000,y101,x101);
A(4,10)=ds_dx_to(y2000,x2000,y101,x101);

A(5,5)=ds_dy_from(y3000,x3000,y100,x100);
A(5,6)=ds_dx_from(y3000,x3000,y100,x100);
A(5,7)=ds_dy_to(y3000,x3000,y100,x100);
A(5,8)=ds_dx_to(y3000,x3000,y100,x100);

A(6,5)=ds_dy_from(y3000,x3000,y103,x103);
A(6,6)=ds_dx_from(y3000,x3000,y103,x103);
A(6,13)=ds_dy_to(y3000,x3000,y103,x103);
A(6,14)=ds_dx_to(y3000,x3000,y103,x103);

A(7,7)=ds_dy_from(y100,x100,y1000,x1000);
A(7,8)=ds_dx_from(y100,x100,y1000,x1000);
A(7,1)=ds_dy_to(y100,x100,y1000,x1000);
A(7,2)=ds_dx_to(y100,x100,y1000,x1000);

A(8,7)=ds_dy_from(y100,x100,y102,x102);
A(8,8)=ds_dx_from(y100,x100,y102,x102);
A(8,11)=ds_dy_to(y100,x100,y102,x102);
A(8,12)=ds_dx_to(y100,x100,y102,x102);

A(9,7)=ds_dy_from(y100,x100,y2000,x2000);
A(9,8)=ds_dx_from(y100,x100,y2000,x2000);
A(9,3)=ds_dy_to(y100,x100,y2000,x2000);
A(9,4)=ds_dx_to(y100,x100,y2000,x2000);

A(10,7)=ds_dy_from(y100,x100,y3000,x3000);
A(10,8)=ds_dx_from(y100,x100,y3000,x3000);
A(10,5)=ds_dy_to(y100,x100,y3000,x3000);
A(10,6)=ds_dx_to(y100,x100,y3000,x3000);

A(11,9)=ds_dy_from(y101,x101,y103,x103);
```

```
A(11,10)=ds_dx_from(y101,x101,y103,x103);
A(11,13)=ds_dy_to(y101,x101,y103,x103);
A(11,14)=ds_dx_to(y101,x101,y103,x103);

A(12,9)=ds_dy_from(y101,x101,y2000,x2000);
A(12,10)=ds_dx_from(y101,x101,y2000,x2000);
A(12,3)=ds_dy_to(y101,x101,y2000,x2000);
A(12,4)=ds_dx_to(y101,x101,y2000,x2000);

A(13,11)=ds_dy_from(y102,x102,y1000,x1000);
A(13,12)=ds_dx_from(y102,x102,y1000,x1000);
A(13,1)=ds_dy_to(y102,x102,y1000,x1000);
A(13,2)=ds_dx_to(y102,x102,y1000,x1000);

A(14,11)=ds_dy_from(y102,x102,y100,x100);
A(14,12)=ds_dx_from(y102,x102,y100,x100);
A(14,7)=ds_dy_to(y102,x102,y100,x100);
A(14,8)=ds_dx_to(y102,x102,y100,x100);

A(15,11)=ds_dy_from(y102,x102,y2000,x2000);
A(15,12)=ds_dx_from(y102,x102,y2000,x2000);
A(15,3)=ds_dy_to(y102,x102,y2000,x2000);
A(15,4)=ds_dx_to(y102,x102,y2000,x2000);

A(16,13)=ds_dy_from(y103,x103,y3000,x3000);
A(16,14)=ds_dx_from(y103,x103,y3000,x3000);
A(16,5)=ds_dy_to(y103,x103,y3000,x3000);
A(16,6)=ds_dx_to(y103,x103,y3000,x3000);

A(17,13)=ds_dy_from(y103,x103,y2000,x2000);
A(17,14)=ds_dx_from(y103,x103,y2000,x2000);
A(17,3)=ds_dy_to(y103,x103,y2000,x2000);
A(17,4)=ds_dx_to(y103,x103,y2000,x2000);

A(18,13)=ds_dy_from(y103,x103,y101,x101);
A(18,14)=ds_dx_from(y103,x103,y101,x101);
A(18,9)=ds_dy_to(y103,x103,y101,x101);
A(18,10)=ds_dx_to(y103,x103,y101,x101);

A(19,1)=dr_dy_from(y1000,x1000,y100,x100);
A(19,2)=dr_dx_from(y1000,x1000,y100,x100);
A(19,7)=dr_dy_to(y1000,x1000,y100,x100);
A(19,8)=dr_dx_to(y1000,x1000,y100,x100);
A(19,15)=-1;

A(20,1)=dr_dy_from(y1000,x1000,y102,x102);
A(20,2)=dr_dx_from(y1000,x1000,y102,x102);
A(20,11)=dr_dy_to(y1000,x1000,y102,x102);
A(20,12)=dr_dx_to(y1000,x1000,y102,x102);
A(20,15)=-1;

A(21,3)=dr_dy_from(y2000,x2000,y103,x103);
A(21,4)=dr_dx_from(y2000,x2000,y103,x103);
A(21,13)=dr_dy_to(y2000,x2000,y103,x103);
A(21,14)=dr_dx_to(y2000,x2000,y103,x103);
A(21,16)=-1;

A(22,3)=dr_dy_from(y2000,x2000,y101,x101);
A(22,4)=dr_dx_from(y2000,x2000,y101,x101);
```

```
A(22,9)=dr_dy_to(y2000,x2000,y101,x101);
A(22,10)=dr_dx_to(y2000,x2000,y101,x101);
A(22,16)=-1;

A(23,5)=dr_dy_from(y3000,x3000,y100,x100);
A(23,6)=dr_dx_from(y3000,x3000,y100,x100);
A(23,7)=dr_dy_to(y3000,x3000,y100,x100);
A(23,8)=dr_dx_to(y3000,x3000,y100,x100);
A(23,17)=-1;

A(24,5)=dr_dy_from(y3000,x3000,y103,x103);
A(24,6)=dr_dx_from(y3000,x3000,y103,x103);
A(24,13)=dr_dy_to(y3000,x3000,y103,x103);
A(24,14)=dr_dx_to(y3000,x3000,y103,x103);
A(24,17)=-1;

A(25,7)=dr_dy_from(y100,x100,y1000,x1000);
A(25,8)=dr_dx_from(y100,x100,y1000,x1000);
A(25,1)=dr_dy_to(y100,x100,y1000,x1000);
A(25,2)=dr_dx_to(y100,x100,y1000,x1000);
A(25,18)=-1;

A(26,7)=dr_dy_from(y100,x100,y2000,x2000);
A(26,8)=dr_dx_from(y100,x100,y2000,x2000);
A(26,3)=dr_dy_to(y100,x100,y2000,x2000);
A(26,4)=dr_dx_to(y100,x100,y2000,x2000);
A(26,18)=-1;

A(27,7)=dr_dy_from(y100,x100,y3000,x3000);
A(27,8)=dr_dx_from(y100,x100,y3000,x3000);
A(27,5)=dr_dy_to(y100,x100,y3000,x3000);
A(27,6)=dr_dx_to(y100,x100,y3000,x3000);
A(27,18)=-1;

A(28,9)=dr_dy_from(y101,x101,y103,x103);
A(28,10)=dr_dx_from(y101,x101,y103,x103);
A(28,13)=dr_dy_to(y101,x101,y103,x103);
A(28,14)=dr_dx_to(y101,x101,y103,x103);
A(28,19)=-1;

A(29,9)=dr_dy_from(y101,x101,y2000,x2000);
A(29,10)=dr_dx_from(y101,x101,y2000,x2000);
A(29,3)=dr_dy_to(y101,x101,y2000,x2000);
A(29,4)=dr_dx_to(y101,x101,y2000,x2000);
A(29,19)=-1;

A(30,11)=dr_dy_from(y102,x102,y1000,x1000);
A(30,12)=dr_dx_from(y102,x102,y1000,x1000);
A(30,1)=dr_dy_to(y102,x102,y1000,x1000);
A(30,2)=dr_dx_to(y102,x102,y1000,x1000);
A(30,20)=-1;

A(31,11)=dr_dy_from(y102,x102,y100,x100);
A(31,12)=dr_dx_from(y102,x102,y100,x100);
A(31,7)=dr_dy_to(y102,x102,y100,x100);
A(31,8)=dr_dx_to(y102,x102,y100,x100);
A(31,20)=-1;

A(32,11)=dr_dy_from(y102,x102,y2000,x2000);
```

```
A(32,12)=dr_dx_from(y102,x102,y2000,x2000);
A(32,3)=dr_dy_to(y102,x102,y2000,x2000);
A(32,4)=dr_dx_to(y102,x102,y2000,x2000);
A(32,20)=-1;

A(33,13)=dr_dy_from(y103,x103,y3000,x3000);
A(33,14)=dr_dx_from(y103,x103,y3000,x3000);
A(33,5)=dr_dy_to(y103,x103,y3000,x3000);
A(33,6)=dr_dx_to(y103,x103,y3000,x3000);
A(33,21)=-1;

A(34,13)=dr_dy_from(y103,x103,y2000,x2000);
A(34,14)=dr_dx_from(y103,x103,y2000,x2000);
A(34,3)=dr_dy_to(y103,x103,y2000,x2000);
A(34,4)=dr_dx_to(y103,x103,y2000,x2000);
A(34,21)=-1;

A(35,13)=dr_dy_from(y103,x103,y101,x101);
A(35,14)=dr_dx_from(y103,x103,y101,x101);
A(35,9)=dr_dy_to(y103,x103,y101,x101);
A(35,10)=dr_dx_to(y103,x103,y101,x101);
A(35,21)=-1;


%Normal matrix
N = A'*P*A;


%Zero matrix
Zz = zeros(3,3);     %c constraints

%B matrix
 %B matrix
B = [1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     -x1000 y1000 -x2000 y2000 -x3000 y3000 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 ];

%Extended normal matrix
%For case a, b, d
 N_ext = [N B'; B Zz];

%For case c
 %N_ext = [N G'; G Zz];

%Vector of right hand side of normal equations
 n = A'*P*l;

%Extended vector of right hand side of normal equations
 n_ext = [n;0;0;0];

%Inversion of normal matrix / Cofactor matrix of the unknowns
 Q_xx_ext = inv(N_ext);

%Reduced cofactor matrix of the unknowns
 Q_xx = Q_xx_ext(1:no_u,1:no_u);     %Q11

%Solution of normal equation
```

```
        x_hat = Q_xx_ext*n_ext;

       %Adjusted unknowns
        X_hat = X_0+x_hat(1:21);

       %Update
       X_0 = X_hat;

       y1000 = X_0(1);
       x1000 = X_0(2);
       y2000 = X_0(3);
       x2000 = X_0(4);
       y3000 = X_0(5);
       x3000 = X_0(6);
       y100 = X_0(7);
       x100 = X_0(8);
       y101 = X_0(9);
       x101 = X_0(10);
       y102 = X_0(11);
       x102 = X_0(12);
       y103 = X_0(13);
       x103 = X_0(14);
       w1000 = X_0(15);
       w2000 = X_0(16);
       w3000 = X_0(17);
       w100 = X_0(18);
       w101 = X_0(19);
       w102 = X_0(20);
       w103 = X_0(21);

       %Check 1
        max_x_hat = max(abs(x_hat));

       %Update number of iterations
       iteration=iteration+1;

   end


   %Vector of residuals
   v = A*x_hat(1:21)-l;
   v_gon = v(19:35,1)*200/pi;             %Convert to [gon]

   %Objective function
   vTPv = v'*P*v;

   %Vector of adjusted observations
   L_hat = L+v;
   L_hat_gon = L_hat(19:35)*200/pi;      %Convert to [gon]


   %Final check for the linearization

       %Vector of reduced distances
       Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
       Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
       Phi_X_hat(3)=dis(y2000,x2000,y103,x103);
       Phi_X_hat(4)=dis(y2000,x2000,y101,x101);
       Phi_X_hat(5)=dis(y3000,x3000,y100,x100);
```

```
    Phi_X_hat(6)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(7)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(8)=dis(y100,x100,y102,x102);
    Phi_X_hat(9)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(10)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(11)=dis(y101,x101,y103,x103);
    Phi_X_hat(12)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(13)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(14)=dis(y102,x102,y100,x100);
    Phi_X_hat(15)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(16)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(17)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(18)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(19)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(20)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(21)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(22)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(23)=direction(y3000,x3000,y100,x100,w3000);
    Phi_X_hat(24)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(25)=direction(y100,x100,y1000,x1000,w100);
    %Phi_X_hat(26)=direction(y100,x100,y102,x102,w100);
    Phi_X_hat(26)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(27)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(28)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(29)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(30)=direction(y102,x102,y1000,x1000,w102);
    Phi_X_hat(31)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(32)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(33)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(34)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(35)=direction(y103,x103,y101,x101,w103);

% Final Check
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end




%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_gon = s_X(15:21,1)*200/pi;        %Convert to [gon]

%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';
```

```
%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;

%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_gon = s_L_hat(19:35)*200/pi;     %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_gon = s_v(19:35,1)*200/pi;        %Convert to [gon]

%De-reduced unknowns
    y1000 = X_0(1)+y_c;
    x1000 = X_0(2)+x_c;
    y2000 = X_0(3)+y_c;
    x2000 = X_0(4)+x_c;
    y3000 = X_0(5)+y_c;
    x3000 = X_0(6)+x_c;
    y100 = X_0(7)+y_c;
    x100 = X_0(8)+x_c;
    y101 = X_0(9)+y_c;
    x101 = X_0(10)+x_c;
    y102 = X_0(11)+y_c;
    x102 = X_0(12)+x_c;
    y103 = X_0(13)+y_c;
    x103 = X_0(14)+x_c;

X_final =  [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102
x102 y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%Convert to [gon] and check the quadrants
gon = X_final(15:21,1)*200/pi;   %always convert the rad to gon
gon = gon+400;
%gon(1) = gon(1)+400;
%gon(2) = gon(2)+400;
%gon(4) = gon(4)+400;


% Write to file: X_hat [m] for case a
% fid = fopen('X_hat_case_a', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);

% Write to file: X_hat [m] for case b
% fid = fopen('X_hat_case_b', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);

% Write to file: X_hat [m] for case c, d
% fid = fopen('X_hat_case_d', 'w');
% fprintf(fid, '%8.25f\n',X_final(1:8,:));
% fclose(fid);
```

```
%------------------------------------------------------------------------------
%  Global test
%------------------------------------------------------------------------------

%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
tx2l = chi2inv(0.025,r);  %a/2

if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%------------------------------------------------------------------------------
% Internal and external reliability parameters
%------------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_gon = GF(19:35,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end

GRZW_d= GRZW(1:18,1);
GRZW_gon = GRZW(19:35,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(17,1);

r_w(1)  = P_diag(19,1)/(P_diag(19,1)+P_diag(20,1));
r_w(2)  = P_diag(20,1)/(P_diag(19,1)+P_diag(20,1));
r_w(3)  = P_diag(21,1)/(P_diag(21,1)+P_diag(22,1));
r_w(4)  = P_diag(22,1)/(P_diag(21,1)+P_diag(22,1));
r_w(5)  = P_diag(23,1)/(P_diag(23,1)+P_diag(24,1));
r_w(6)  = P_diag(24,1)/(P_diag(23,1)+P_diag(24,1));
r_w(7)  = P_diag(25,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1));
```

```
%r_w(8) =
P_diag(26,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1)+P_diag(28,1));

r_w(8) = P_diag(26,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1));
r_w(9) = P_diag(27,1)/(P_diag(25,1)+P_diag(26,1)+P_diag(27,1));
r_w(10) = P_diag(28,1)/(P_diag(28,1)+P_diag(29,1));
r_w(11) = P_diag(29,1)/(P_diag(28,1)+P_diag(29,1));
r_w(12) = P_diag(30,1)/(P_diag(30,1)+P_diag(31,1)+P_diag(32,1));
r_w(13) = P_diag(31,1)/(P_diag(30,1)+P_diag(31,1)+P_diag(32,1));
r_w(14) = P_diag(32,1)/(P_diag(30,1)+P_diag(31,1)+P_diag(32,1));
r_w(15) = P_diag(33,1)/(P_diag(33,1)+P_diag(34,1)+P_diag(35,1));
r_w(16) = P_diag(34,1)/(P_diag(33,1)+P_diag(34,1)+P_diag(35,1));
r_w(17) = P_diag(35,1)/(P_diag(33,1)+P_diag(34,1)+P_diag(35,1));

%dd = dist(:,3);

dd(1) = distances(1,3);
dd(2) = distances(2,3);
dd(3) = distances(3,3);
dd(4) = distances(4,3);
dd(5) = distances(5,3);
dd(6) = distances(6,3);
dd(7) = distances(7,3);
dd(8) = distances(9,3);
dd(9) = distances(10,3);
dd(10) = distances(11,3);
dd(11) = distances(12,3);
dd(12) = distances(13,3);
dd(13) = distances(14,3);
dd(14) = distances(15,3);
dd(15) = distances(16,3);
dd(16) = distances(17,3);
dd(17) = distances(18,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(35,1);

for i=1:18
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end

for i=19:35
  EGK(i) = (1-EV(i,1)-r_w(i-18,1))*GRZW(i,1)*dd(i-18);
end

EGK_gon = EGK(19:35,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
EP = ones(35,1);

for i=1:18
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=19:35
  EP(i,1) = (1-EV(i,1)-r_w(i-18,1))*GF(i,1)*dd(i-18);
end
```

```
EP_gon = EP(19:35,1)*200/pi;


results.L=[L(1:18,1); directions(:,3)];
results.v=[v(1:18,1); v_gon(:,1)];
results.L=[L_hat(1:18,1); L_hat_gon(:,1)];
results.s=[s_v(1:18,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:18,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:18,1); GF_gon(:,1)];
results.GRZW=GRZW(1:18,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:18,1); EGK_gon(:,1)];
results.EP=[EP(1:18,1); EP_gon(:,1)];

results= struct2table(results);
writetable(results, 'task1_2.xls');
```

Task 1 Part 3

```
%--------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%--------------------------------------------------------------------------


clc;
clear all;
close all;
format long g;

distances = load('Distances_1.txt');
directions = load('Directions_1.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');

L = [distances(:,3); directions(:,3)*pi/200];     %Convert to [rad]

%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);
%New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
```

```
x103 = new_point(4,3);

%Initial values for orientation unknowns
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;

%Initial values for unknowns
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%---------------------------------------------------------------------------
%   Points for datum definition
%---------------------------------------------------------------------------
xy = reshape(X_0(1:14),2,7);

%Case 1, all control points for datum definition
datum = diag([1 1 1 0 0 0 0]);

%Number of points
p = sum(sum(datum));
%Centroid
x_c = (1/p)*sum(datum*xy(2,:)');
y_c = (1/p)*sum(datum*xy(1,:)');

%Coordinates reduced to the centroid
y1000 = y1000-y_c;
y2000 = y2000-y_c;
y3000 = y3000-y_c;
y100 = y100-y_c;
y101 = y101-y_c;
y102 = y102-y_c;
y103 = y103-y_c;
x1000 = x1000-x_c;
x2000 = x2000-x_c;
x3000 = x3000-x_c;
x100 = x100-x_c;
x101 = x101-x_c;
x102 = x102-x_c;
x103 = x103-x_c;

%Initial values for unknowns after reduction to the centroid
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%---------------------------------------------------------------------------

%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u + 3; %3 constraint equations
```

```
%-----------------------------------------------------------------------------
%  Stochastic model
%-----------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                              %[m]
s_dir = 0.001*pi/200;                      %Convert to [rad]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1)];
S_LL = diag(s_LL);

%Theoretical standard deviation
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
P = inv(Q_LL);

%-----------------------------------------------------------------------------
%  Adjustment
%-----------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-7; %given accuracy in 0.001
delta = 10^-11;
max_x_hat = 10^Inf;

%Number of iterations
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y101,x101);
    L_0(4)=dis(y3000,x3000,y100,x100);
    L_0(5)=dis(y3000,x3000,y103,x103);
    L_0(6)=dis(y100,x100,y1000,x1000);
    L_0(7)=dis(y100,x100,y102,x102);
    L_0(8)=dis(y100,x100,y2000,x2000);
    L_0(9)=dis(y100,x100,y3000,x3000);
    L_0(10)=dis(y101,x101,y103,x103);
    L_0(11)=dis(y101,x101,y2000,x2000);
    L_0(12)=dis(y102,x102,y1000,x1000);
    L_0(13)=dis(y102,x102,y100,x100);
    L_0(14)=dis(y102,x102,y2000,x2000);
    L_0(15)=dis(y103,x103,y3000,x3000);
    L_0(16)=dis(y103,x103,y2000,x2000);
    L_0(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    L_0(18)=direction(y1000,x1000,y100,x100,w1000);
```

```
L_0(19)=direction(y1000,x1000,y102,x102,w1000);
L_0(20)=direction(y2000,x2000,y103,x103,w2000);
L_0(21)=direction(y2000,x2000,y101,x101,w2000);
L_0(22)=direction(y3000,x3000,y100,x100,w3000);
L_0(23)=direction(y3000,x3000,y103,x103,w3000);
L_0(24)=direction(y100,x100,y1000,x1000,w100);
L_0(25)=direction(y100,x100,y2000,x2000,w100);
L_0(26)=direction(y100,x100,y3000,x3000,w100);
L_0(27)=direction(y101,x101,y103,x103,w101);
L_0(28)=direction(y101,x101,y2000,x2000,w101);
L_0(29)=direction(y102,x102,y1000,x1000,w102);
L_0(30)=direction(y102,x102,y100,x100,w102);
L_0(31)=direction(y102,x102,y2000,x2000,w102);
L_0(32)=direction(y103,x103,y3000,x3000,w103);
L_0(33)=direction(y103,x103,y2000,x2000,w103);
L_0(34)=direction(y103,x103,y101,x101,w103);

l = L-L_0';

%Design matrix
A(1,1)=ds_dy_from(y1000,x1000,y100,x100);
A(1,2)=ds_dx_from(y1000,x1000,y100,x100);
A(1,7)=ds_dy_to(y1000,x1000,y100,x100);
A(1,8)=ds_dx_to(y1000,x1000,y100,x100);

A(2,1)=ds_dy_from(y1000,x1000,y102,x102);
A(2,2)=ds_dx_from(y1000,x1000,y102,x102);
A(2,11)=ds_dy_to(y1000,x1000,y102,x102);
A(2,12)=ds_dx_to(y1000,x1000,y102,x102);


A(3,3)=ds_dy_from(y2000,x2000,y101,x101);
A(3,4)=ds_dx_from(y2000,x2000,y101,x101);
A(3,9)=ds_dy_to(y2000,x2000,y101,x101);
A(3,10)=ds_dx_to(y2000,x2000,y101,x101);

A(4,5)=ds_dy_from(y3000,x3000,y100,x100);
A(4,6)=ds_dx_from(y3000,x3000,y100,x100);
A(4,7)=ds_dy_to(y3000,x3000,y100,x100);
A(4,8)=ds_dx_to(y3000,x3000,y100,x100);

A(5,5)=ds_dy_from(y3000,x3000,y103,x103);
A(5,6)=ds_dx_from(y3000,x3000,y103,x103);
A(5,13)=ds_dy_to(y3000,x3000,y103,x103);
A(5,14)=ds_dx_to(y3000,x3000,y103,x103);

A(6,7)=ds_dy_from(y100,x100,y1000,x1000);
A(6,8)=ds_dx_from(y100,x100,y1000,x1000);
A(6,1)=ds_dy_to(y100,x100,y1000,x1000);
A(6,2)=ds_dx_to(y100,x100,y1000,x1000);

A(7,7)=ds_dy_from(y100,x100,y102,x102);
A(7,8)=ds_dx_from(y100,x100,y102,x102);
A(7,11)=ds_dy_to(y100,x100,y102,x102);
A(7,12)=ds_dx_to(y100,x100,y102,x102);

A(8,7)=ds_dy_from(y100,x100,y2000,x2000);
A(8,8)=ds_dx_from(y100,x100,y2000,x2000);
A(8,3)=ds_dy_to(y100,x100,y2000,x2000);
```

```
A(8,4)=ds_dx_to(y100,x100,y2000,x2000);

A(9,7)=ds_dy_from(y100,x100,y3000,x3000);
A(9,8)=ds_dx_from(y100,x100,y3000,x3000);
A(9,5)=ds_dy_to(y100,x100,y3000,x3000);
A(9,6)=ds_dx_to(y100,x100,y3000,x3000);

A(10,9)=ds_dy_from(y101,x101,y103,x103);
A(10,10)=ds_dx_from(y101,x101,y103,x103);
A(10,13)=ds_dy_to(y101,x101,y103,x103);
A(10,14)=ds_dx_to(y101,x101,y103,x103);

A(11,9)=ds_dy_from(y101,x101,y2000,x2000);
A(11,10)=ds_dx_from(y101,x101,y2000,x2000);
A(11,3)=ds_dy_to(y101,x101,y2000,x2000);
A(11,4)=ds_dx_to(y101,x101,y2000,x2000);

A(12,11)=ds_dy_from(y102,x102,y1000,x1000);
A(12,12)=ds_dx_from(y102,x102,y1000,x1000);
A(12,1)=ds_dy_to(y102,x102,y1000,x1000);
A(12,2)=ds_dx_to(y102,x102,y1000,x1000);

A(13,11)=ds_dy_from(y102,x102,y100,x100);
A(13,12)=ds_dx_from(y102,x102,y100,x100);
A(13,7)=ds_dy_to(y102,x102,y100,x100);
A(13,8)=ds_dx_to(y102,x102,y100,x100);

A(14,11)=ds_dy_from(y102,x102,y2000,x2000);
A(14,12)=ds_dx_from(y102,x102,y2000,x2000);
A(14,3)=ds_dy_to(y102,x102,y2000,x2000);
A(14,4)=ds_dx_to(y102,x102,y2000,x2000);

A(15,13)=ds_dy_from(y103,x103,y3000,x3000);
A(15,14)=ds_dx_from(y103,x103,y3000,x3000);
A(15,5)=ds_dy_to(y103,x103,y3000,x3000);
A(15,6)=ds_dx_to(y103,x103,y3000,x3000);

A(16,13)=ds_dy_from(y103,x103,y2000,x2000);
A(16,14)=ds_dx_from(y103,x103,y2000,x2000);
A(16,3)=ds_dy_to(y103,x103,y2000,x2000);
A(16,4)=ds_dx_to(y103,x103,y2000,x2000);

A(17,13)=ds_dy_from(y103,x103,y101,x101);
A(17,14)=ds_dx_from(y103,x103,y101,x101);
A(17,9)=ds_dy_to(y103,x103,y101,x101);
A(17,10)=ds_dx_to(y103,x103,y101,x101);

A(18,1)=dr_dy_from(y1000,x1000,y100,x100);
A(18,2)=dr_dx_from(y1000,x1000,y100,x100);
A(18,7)=dr_dy_to(y1000,x1000,y100,x100);
A(18,8)=dr_dx_to(y1000,x1000,y100,x100);
A(18,15)=-1;

A(19,1)=dr_dy_from(y1000,x1000,y102,x102);
A(19,2)=dr_dx_from(y1000,x1000,y102,x102);
A(19,11)=dr_dy_to(y1000,x1000,y102,x102);
A(19,12)=dr_dx_to(y1000,x1000,y102,x102);
A(19,15)=-1;
```

```
A(20,3)=dr_dy_from(y2000,x2000,y103,x103);
A(20,4)=dr_dx_from(y2000,x2000,y103,x103);
A(20,13)=dr_dy_to(y2000,x2000,y103,x103);
A(20,14)=dr_dx_to(y2000,x2000,y103,x103);
A(20,16)=-1;

A(21,3)=dr_dy_from(y2000,x2000,y101,x101);
A(21,4)=dr_dx_from(y2000,x2000,y101,x101);
A(21,9)=dr_dy_to(y2000,x2000,y101,x101);
A(21,10)=dr_dx_to(y2000,x2000,y101,x101);
A(21,16)=-1;

A(22,5)=dr_dy_from(y3000,x3000,y100,x100);
A(22,6)=dr_dx_from(y3000,x3000,y100,x100);
A(22,7)=dr_dy_to(y3000,x3000,y100,x100);
A(22,8)=dr_dx_to(y3000,x3000,y100,x100);
A(22,17)=-1;

A(23,5)=dr_dy_from(y3000,x3000,y103,x103);
A(23,6)=dr_dx_from(y3000,x3000,y103,x103);
A(23,13)=dr_dy_to(y3000,x3000,y103,x103);
A(23,14)=dr_dx_to(y3000,x3000,y103,x103);
A(23,17)=-1;

A(24,7)=dr_dy_from(y100,x100,y1000,x1000);
A(24,8)=dr_dx_from(y100,x100,y1000,x1000);
A(24,1)=dr_dy_to(y100,x100,y1000,x1000);
A(24,2)=dr_dx_to(y100,x100,y1000,x1000);
A(24,18)=-1;

A(25,7)=dr_dy_from(y100,x100,y2000,x2000);
A(25,8)=dr_dx_from(y100,x100,y2000,x2000);
A(25,3)=dr_dy_to(y100,x100,y2000,x2000);
A(25,4)=dr_dx_to(y100,x100,y2000,x2000);
A(25,18)=-1;

A(26,7)=dr_dy_from(y100,x100,y3000,x3000);
A(26,8)=dr_dx_from(y100,x100,y3000,x3000);
A(26,5)=dr_dy_to(y100,x100,y3000,x3000);
A(26,6)=dr_dx_to(y100,x100,y3000,x3000);
A(26,18)=-1;

A(27,9)=dr_dy_from(y101,x101,y103,x103);
A(27,10)=dr_dx_from(y101,x101,y103,x103);
A(27,13)=dr_dy_to(y101,x101,y103,x103);
A(27,14)=dr_dx_to(y101,x101,y103,x103);
A(27,19)=-1;

A(28,9)=dr_dy_from(y101,x101,y2000,x2000);
A(28,10)=dr_dx_from(y101,x101,y2000,x2000);
A(28,3)=dr_dy_to(y101,x101,y2000,x2000);
A(28,4)=dr_dx_to(y101,x101,y2000,x2000);
A(28,19)=-1;

A(29,11)=dr_dy_from(y102,x102,y1000,x1000);
A(29,12)=dr_dx_from(y102,x102,y1000,x1000);
A(29,1)=dr_dy_to(y102,x102,y1000,x1000);
A(29,2)=dr_dx_to(y102,x102,y1000,x1000);
A(29,20)=-1;
```

```
A(30,11)=dr_dy_from(y102,x102,y100,x100);
A(30,12)=dr_dx_from(y102,x102,y100,x100);
A(30,7)=dr_dy_to(y102,x102,y100,x100);
A(30,8)=dr_dx_to(y102,x102,y100,x100);
A(30,20)=-1;

A(31,11)=dr_dy_from(y102,x102,y2000,x2000);
A(31,12)=dr_dx_from(y102,x102,y2000,x2000);
A(31,3)=dr_dy_to(y102,x102,y2000,x2000);
A(31,4)=dr_dx_to(y102,x102,y2000,x2000);
A(31,20)=-1;

A(32,13)=dr_dy_from(y103,x103,y3000,x3000);
A(32,14)=dr_dx_from(y103,x103,y3000,x3000);
A(32,5)=dr_dy_to(y103,x103,y3000,x3000);
A(32,6)=dr_dx_to(y103,x103,y3000,x3000);
A(32,21)=-1;

A(33,13)=dr_dy_from(y103,x103,y2000,x2000);
A(33,14)=dr_dx_from(y103,x103,y2000,x2000);
A(33,3)=dr_dy_to(y103,x103,y2000,x2000);
A(33,4)=dr_dx_to(y103,x103,y2000,x2000);
A(33,21)=-1;

A(34,13)=dr_dy_from(y103,x103,y101,x101);
A(34,14)=dr_dx_from(y103,x103,y101,x101);
A(34,9)=dr_dy_to(y103,x103,y101,x101);
A(34,10)=dr_dx_to(y103,x103,y101,x101);
A(34,21)=-1;


%Normal matrix
N = A'*P*A;


%Zero matrix
Zz = zeros(3,3);    %c constraints

%B matrix
 %B matrix
B = [1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
     -x1000 y1000 -x2000 y2000 -x3000 y3000 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 ];

%Extended normal matrix
%For case a, b, d
 N_ext = [N B'; B Zz];

%For case c
 %N_ext = [N G'; G Zz];

%Vector of right hand side of normal equations
 n = A'*P*l;

%Extended vector of right hand side of normal equations
 n_ext = [n;0;0;0];
```

```
    %Inversion of normal matrix / Cofactor matrix of the unknowns
     Q_xx_ext = inv(N_ext);

    %Reduced cofactor matrix of the unknowns
     Q_xx = Q_xx_ext(1:no_u,1:no_u);    %Q11

    %Solution of normal equation
     x_hat = Q_xx_ext*n_ext;

    %Adjusted unknowns
     X_hat = X_0+x_hat(1:21);

    %Update
    X_0 = X_hat;

    y1000 = X_0(1);
    x1000 = X_0(2);
    y2000 = X_0(3);
    x2000 = X_0(4);
    y3000 = X_0(5);
    x3000 = X_0(6);
    y100 = X_0(7);
    x100 = X_0(8);
    y101 = X_0(9);
    x101 = X_0(10);
    y102 = X_0(11);
    x102 = X_0(12);
    y103 = X_0(13);
    x103 = X_0(14);
    w1000 = X_0(15);
    w2000 = X_0(16);
    w3000 = X_0(17);
    w100 = X_0(18);
    w101 = X_0(19);
    w102 = X_0(20);
    w103 = X_0(21);

    %Check 1
     max_x_hat = max(abs(x_hat));

    %Update number of iterations
    iteration=iteration+1;

end


%Vector of residuals
v = A*x_hat(1:21)-l;
v_gon = v(18:34,1)*200/pi;             %Convert to [gon]

%Objective function
vTPv = v'*P*v;

%Vector of adjusted observations
L_hat = L+v;
L_hat_gon = L_hat(18:34)*200/pi;      %Convert to [gon]
```

```
%Final check for the linearization

    %Vector of reduced distances
    Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
    Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
    Phi_X_hat(3)=dis(y2000,x2000,y101,x101);
    Phi_X_hat(4)=dis(y3000,x3000,y100,x100);
    Phi_X_hat(5)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(6)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(7)=dis(y100,x100,y102,x102);
    Phi_X_hat(8)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(9)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(10)=dis(y101,x101,y103,x103);
    Phi_X_hat(11)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(12)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(13)=dis(y102,x102,y100,x100);
    Phi_X_hat(14)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(15)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(16)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(18)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(19)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(20)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(21)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(22)=direction(y3000,x3000,y100,x100,w3000);
    Phi_X_hat(23)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(24)=direction(y100,x100,y1000,x1000,w100);
    Phi_X_hat(25)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(26)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(27)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(28)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(29)=direction(y102,x102,y1000,x1000,w102);
    Phi_X_hat(30)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(31)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(32)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(33)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(34)=direction(y103,x103,y101,x101,w103);

% Final Check
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end

%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_gon = s_X(15:21,1)*200/pi;        %Convert to [gon]
```

```
%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';

%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;

%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_gon = s_L_hat(18:34)*200/pi;     %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_gon = s_v(18:34,1)*200/pi;        %Convert to [gon]

%De-reduced unknowns
    y1000 = X_0(1)+y_c;
    x1000 = X_0(2)+x_c;
    y2000 = X_0(3)+y_c;
    x2000 = X_0(4)+x_c;
    y3000 = X_0(5)+y_c;
    x3000 = X_0(6)+x_c;
    y100  = X_0(7)+y_c;
    x100  = X_0(8)+x_c;
    y101  = X_0(9)+y_c;
    x101  = X_0(10)+x_c;
    y102  = X_0(11)+y_c;
    x102  = X_0(12)+x_c;
    y103  = X_0(13)+y_c;
    x103  = X_0(14)+x_c;

X_final =  [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102
x102 y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';

%Convert to [gon] and check the quadrants
gon = X_final(15:21,1)*200/pi;   %always convert the rad to gon
gon = gon+400;
%gon(1)  = gon(1)+400;
%gon(2)  = gon(2)+400;
%gon(4)  = gon(4)+400;



%---------------------------------------------------------------------------
%  Global test
%---------------------------------------------------------------------------

%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
```

```
tx2l = chi2inv(0.025,r);  %a/2

if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%---------------------------------------------------------------------------
% Internal and external reliability parameters
%---------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_gon = GF(18:34,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end
GRZW_d= GRZW(1:18,1);
GRZW_gon = GRZW(18:34,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(17,1);

r_w(1)  = P_diag(18,1)/(P_diag(18,1)+P_diag(19,1));
r_w(2)  = P_diag(19,1)/(P_diag(18,1)+P_diag(19,1));
r_w(3)  = P_diag(20,1)/(P_diag(20,1)+P_diag(21,1));
r_w(4)  = P_diag(21,1)/(P_diag(20,1)+P_diag(21,1));
r_w(5)  = P_diag(22,1)/(P_diag(22,1)+P_diag(23,1));
r_w(6)  = P_diag(23,1)/(P_diag(22,1)+P_diag(23,1));
r_w(7)  = P_diag(24,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
%r_w(8)  =
P_diag(25,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1)+P_diag(27,1));
r_w(8)  = P_diag(25,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(9)  = P_diag(26,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(10) = P_diag(27,1)/(P_diag(27,1)+P_diag(28,1));
r_w(11) = P_diag(28,1)/(P_diag(27,1)+P_diag(28,1));
r_w(12) = P_diag(29,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(13) = P_diag(30,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(14) = P_diag(31,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(15) = P_diag(32,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(16) = P_diag(33,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(17) = P_diag(34,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
```

```
%dd = dist(:,3);

dd(1) = distances(1,3);
dd(2) = distances(2,3);
dd(3) = distances(15,3);
dd(4) = distances(3,3);
dd(5) = distances(4,3);
dd(6) = distances(5,3);
dd(7) = distances(6,3);
dd(8) = distances(8,3);
dd(9) = distances(9,3);
dd(10) = distances(10,3);
dd(11) = distances(11,3);
dd(12) = distances(12,3);
dd(13) = distances(13,3);
dd(14) = distances(14,3);
dd(15) = distances(15,3);
dd(16) = distances(16,3);
dd(17) = distances(17,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(34,1);

for i=1:17
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end

for i=18:34
  EGK(i) = (1-EV(i,1)-r_w(i-17,1))*GRZW(i,1)*dd(i-17);
end

EGK_gon = EGK(18:34,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
EP = ones(34,1);

for i=1:17
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=18:34
  EP(i,1) = (1-EV(i,1)-r_w(i-17,1))*GF(i,1)*dd(i-17);
end

EP_gon = EP(18:34,1)*200/pi;


%Display the result
results.L=[L(1:18,1); directions(:,3)];
results.v=[v(1:18,1); v_gon(:,1)];
results.L=[L_hat(1:18,1); L_hat_gon(:,1)];
results.s=[s_v(1:18,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:18,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:18,1); GF_gon(:,1)];
results.GRZW=GRZW(1:18,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:18,1); EGK_gon(:,1)];
results.EP=[EP(1:18,1); EP_gon(:,1)];
```

```
results= struct2table(results);
writetable(results, 'task1_3.xls');
```

Task 2 part 1

```
%--------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%--------------------------------------------------------------------------


clc;
clear all;
close all;
format long g;

distances = load('Distances_1.txt');
directions = load('Directions_1.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');

%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);
%New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
x103 = new_point(4,3);

%Initial values for orientation unknowns
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;


L = [distances(:,3); directions(:,3)*pi/200; y1000; x1000; y2000; x2000;
y3000; x3000];    %Convert to [rad]
```

```
%Initial values for unknowns
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';


%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u;

xy = reshape(X_0(1:6),1,6);

%---------------------------------------------------------------------------
%  Stochastic model
%---------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                         %[m]
s_dir = 0.001*pi/200;                   %Convert to [rad]
s_xy = 0.01;                            %[m]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1); s_xy^2*ones(length(xy),1)];
S_LL = diag(s_LL);

%Theoretical standard deviation
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
P = inv(Q_LL);

%---------------------------------------------------------------------------
%  Adjustment
%---------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-7; %given accuracy in 0.001
delta = 10^-9;
max_x_hat = 10^Inf;

%Number of iterations
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y101,x101);
```

```
L_0(4)=dis(y3000,x3000,y100,x100);
L_0(5)=dis(y3000,x3000,y103,x103);
L_0(6)=dis(y100,x100,y1000,x1000);
L_0(7)=dis(y100,x100,y102,x102);
L_0(8)=dis(y100,x100,y2000,x2000);
L_0(9)=dis(y100,x100,y3000,x3000);
L_0(10)=dis(y101,x101,y103,x103);
L_0(11)=dis(y101,x101,y2000,x2000);
L_0(12)=dis(y102,x102,y1000,x1000);
L_0(13)=dis(y102,x102,y100,x100);
L_0(14)=dis(y102,x102,y2000,x2000);
L_0(15)=dis(y103,x103,y3000,x3000);
L_0(16)=dis(y103,x103,y2000,x2000);
L_0(17)=dis(y103,x103,y101,x101);

%Vector of reduced directions
L_0(18)=direction(y1000,x1000,y100,x100,w1000);
L_0(19)=direction(y1000,x1000,y102,x102,w1000);
L_0(20)=direction(y2000,x2000,y103,x103,w2000);
L_0(21)=direction(y2000,x2000,y101,x101,w2000);
L_0(22)=direction(y3000,x3000,y100,x100,w3000);
L_0(23)=direction(y3000,x3000,y103,x103,w3000);
L_0(24)=direction(y100,x100,y1000,x1000,w100);
L_0(25)=direction(y100,x100,y2000,x2000,w100);
L_0(26)=direction(y100,x100,y3000,x3000,w100);
L_0(27)=direction(y101,x101,y103,x103,w101);
L_0(28)=direction(y101,x101,y2000,x2000,w101);
L_0(29)=direction(y102,x102,y1000,x1000,w102);
L_0(30)=direction(y102,x102,y100,x100,w102);
L_0(31)=direction(y102,x102,y2000,x2000,w102);
L_0(32)=direction(y103,x103,y3000,x3000,w103);
L_0(33)=direction(y103,x103,y2000,x2000,w103);
L_0(34)=direction(y103,x103,y101,x101,w103);

%Observed unknowns
L_0(35) = y1000;
L_0(36) = x1000;
L_0(37) = y2000;
L_0(38) = x2000;
L_0(39) = y3000;
L_0(40) = x3000;


l = L-L_0';

%Design matrix
A(1,1)=ds_dy_from(y1000,x1000,y100,x100);
A(1,2)=ds_dx_from(y1000,x1000,y100,x100);
A(1,7)=ds_dy_to(y1000,x1000,y100,x100);
A(1,8)=ds_dx_to(y1000,x1000,y100,x100);

A(2,1)=ds_dy_from(y1000,x1000,y102,x102);
A(2,2)=ds_dx_from(y1000,x1000,y102,x102);
A(2,11)=ds_dy_to(y1000,x1000,y102,x102);
A(2,12)=ds_dx_to(y1000,x1000,y102,x102);


A(3,3)=ds_dy_from(y2000,x2000,y101,x101);
A(3,4)=ds_dx_from(y2000,x2000,y101,x101);
A(3,9)=ds_dy_to(y2000,x2000,y101,x101);
```

```
A(3,10)=ds_dx_to(y2000,x2000,y101,x101);

A(4,5)=ds_dy_from(y3000,x3000,y100,x100);
A(4,6)=ds_dx_from(y3000,x3000,y100,x100);
A(4,7)=ds_dy_to(y3000,x3000,y100,x100);
A(4,8)=ds_dx_to(y3000,x3000,y100,x100);

A(5,5)=ds_dy_from(y3000,x3000,y103,x103);
A(5,6)=ds_dx_from(y3000,x3000,y103,x103);
A(5,13)=ds_dy_to(y3000,x3000,y103,x103);
A(5,14)=ds_dx_to(y3000,x3000,y103,x103);

A(6,7)=ds_dy_from(y100,x100,y1000,x1000);
A(6,8)=ds_dx_from(y100,x100,y1000,x1000);
A(6,1)=ds_dy_to(y100,x100,y1000,x1000);
A(6,2)=ds_dx_to(y100,x100,y1000,x1000);

A(7,7)=ds_dy_from(y100,x100,y102,x102);
A(7,8)=ds_dx_from(y100,x100,y102,x102);
A(7,11)=ds_dy_to(y100,x100,y102,x102);
A(7,12)=ds_dx_to(y100,x100,y102,x102);

A(8,7)=ds_dy_from(y100,x100,y2000,x2000);
A(8,8)=ds_dx_from(y100,x100,y2000,x2000);
A(8,3)=ds_dy_to(y100,x100,y2000,x2000);
A(8,4)=ds_dx_to(y100,x100,y2000,x2000);

A(9,7)=ds_dy_from(y100,x100,y3000,x3000);
A(9,8)=ds_dx_from(y100,x100,y3000,x3000);
A(9,5)=ds_dy_to(y100,x100,y3000,x3000);
A(9,6)=ds_dx_to(y100,x100,y3000,x3000);

A(10,9)=ds_dy_from(y101,x101,y103,x103);
A(10,10)=ds_dx_from(y101,x101,y103,x103);
A(10,13)=ds_dy_to(y101,x101,y103,x103);
A(10,14)=ds_dx_to(y101,x101,y103,x103);

A(11,9)=ds_dy_from(y101,x101,y2000,x2000);
A(11,10)=ds_dx_from(y101,x101,y2000,x2000);
A(11,3)=ds_dy_to(y101,x101,y2000,x2000);
A(11,4)=ds_dx_to(y101,x101,y2000,x2000);

A(12,11)=ds_dy_from(y102,x102,y1000,x1000);
A(12,12)=ds_dx_from(y102,x102,y1000,x1000);
A(12,1)=ds_dy_to(y102,x102,y1000,x1000);
A(12,2)=ds_dx_to(y102,x102,y1000,x1000);

A(13,11)=ds_dy_from(y102,x102,y100,x100);
A(13,12)=ds_dx_from(y102,x102,y100,x100);
A(13,7)=ds_dy_to(y102,x102,y100,x100);
A(13,8)=ds_dx_to(y102,x102,y100,x100);

A(14,11)=ds_dy_from(y102,x102,y2000,x2000);
A(14,12)=ds_dx_from(y102,x102,y2000,x2000);
A(14,3)=ds_dy_to(y102,x102,y2000,x2000);
A(14,4)=ds_dx_to(y102,x102,y2000,x2000);

A(15,13)=ds_dy_from(y103,x103,y3000,x3000);
A(15,14)=ds_dx_from(y103,x103,y3000,x3000);
```

```
A(15,5)=ds_dy_to(y103,x103,y3000,x3000);
A(15,6)=ds_dx_to(y103,x103,y3000,x3000);

A(16,13)=ds_dy_from(y103,x103,y2000,x2000);
A(16,14)=ds_dx_from(y103,x103,y2000,x2000);
A(16,3)=ds_dy_to(y103,x103,y2000,x2000);
A(16,4)=ds_dx_to(y103,x103,y2000,x2000);

A(17,13)=ds_dy_from(y103,x103,y101,x101);
A(17,14)=ds_dx_from(y103,x103,y101,x101);
A(17,9)=ds_dy_to(y103,x103,y101,x101);
A(17,10)=ds_dx_to(y103,x103,y101,x101);


A(18,1)=dr_dy_from(y1000,x1000,y100,x100);
A(18,2)=dr_dx_from(y1000,x1000,y100,x100);
A(18,7)=dr_dy_to(y1000,x1000,y100,x100);
A(18,8)=dr_dx_to(y1000,x1000,y100,x100);
A(18,15)=-1;

A(19,1)=dr_dy_from(y1000,x1000,y102,x102);
A(19,2)=dr_dx_from(y1000,x1000,y102,x102);
A(19,11)=dr_dy_to(y1000,x1000,y102,x102);
A(19,12)=dr_dx_to(y1000,x1000,y102,x102);
A(19,15)=-1;

A(20,3)=dr_dy_from(y2000,x2000,y103,x103);
A(20,4)=dr_dx_from(y2000,x2000,y103,x103);
A(20,13)=dr_dy_to(y2000,x2000,y103,x103);
A(20,14)=dr_dx_to(y2000,x2000,y103,x103);
A(20,16)=-1;

A(21,3)=dr_dy_from(y2000,x2000,y101,x101);
A(21,4)=dr_dx_from(y2000,x2000,y101,x101);
A(21,9)=dr_dy_to(y2000,x2000,y101,x101);
A(21,10)=dr_dx_to(y2000,x2000,y101,x101);
A(21,16)=-1;

A(22,5)=dr_dy_from(y3000,x3000,y100,x100);
A(22,6)=dr_dx_from(y3000,x3000,y100,x100);
A(22,7)=dr_dy_to(y3000,x3000,y100,x100);
A(22,8)=dr_dx_to(y3000,x3000,y100,x100);
A(22,17)=-1;

A(23,5)=dr_dy_from(y3000,x3000,y103,x103);
A(23,6)=dr_dx_from(y3000,x3000,y103,x103);
A(23,13)=dr_dy_to(y3000,x3000,y103,x103);
A(23,14)=dr_dx_to(y3000,x3000,y103,x103);
A(23,17)=-1;

A(24,7)=dr_dy_from(y100,x100,y1000,x1000);
A(24,8)=dr_dx_from(y100,x100,y1000,x1000);
A(24,1)=dr_dy_to(y100,x100,y1000,x1000);
A(24,2)=dr_dx_to(y100,x100,y1000,x1000);
A(24,18)=-1;


A(25,7)=dr_dy_from(y100,x100,y2000,x2000);
A(25,8)=dr_dx_from(y100,x100,y2000,x2000);
```

```
A(25,3)=dr_dy_to(y100,x100,y2000,x2000);
A(25,4)=dr_dx_to(y100,x100,y2000,x2000);
A(25,18)=-1;

A(26,7)=dr_dy_from(y100,x100,y3000,x3000);
A(26,8)=dr_dx_from(y100,x100,y3000,x3000);
A(26,5)=dr_dy_to(y100,x100,y3000,x3000);
A(26,6)=dr_dx_to(y100,x100,y3000,x3000);
A(26,18)=-1;

A(27,9)=dr_dy_from(y101,x101,y103,x103);
A(27,10)=dr_dx_from(y101,x101,y103,x103);
A(27,13)=dr_dy_to(y101,x101,y103,x103);
A(27,14)=dr_dx_to(y101,x101,y103,x103);
A(27,19)=-1;

A(28,9)=dr_dy_from(y101,x101,y2000,x2000);
A(28,10)=dr_dx_from(y101,x101,y2000,x2000);
A(28,3)=dr_dy_to(y101,x101,y2000,x2000);
A(28,4)=dr_dx_to(y101,x101,y2000,x2000);
A(28,19)=-1;

A(29,11)=dr_dy_from(y102,x102,y1000,x1000);
A(29,12)=dr_dx_from(y102,x102,y1000,x1000);
A(29,1)=dr_dy_to(y102,x102,y1000,x1000);
A(29,2)=dr_dx_to(y102,x102,y1000,x1000);
A(29,20)=-1;

A(30,11)=dr_dy_from(y102,x102,y100,x100);
A(30,12)=dr_dx_from(y102,x102,y100,x100);
A(30,7)=dr_dy_to(y102,x102,y100,x100);
A(30,8)=dr_dx_to(y102,x102,y100,x100);
A(30,20)=-1;

A(31,11)=dr_dy_from(y102,x102,y2000,x2000);
A(31,12)=dr_dx_from(y102,x102,y2000,x2000);
A(31,3)=dr_dy_to(y102,x102,y2000,x2000);
A(31,4)=dr_dx_to(y102,x102,y2000,x2000);
A(31,20)=-1;

A(32,13)=dr_dy_from(y103,x103,y3000,x3000);
A(32,14)=dr_dx_from(y103,x103,y3000,x3000);
A(32,5)=dr_dy_to(y103,x103,y3000,x3000);
A(32,6)=dr_dx_to(y103,x103,y3000,x3000);
A(32,21)=-1;

A(33,13)=dr_dy_from(y103,x103,y2000,x2000);
A(33,14)=dr_dx_from(y103,x103,y2000,x2000);
A(33,3)=dr_dy_to(y103,x103,y2000,x2000);
A(33,4)=dr_dx_to(y103,x103,y2000,x2000);
A(33,21)=-1;

A(34,13)=dr_dy_from(y103,x103,y101,x101);
A(34,14)=dr_dx_from(y103,x103,y101,x101);
A(34,9)=dr_dy_to(y103,x103,y101,x101);
A(34,10)=dr_dx_to(y103,x103,y101,x101);
A(34,21)=-1;

A(35,1)=1;
```

```
    A(36,2)=1;
    A(37,3)=1;
    A(38,4)=1;
    A(39,5)=1;
    A(40,6)=1;


    %Normal matrix
    N = A'*P*A;


    %Vector of right hand side of normal equations
     n = A'*P*l;

    %Inversion of normal matrix / Cofactor matrix of the unknowns
     Q_xx = inv(N);


    %Solution of normal equation
     x_hat = Q_xx*n;

    %Adjusted unknowns
     X_hat = X_0+x_hat;

    %Update
    X_0 = X_hat;

    y1000 = X_0(1);
    x1000 = X_0(2);
    y2000 = X_0(3);
    x2000 = X_0(4);
    y3000 = X_0(5);
    x3000 = X_0(6);
    y100 = X_0(7);
    x100 = X_0(8);
    y101 = X_0(9);
    x101 = X_0(10);
    y102 = X_0(11);
    x102 = X_0(12);
    y103 = X_0(13);
    x103 = X_0(14);
    w1000 = X_0(15);
    w2000 = X_0(16);
    w3000 = X_0(17);
    w100 = X_0(18);
    w101 = X_0(19);
    w102 = X_0(20);
    w103 = X_0(21);

    %Check 1
     max_x_hat = max(abs(x_hat));

    %Update number of iterations
    iteration=iteration+1;

end

%Convert to [gon] and check the quadrants
```

```
gon = X_0(15:21,1)*200/pi;      %convert w to gon

gon = gon+400;

%gon(1)  = gon(1)+400;           %check w1-w9 in X_0 >> check the quadrants
manually >> add +400 for the minus value
%gon(2)  = gon(2)+400;
%gon(3)  = gon(3)+400;
%gon(4)  = gon(4)+400;




%Vector of residuals
v = A*x_hat-l;
v_gon = v(18:34,1)*200/pi;              %Convert to [gon]

%Objective function
vTPv = v'*P*v;

%Vector of adjusted observations
L_hat = L+v;
L_hat_gon = L_hat(18:34)*200/pi;      %Convert to [gon]


%Final check for the linearization

    %Vector of reduced distances
    Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
    Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
    Phi_X_hat(3)=dis(y2000,x2000,y101,x101);
    Phi_X_hat(4)=dis(y3000,x3000,y100,x100);
    Phi_X_hat(5)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(6)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(7)=dis(y100,x100,y102,x102);
    Phi_X_hat(8)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(9)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(10)=dis(y101,x101,y103,x103);
    Phi_X_hat(11)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(12)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(13)=dis(y102,x102,y100,x100);
    Phi_X_hat(14)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(15)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(16)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(18)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(19)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(20)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(21)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(22)=direction(y3000,x3000,y100,x100,w3000);
    Phi_X_hat(23)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(24)=direction(y100,x100,y1000,x1000,w100);
    Phi_X_hat(25)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(26)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(27)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(28)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(29)=direction(y102,x102,y1000,x1000,w102);
```

```
    Phi_X_hat(30)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(31)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(32)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(33)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(34)=direction(y103,x103,y101,x101,w103);

    % Control points
    Phi_X_hat(35) = y1000;
    Phi_X_hat(36) = x1000;
    Phi_X_hat(37) = y2000;
    Phi_X_hat(38) = x2000;
    Phi_X_hat(39) = y3000;
   Phi_X_hat(40) = x3000;

% Final Check
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end




%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_gon = s_X(15:21,1)*200/pi;        %Convert to [gon]

%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';

%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;

%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_gon = s_L_hat(18:34)*200/pi;    %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_gon = s_v(18:34,1)*200/pi;        %Convert to [gon]
```

```
%--------------------------------------------------------------------------
%  Global test
%--------------------------------------------------------------------------

%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
tx2l = chi2inv(0.025,r);  %a/2

if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%--------------------------------------------------------------------------
% Internal and external reliability parameters
%--------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_gon = GF(18:34,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end
GRZW_d= GRZW(1:17,1);
GRZW_gon = GRZW(18:34,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(17,1);

r_w(1)  = P_diag(18,1)/(P_diag(18,1)+P_diag(19,1));
r_w(2)  = P_diag(19,1)/(P_diag(18,1)+P_diag(19,1));
r_w(3)  = P_diag(20,1)/(P_diag(20,1)+P_diag(21,1));
r_w(4)  = P_diag(21,1)/(P_diag(20,1)+P_diag(21,1));
r_w(5)  = P_diag(22,1)/(P_diag(22,1)+P_diag(23,1));
r_w(6)  = P_diag(23,1)/(P_diag(22,1)+P_diag(23,1));
r_w(7)  = P_diag(24,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(8)  = P_diag(25,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(9)  = P_diag(26,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(10) = P_diag(27,1)/(P_diag(27,1)+P_diag(28,1));
```

```
r_w(11) = P_diag(28,1)/(P_diag(27,1)+P_diag(28,1));
r_w(12) = P_diag(29,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(13) = P_diag(30,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(14) = P_diag(31,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(15) = P_diag(32,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(16) = P_diag(33,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(17) = P_diag(34,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));

%dd = dist(:,3);

dd(1) = distances(1,3);
dd(2) = distances(2,3);
dd(3) = distances(15,3);
dd(4) = distances(3,3);
dd(5) = distances(4,3);
dd(6) = distances(5,3);
dd(7) = distances(6,3);
dd(8) = distances(8,3);
dd(9) = distances(9,3);
dd(10) = distances(10,3);
dd(11) = distances(11,3);
dd(12) = distances(12,3);
dd(13) = distances(13,3);
dd(14) = distances(14,3);
dd(15) = distances(15,3);
dd(16) = distances(16,3);
dd(17) = distances(17,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(40,1);
for i=1:17
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end

for i=18:34
  EGK(i) = (1-EV(i,1)-r_w(i-17,1))*GRZW(i,1)*dd(i-17);
end

for i=35:40
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end


EGK_gon = EGK(18:34,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
EP = ones(40,1);

for i=1:17
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=18:34
  EP(i,1) = (1-EV(i,1)-r_w(i-17,1))*GF(i,1)*dd(i-17);
end

for i=35:40
  EP(i,1) = (1-EV(i,1))*GF(i,1);
```

```
end

EP_gon = EP(18:34,1)*200/pi;


results.L=[L(1:18,1); directions(:,3)];
results.v=[v(1:18,1); v_gon(:,1)];
results.L=[L_hat(1:18,1); L_hat_gon(:,1)];
results.s=[s_v(1:18,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:18,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:18,1); GF_gon(:,1)];
results.GRZW=[GRZW(1:18,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:18,1); EGK_gon(:,1)];
results.EP=[EP(1:18,1); EP_gon(:,1)];

results= struct2table(results);
writetable(results, 'task2_1.xls');
```

Task 2 part 2

```
%--------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%--------------------------------------------------------------------------


clc;
clear all;
close all;
format long g;

distances = load('Distances_1.txt');
directions = load('Directions_1.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');
%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);
%New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
x103 = new_point(4,3);

%Initial values for orientation unknowns
```

```
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;

L_val= [distances(:,3); directions(:,3); y1000; x1000; y2000; x2000]
L = [distances(:,3); directions(:,3)*pi/200; y1000; x1000; y2000; x2000];


%Initial values for unknowns
X_0 = [y1000 x1000 y2000 x2000 y3000 x3000 y100 x100 y101 x101 y102 x102
y103 x103 w1000 w2000 w3000 w100 w101 w102 w103]';


%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u;

xy = reshape(X_0(1:4),1,4);

%--------------------------------------------------------------------------
%  Stochastic model
%--------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                      %[m]
s_dir = 0.001*pi/200;                %Convert to [rad]
s_xy = 0.01;                         %[m]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1); s_xy^2*ones(length(xy),1)];
S_LL = diag(s_LL);

%Theoretical standard deviation
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
P = inv(Q_LL);

%--------------------------------------------------------------------------
%  Adjustment
%--------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-7; %given accuracy in 0.001
delta = 10^-9;
max_x_hat = 10^Inf;

%Number of iterations
```

```
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y101,x101);
    L_0(4)=dis(y3000,x3000,y100,x100);
    L_0(5)=dis(y3000,x3000,y103,x103);
    L_0(6)=dis(y100,x100,y1000,x1000);
    L_0(7)=dis(y100,x100,y102,x102);
    L_0(8)=dis(y100,x100,y2000,x2000);
    L_0(9)=dis(y100,x100,y3000,x3000);
    L_0(10)=dis(y101,x101,y103,x103);
    L_0(11)=dis(y101,x101,y2000,x2000);
    L_0(12)=dis(y102,x102,y1000,x1000);
    L_0(13)=dis(y102,x102,y100,x100);
    L_0(14)=dis(y102,x102,y2000,x2000);
    L_0(15)=dis(y103,x103,y3000,x3000);
    L_0(16)=dis(y103,x103,y2000,x2000);
    L_0(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    L_0(18)=direction(y1000,x1000,y100,x100,w1000);
    L_0(19)=direction(y1000,x1000,y102,x102,w1000);
    L_0(20)=direction(y2000,x2000,y103,x103,w2000);
    L_0(21)=direction(y2000,x2000,y101,x101,w2000);
    L_0(22)=direction(y3000,x3000,y100,x100,w3000);
    L_0(23)=direction(y3000,x3000,y103,x103,w3000);
    L_0(24)=direction(y100,x100,y1000,x1000,w100);
    L_0(25)=direction(y100,x100,y2000,x2000,w100);
    L_0(26)=direction(y100,x100,y3000,x3000,w100);
    L_0(27)=direction(y101,x101,y103,x103,w101);
    L_0(28)=direction(y101,x101,y2000,x2000,w101);
    L_0(29)=direction(y102,x102,y1000,x1000,w102);
    L_0(30)=direction(y102,x102,y100,x100,w102);
    L_0(31)=direction(y102,x102,y2000,x2000,w102);
    L_0(32)=direction(y103,x103,y3000,x3000,w103);
    L_0(33)=direction(y103,x103,y2000,x2000,w103);
    L_0(34)=direction(y103,x103,y101,x101,w103);

    %Observed unknowns
    L_0(35) = y1000;
    L_0(36) = x1000;
    L_0(37) = y2000;
    L_0(38) = x2000;


    l = L-L_0';

    %Design matrix
    A(1,1)=ds_dy_from(y1000,x1000,y100,x100);
    A(1,2)=ds_dx_from(y1000,x1000,y100,x100);
    A(1,7)=ds_dy_to(y1000,x1000,y100,x100);
    A(1,8)=ds_dx_to(y1000,x1000,y100,x100);
```

```
A(2,1)=ds_dy_from(y1000,x1000,y102,x102);
A(2,2)=ds_dx_from(y1000,x1000,y102,x102);
A(2,11)=ds_dy_to(y1000,x1000,y102,x102);
A(2,12)=ds_dx_to(y1000,x1000,y102,x102);


A(3,3)=ds_dy_from(y2000,x2000,y101,x101);
A(3,4)=ds_dx_from(y2000,x2000,y101,x101);
A(3,9)=ds_dy_to(y2000,x2000,y101,x101);
A(3,10)=ds_dx_to(y2000,x2000,y101,x101);

A(4,5)=ds_dy_from(y3000,x3000,y100,x100);
A(4,6)=ds_dx_from(y3000,x3000,y100,x100);
A(4,7)=ds_dy_to(y3000,x3000,y100,x100);
A(4,8)=ds_dx_to(y3000,x3000,y100,x100);

A(5,5)=ds_dy_from(y3000,x3000,y103,x103);
A(5,6)=ds_dx_from(y3000,x3000,y103,x103);
A(5,13)=ds_dy_to(y3000,x3000,y103,x103);
A(5,14)=ds_dx_to(y3000,x3000,y103,x103);

A(6,7)=ds_dy_from(y100,x100,y1000,x1000);
A(6,1)=ds_dy_to(y100,x100,y1000,x1000);
A(6,2)=ds_dx_to(y100,x100,y1000,x1000);

A(7,7)=ds_dy_from(y100,x100,y102,x102);
A(7,8)=ds_dx_from(y100,x100,y102,x102);
A(7,11)=ds_dy_to(y100,x100,y102,x102);
A(7,12)=ds_dx_to(y100,x100,y102,x102);

A(8,7)=ds_dy_from(y100,x100,y2000,x2000);
A(8,8)=ds_dx_from(y100,x100,y2000,x2000);
A(8,3)=ds_dy_to(y100,x100,y2000,x2000);
A(8,4)=ds_dx_to(y100,x100,y2000,x2000);

A(9,7)=ds_dy_from(y100,x100,y3000,x3000);
A(9,8)=ds_dx_from(y100,x100,y3000,x3000);
A(9,5)=ds_dy_to(y100,x100,y3000,x3000);
A(9,6)=ds_dx_to(y100,x100,y3000,x3000);

A(10,9)=ds_dy_from(y101,x101,y103,x103);
A(10,10)=ds_dx_from(y101,x101,y103,x103);
A(10,13)=ds_dy_to(y101,x101,y103,x103);
A(10,14)=ds_dx_to(y101,x101,y103,x103);

A(11,9)=ds_dy_from(y101,x101,y2000,x2000);
A(11,10)=ds_dx_from(y101,x101,y2000,x2000);
A(11,3)=ds_dy_to(y101,x101,y2000,x2000);
A(11,4)=ds_dx_to(y101,x101,y2000,x2000);

A(12,11)=ds_dy_from(y102,x102,y1000,x1000);
A(12,12)=ds_dx_from(y102,x102,y1000,x1000);
A(12,1)=ds_dy_to(y102,x102,y1000,x1000);
A(12,2)=ds_dx_to(y102,x102,y1000,x1000);

A(13,11)=ds_dy_from(y102,x102,y100,x100);
A(13,12)=ds_dx_from(y102,x102,y100,x100);
A(13,7)=ds_dy_to(y102,x102,y100,x100);
```

```
A(13,8)=ds_dx_to(y102,x102,y100,x100);

A(14,11)=ds_dy_from(y102,x102,y2000,x2000);
A(14,12)=ds_dx_from(y102,x102,y2000,x2000);
A(14,3)=ds_dy_to(y102,x102,y2000,x2000);
A(14,4)=ds_dx_to(y102,x102,y2000,x2000);

A(15,13)=ds_dy_from(y103,x103,y3000,x3000);
A(15,14)=ds_dx_from(y103,x103,y3000,x3000);
A(15,5)=ds_dy_to(y103,x103,y3000,x3000);
A(15,6)=ds_dx_to(y103,x103,y3000,x3000);

A(16,13)=ds_dy_from(y103,x103,y2000,x2000);
A(16,14)=ds_dx_from(y103,x103,y2000,x2000);
A(16,3)=ds_dy_to(y103,x103,y2000,x2000);
A(16,4)=ds_dx_to(y103,x103,y2000,x2000);

A(17,13)=ds_dy_from(y103,x103,y101,x101);
A(17,14)=ds_dx_from(y103,x103,y101,x101);
A(17,9)=ds_dy_to(y103,x103,y101,x101);
A(17,10)=ds_dx_to(y103,x103,y101,x101);


A(18,1)=dr_dy_from(y1000,x1000,y100,x100);
A(18,2)=dr_dx_from(y1000,x1000,y100,x100);
A(18,7)=dr_dy_to(y1000,x1000,y100,x100);
A(18,8)=dr_dx_to(y1000,x1000,y100,x100);
A(18,15)=-1;

A(19,1)=dr_dy_from(y1000,x1000,y102,x102);
A(19,2)=dr_dx_from(y1000,x1000,y102,x102);
A(19,11)=dr_dy_to(y1000,x1000,y102,x102);
A(19,12)=dr_dx_to(y1000,x1000,y102,x102);
A(19,15)=-1;

A(20,3)=dr_dy_from(y2000,x2000,y103,x103);
A(20,4)=dr_dx_from(y2000,x2000,y103,x103);
A(20,13)=dr_dy_to(y2000,x2000,y103,x103);
A(20,14)=dr_dx_to(y2000,x2000,y103,x103);
A(20,16)=-1;

A(21,3)=dr_dy_from(y2000,x2000,y101,x101);
A(21,4)=dr_dx_from(y2000,x2000,y101,x101);
A(21,9)=dr_dy_to(y2000,x2000,y101,x101);
A(21,10)=dr_dx_to(y2000,x2000,y101,x101);
A(21,16)=-1;

A(22,5)=dr_dy_from(y3000,x3000,y100,x100);
A(22,6)=dr_dx_from(y3000,x3000,y100,x100);
A(22,7)=dr_dy_to(y3000,x3000,y100,x100);
A(22,8)=dr_dx_to(y3000,x3000,y100,x100);
A(22,17)=-1;

A(23,5)=dr_dy_from(y3000,x3000,y103,x103);
A(23,6)=dr_dx_from(y3000,x3000,y103,x103);
A(23,13)=dr_dy_to(y3000,x3000,y103,x103);
A(23,14)=dr_dx_to(y3000,x3000,y103,x103);
A(23,17)=-1;
```

```
A(24,7)=dr_dy_from(y100,x100,y1000,x1000);
A(24,8)=dr_dx_from(y100,x100,y1000,x1000);
A(24,1)=dr_dy_to(y100,x100,y1000,x1000);
A(24,2)=dr_dx_to(y100,x100,y1000,x1000);
A(24,18)=-1;


A(25,7)=dr_dy_from(y100,x100,y2000,x2000);
A(25,8)=dr_dx_from(y100,x100,y2000,x2000);
A(25,3)=dr_dy_to(y100,x100,y2000,x2000);
A(25,4)=dr_dx_to(y100,x100,y2000,x2000);
A(25,18)=-1;

A(26,7)=dr_dy_from(y100,x100,y3000,x3000);
A(26,8)=dr_dx_from(y100,x100,y3000,x3000);
A(26,5)=dr_dy_to(y100,x100,y3000,x3000);
A(26,6)=dr_dx_to(y100,x100,y3000,x3000);
A(26,18)=-1;

A(27,9)=dr_dy_from(y101,x101,y103,x103);
A(27,10)=dr_dx_from(y101,x101,y103,x103);
A(27,13)=dr_dy_to(y101,x101,y103,x103);
A(27,14)=dr_dx_to(y101,x101,y103,x103);
A(27,19)=-1;

A(28,9)=dr_dy_from(y101,x101,y2000,x2000);
A(28,10)=dr_dx_from(y101,x101,y2000,x2000);
A(28,3)=dr_dy_to(y101,x101,y2000,x2000);
A(28,4)=dr_dx_to(y101,x101,y2000,x2000);
A(28,19)=-1;

A(29,11)=dr_dy_from(y102,x102,y1000,x1000);
A(29,12)=dr_dx_from(y102,x102,y1000,x1000);
A(29,1)=dr_dy_to(y102,x102,y1000,x1000);
A(29,2)=dr_dx_to(y102,x102,y1000,x1000);
A(29,20)=-1;

A(30,11)=dr_dy_from(y102,x102,y100,x100);
A(30,12)=dr_dx_from(y102,x102,y100,x100);
A(30,7)=dr_dy_to(y102,x102,y100,x100);
A(30,8)=dr_dx_to(y102,x102,y100,x100);
A(30,20)=-1;

A(31,11)=dr_dy_from(y102,x102,y2000,x2000);
A(31,12)=dr_dx_from(y102,x102,y2000,x2000);
A(31,3)=dr_dy_to(y102,x102,y2000,x2000);
A(31,4)=dr_dx_to(y102,x102,y2000,x2000);
A(31,20)=-1;

A(32,13)=dr_dy_from(y103,x103,y3000,x3000);
A(32,14)=dr_dx_from(y103,x103,y3000,x3000);
A(32,5)=dr_dy_to(y103,x103,y3000,x3000);
A(32,6)=dr_dx_to(y103,x103,y3000,x3000);
A(32,21)=-1;

A(33,13)=dr_dy_from(y103,x103,y2000,x2000);
A(33,14)=dr_dx_from(y103,x103,y2000,x2000);
A(33,3)=dr_dy_to(y103,x103,y2000,x2000);
A(33,4)=dr_dx_to(y103,x103,y2000,x2000);
```

```
A(33,21)=-1;

A(34,13)=dr_dy_from(y103,x103,y101,x101);
A(34,14)=dr_dx_from(y103,x103,y101,x101);
A(34,9)=dr_dy_to(y103,x103,y101,x101);
A(34,10)=dr_dx_to(y103,x103,y101,x101);
A(34,21)=-1;

A(35,1)=1;
A(36,2)=1;
A(37,3)=1;
A(38,4)=1;

%Normal matrix
N = A'*P*A;


%Vector of right hand side of normal equations
 n = A'*P*l;

%Inversion of normal matrix / Cofactor matrix of the unknowns
 Q_xx = inv(N);


%Solution of normal equation
 x_hat = Q_xx*n;

%Adjusted unknowns
 X_hat = X_0+x_hat;

%Update
X_0 = X_hat;

y1000 = X_0(1);
x1000 = X_0(2);
y2000 = X_0(3);
x2000 = X_0(4);
y3000 = X_0(5);
x3000 = X_0(6);
y100 = X_0(7);
x100 = X_0(8);
y101 = X_0(9);
x101 = X_0(10);
y102 = X_0(11);
x102 = X_0(12);
y103 = X_0(13);
x103 = X_0(14);
w1000 = X_0(15);
w2000 = X_0(16);
w3000 = X_0(17);
w100 = X_0(18);
w101 = X_0(19);
w102 = X_0(20);
w103 = X_0(21);

%Check 1
 max_x_hat = max(abs(x_hat));

%Update number of iterations
```

```
    iteration=iteration+1;

end

%Convert to [gon] and check the quadrants
gon = X_0(15:21,1)*200/pi;     %convert w to gon

gon = gon+400;

%gon(1) = gon(1)+400;          %check w1-w9 in X_0 >> check the quadrants
manually >> add +400 for the minus value
%gon(2) = gon(2)+400;
%gon(3) = gon(3)+400;
%gon(4) = gon(4)+400;




%Vector of residuals
v = A*x_hat-l;
v_p=v(1:17,1);
v_gon = v(18:34,1)*200/pi;             %Convert to [gon]

%Objective function
vTPv = v'*P*v;

%Vector of adjusted observations
L_hat = L+v;
L_hat_gon = L_hat(18:34)*200/pi;      %Convert to [gon]


%Final check for the linearization

    %Vector of reduced distances
    Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
    Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
    Phi_X_hat(3)=dis(y2000,x2000,y101,x101);
    Phi_X_hat(4)=dis(y3000,x3000,y100,x100);
    Phi_X_hat(5)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(6)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(7)=dis(y100,x100,y102,x102);
    Phi_X_hat(8)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(9)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(10)=dis(y101,x101,y103,x103);
    Phi_X_hat(11)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(12)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(13)=dis(y102,x102,y100,x100);
    Phi_X_hat(14)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(15)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(16)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(18)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(19)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(20)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(21)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(22)=direction(y3000,x3000,y100,x100,w3000);
```

```
    Phi_X_hat(23)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(24)=direction(y100,x100,y1000,x1000,w100);
    Phi_X_hat(25)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(26)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(27)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(28)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(29)=direction(y102,x102,y1000,x1000,w102);
    Phi_X_hat(30)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(31)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(32)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(33)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(34)=direction(y103,x103,y101,x101,w103);

    % Control points
    Phi_X_hat(35) = y1000;
    Phi_X_hat(36) = x1000;
    Phi_X_hat(37) = y2000;
    Phi_X_hat(38) = x2000;


% Final Check
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end




%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_gon = s_X(15:21,1)*200/pi;        %Convert to [gon]

%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';

%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;

%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_d=s_L_hat(1:17,1);
s_L_hat_gon = s_L_hat(18:34)*200/pi;    %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
```

```
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_d=s_v(1:17,1);
s_v_gon = s_v(18:34,1)*200/pi;        %Convert to [gon]



%---------------------------------------------------------------------------
%  Global test
%---------------------------------------------------------------------------

%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
tx2l = chi2inv(0.025,r);  %a/2

if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%---------------------------------------------------------------------------
% Internal and external reliability parameters
%---------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_d=GF(1:17,1);
GF_gon = GF(18:34,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end

GRZW_d=GRZW(1:17,1);
GRZW_gon = GRZW(18:34,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(17,1);
```

```
r_w(1)  = P_diag(18,1)/(P_diag(18,1)+P_diag(19,1));
r_w(2)  = P_diag(19,1)/(P_diag(18,1)+P_diag(19,1));
r_w(3)  = P_diag(20,1)/(P_diag(20,1)+P_diag(21,1));
r_w(4)  = P_diag(21,1)/(P_diag(20,1)+P_diag(21,1));
r_w(5)  = P_diag(22,1)/(P_diag(22,1)+P_diag(23,1));
r_w(6)  = P_diag(23,1)/(P_diag(22,1)+P_diag(23,1));
r_w(7)  = P_diag(24,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(8)  = P_diag(25,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(9)  = P_diag(26,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(10)  = P_diag(27,1)/(P_diag(27,1)+P_diag(28,1));
r_w(11)  = P_diag(28,1)/(P_diag(27,1)+P_diag(28,1));
r_w(12)  = P_diag(29,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(13)  = P_diag(30,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(14)  = P_diag(31,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(15)  = P_diag(32,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(16)  = P_diag(33,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(17)  = P_diag(34,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));

%dd = dist(:,3);

dd(1)  = distances(1,3);
dd(2)  = distances(2,3);
dd(3)  = distances(15,3);
dd(4)  = distances(3,3);
dd(5)  = distances(4,3);
dd(6)  = distances(5,3);
dd(7)  = distances(6,3);
dd(8)  = distances(8,3);
dd(9)  = distances(9,3);
dd(10)  = distances(10,3);
dd(11)  = distances(11,3);
dd(12)  = distances(12,3);
dd(13)  = distances(13,3);
dd(14)  = distances(14,3);
dd(15)  = distances(15,3);
dd(16)  = distances(16,3);
dd(17)  = distances(17,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(38,1);

for i=1:17
  EGK(i)  = (1-EV(i,1))*GRZW(i,1);
end

for i=18:34
  EGK(i)  = (1-EV(i,1)-r_w(i-17,1))*GRZW(i,1)*dd(i-17);
end

for i=35:38
  EGK(i)  = (1-EV(i,1))*GRZW(i,1);
end

EGK_d=EGK(1:17,1);
EGK_gon = EGK(18:34,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
```

```
EP = ones(38,1);

for i=1:17
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=18:34
  EP(i,1) = (1-EV(i,1)-r_w(i-17,1))*GF(i,1)*dd(i-17);
end

for i=35:38
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

EP_d=EP(1:17,1);
EP_gon = EP(18:34,1)*200/pi;


results.L=[L(1:18,1); directions(:,3)];
results.v=[v(1:18,1); v_gon(:,1)];
results.L=[L_hat(1:18,1); L_hat_gon(:,1)];
results.s=[s_v(1:18,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:18,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:18,1); GF_gon(:,1)];
results.GRZW=[GRZW(1:18,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:18,1); EGK_gon(:,1)];
results.EP=[EP(1:18,1); EP_gon(:,1)];

results= struct2table(results);
writetable(results, 'task2_2.xls');
```

Task 3

```
%-----------------------------------------------------------------------------
%
%   SELECTED SECTIONS OF ADJUSTMENT CALCULATION
%                  HOMEWORK 2
%         Combined Horizontal Network
%
%   Author          : Arghadeep Mazumder
%   Mat. No         : 378554
%   Version         : May 30, 2017
%-----------------------------------------------------------------------------

clc;
clear all;
close all;
format long g;
distances = load('Distances_1.txt');
directions = load('Directions_1.txt');
control_point = load('Control_Points.txt');
new_point = load('New_Points.txt');

%Gauss-Krueger coordinates for control points [m]
y1000 = control_point(1,2);
x1000 = control_point(1,3);
y2000 = control_point(2,2);
x2000 = control_point(2,3);
```

```
%New points [m]
y100 = new_point(1,2);
x100 = new_point(1,3);
y101 = new_point(2,2);
x101 = new_point(2,3);
y102 = new_point(3,2);
x102 = new_point(3,3);
y103 = new_point(4,2);
x103 = new_point(4,3);
y3000 = control_point(3,2);
x3000 = control_point(3,3);


%Initial values for orientation unknowns
w1000 = 0;
w2000 = 0;
w3000 = 0;
w100 = 0;
w101 = 0;
w102 = 0;
w103 = 0;

%Vector of observations
%l_dist = [dist(:,3)];
%l_dir = [dir(:,3)*pi/200];

L = [distances(:,3); directions(:,3)*pi/200];    %Convert to [rad]


%Initial values for unknowns
X_0 = [y3000 x3000 y100 x100 y101 x101 y102 x102 y103 x103 w1000 w2000
w3000 w100 w101 w102 w103]';


%Number of observations
no_n = length(L);

%Number of unknowns
no_u = length(X_0);

%Redundancy
r = no_n - no_u;

%xy = reshape(X_0(1:4),1,4);

%--------------------------------------------------------------------------
%  Stochastic model
%--------------------------------------------------------------------------
%VC Matrix of the observations
s_dist = 0.001;                      %[m]
s_dir = 0.001*pi/200;                %Convert to [rad]
s_xy = 0.01;                         %[m]

s_LL = [s_dist^2*ones(length(distances),1);
s_dir^2*ones(length(directions),1)];
S_LL = diag(s_LL);

%Theoretical standard deviation
```

```
sigma_0 = 1;

%Cofactor matrix of the observations
Q_LL = 1/sigma_0^2*S_LL;

%Weight matrix
P = inv(Q_LL);

%--------------------------------------------------------------------------
%  Adjustment
%--------------------------------------------------------------------------
%break-off conditions
epsilon = 10^-7; %given accuracy in 0.001
delta = 10^-9;
max_x_hat = 10^Inf;

%Number of iterations
iteration = 0;

%Initialising A
A = zeros(no_n,no_u);

%Iteration
while max_x_hat > epsilon

    %Vector of reduced distances
    L_0(1)=dis(y1000,x1000,y100,x100);
    L_0(2)=dis(y1000,x1000,y102,x102);
    L_0(3)=dis(y2000,x2000,y101,x101);
    L_0(4)=dis(y3000,x3000,y100,x100);
    L_0(5)=dis(y3000,x3000,y103,x103);
    L_0(6)=dis(y100,x100,y1000,x1000);
    L_0(7)=dis(y100,x100,y102,x102);
    L_0(8)=dis(y100,x100,y2000,x2000);
    L_0(9)=dis(y100,x100,y3000,x3000);
    L_0(10)=dis(y101,x101,y103,x103);
    L_0(11)=dis(y101,x101,y2000,x2000);
    L_0(12)=dis(y102,x102,y1000,x1000);
    L_0(13)=dis(y102,x102,y100,x100);
    L_0(14)=dis(y102,x102,y2000,x2000);
    L_0(15)=dis(y103,x103,y3000,x3000);
    L_0(16)=dis(y103,x103,y2000,x2000);
    L_0(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    L_0(18)=direction(y1000,x1000,y100,x100,w1000);
    L_0(19)=direction(y1000,x1000,y102,x102,w1000);
    L_0(20)=direction(y2000,x2000,y103,x103,w2000);
    L_0(21)=direction(y2000,x2000,y101,x101,w2000);
    L_0(22)=direction(y3000,x3000,y100,x100,w3000);
    L_0(23)=direction(y3000,x3000,y103,x103,w3000);
    L_0(24)=direction(y100,x100,y1000,x1000,w100);
    L_0(25)=direction(y100,x100,y2000,x2000,w100);
    L_0(26)=direction(y100,x100,y3000,x3000,w100);
    L_0(27)=direction(y101,x101,y103,x103,w101);
    L_0(28)=direction(y101,x101,y2000,x2000,w101);
    L_0(29)=direction(y102,x102,y1000,x1000,w102);
    L_0(30)=direction(y102,x102,y100,x100,w102);
    L_0(31)=direction(y102,x102,y2000,x2000,w102);
    L_0(32)=direction(y103,x103,y3000,x3000,w103);
```

```
L_0(33)=direction(y103,x103,y2000,x2000,w103);
L_0(34)=direction(y103,x103,y101,x101,w103);


l = L-L_0';

%Design matrix

A(1,3)=ds_dy_to(y1000,x1000,y100,x100);
A(1,4)=ds_dx_to(y1000,x1000,y100,x100);


A(2,7)=ds_dy_to(y1000,x1000,y102,x102);
A(2,8)=ds_dx_to(y1000,x1000,y102,x102);


A(3,5)=ds_dy_to(y2000,x2000,y101,x101);
A(3,6)=ds_dx_to(y2000,x2000,y101,x101);

A(4,1)=ds_dy_from(y3000,x3000,y100,x100);
A(4,2)=ds_dx_from(y3000,x3000,y100,x100);
A(4,3)=ds_dy_to(y3000,x3000,y100,x100);
A(4,4)=ds_dx_to(y3000,x3000,y100,x100);

A(5,1)=ds_dy_from(y3000,x3000,y103,x103);
A(5,2)=ds_dx_from(y3000,x3000,y103,x103);
A(5,9)=ds_dy_to(y3000,x3000,y103,x103);
A(5,10)=ds_dx_to(y3000,x3000,y103,x103);

A(6,3)=ds_dy_from(y100,x100,y1000,x1000);
A(6,4)=ds_dx_from(y100,x100,y1000,x1000);


A(7,3)=ds_dy_from(y100,x100,y102,x102);
A(7,4)=ds_dx_from(y100,x100,y102,x102);
A(7,7)=ds_dy_to(y100,x100,y102,x102);
A(7,8)=ds_dx_to(y100,x100,y102,x102);

A(8,3)=ds_dy_from(y100,x100,y2000,x2000);
A(8,4)=ds_dx_from(y100,x100,y2000,x2000);


A(9,3)=ds_dy_from(y100,x100,y3000,x3000);
A(9,4)=ds_dx_from(y100,x100,y3000,x3000);
A(9,1)=ds_dy_to(y100,x100,y3000,x3000);
A(9,2)=ds_dx_to(y100,x100,y3000,x3000);

A(10,5)=ds_dy_from(y101,x101,y103,x103);
A(10,6)=ds_dx_from(y101,x101,y103,x103);
A(10,9)=ds_dy_to(y101,x101,y103,x103);
A(10,10)=ds_dx_to(y101,x101,y103,x103);

A(11,5)=ds_dy_from(y101,x101,y2000,x2000);
A(11,6)=ds_dx_from(y101,x101,y2000,x2000);


A(12,7)=ds_dy_from(y102,x102,y1000,x1000);
```

```
A(12,8)=ds_dx_from(y102,x102,y1000,x1000);


A(13,7)=ds_dy_from(y102,x102,y100,x100);
A(13,8)=ds_dx_from(y102,x102,y100,x100);
A(13,3)=ds_dy_to(y102,x102,y100,x100);
A(13,4)=ds_dx_to(y102,x102,y100,x100);

A(14,7)=ds_dy_from(y102,x102,y2000,x2000);
A(14,8)=ds_dx_from(y102,x102,y2000,x2000);


A(15,9)=ds_dy_from(y103,x103,y3000,x3000);
A(15,10)=ds_dx_from(y103,x103,y3000,x3000);
A(15,1)=ds_dy_to(y103,x103,y3000,x3000);
A(15,2)=ds_dx_to(y103,x103,y3000,x3000);

A(16,9)=ds_dy_from(y103,x103,y2000,x2000);
A(16,10)=ds_dx_from(y103,x103,y2000,x2000);


A(17,9)=ds_dy_from(y103,x103,y101,x101);
A(17,10)=ds_dx_from(y103,x103,y101,x101);
A(17,5)=ds_dy_to(y103,x103,y101,x101);
A(17,6)=ds_dx_to(y103,x103,y101,x101);


A(18,3)=dr_dy_to(y1000,x1000,y100,x100);
A(18,4)=dr_dx_to(y1000,x1000,y100,x100);
A(18,11)=-1;


A(19,7)=dr_dy_to(y1000,x1000,y102,x102);
A(19,8)=dr_dx_to(y1000,x1000,y102,x102);
A(19,11)=-1;


A(20,9)=dr_dy_to(y2000,x2000,y103,x103);
A(20,10)=dr_dx_to(y2000,x2000,y103,x103);
A(20,12)=-1;

A(21,5)=dr_dy_to(y2000,x2000,y101,x101);
A(21,6)=dr_dx_to(y2000,x2000,y101,x101);
A(21,12)=-1;

A(22,1)=dr_dy_from(y3000,x3000,y100,x100);
A(22,2)=dr_dx_from(y3000,x3000,y100,x100);
A(22,3)=dr_dy_to(y3000,x3000,y100,x100);
A(22,4)=dr_dx_to(y3000,x3000,y100,x100);
A(22,13)=-1;

A(23,1)=dr_dy_from(y3000,x3000,y103,x103);
A(23,2)=dr_dx_from(y3000,x3000,y103,x103);
A(23,9)=dr_dy_to(y3000,x3000,y103,x103);
A(23,10)=dr_dx_to(y3000,x3000,y103,x103);
A(23,13)=-1;

A(24,3)=dr_dy_from(y100,x100,y1000,x1000);
```

```
A(24,4)=dr_dx_from(y100,x100,y1000,x1000);
A(24,14)=-1;


A(25,3)=dr_dy_from(y100,x100,y2000,x2000);
A(25,4)=dr_dx_from(y100,x100,y2000,x2000);

A(25,14)=-1;

A(26,3)=dr_dy_from(y100,x100,y3000,x3000);
A(26,4)=dr_dx_from(y100,x100,y3000,x3000);
A(26,1)=dr_dy_to(y100,x100,y3000,x3000);
A(26,2)=dr_dx_to(y100,x100,y3000,x3000);
A(26,14)=-1;

A(27,5)=dr_dy_from(y101,x101,y103,x103);
A(27,6)=dr_dx_from(y101,x101,y103,x103);
A(27,9)=dr_dy_to(y101,x101,y103,x103);
A(27,10)=dr_dx_to(y101,x101,y103,x103);
A(27,15)=-1;

A(28,5)=dr_dy_from(y101,x101,y2000,x2000);
A(28,6)=dr_dx_from(y101,x101,y2000,x2000);
A(28,15)=-1;

A(29,7)=dr_dy_from(y102,x102,y1000,x1000);
A(29,8)=dr_dx_from(y102,x102,y1000,x1000);

A(29,16)=-1;

A(30,7)=dr_dy_from(y102,x102,y100,x100);
A(30,8)=dr_dx_from(y102,x102,y100,x100);
A(30,3)=dr_dy_to(y102,x102,y100,x100);
A(30,4)=dr_dx_to(y102,x102,y100,x100);
A(30,16)=-1;

A(31,7)=dr_dy_from(y102,x102,y2000,x2000);
A(31,8)=dr_dx_from(y102,x102,y2000,x2000);

A(31,16)=-1;

A(32,9)=dr_dy_from(y103,x103,y3000,x3000);
A(32,10)=dr_dx_from(y103,x103,y3000,x3000);
A(32,1)=dr_dy_to(y103,x103,y3000,x3000);
A(32,2)=dr_dx_to(y103,x103,y3000,x3000);
A(32,17)=-1;

A(33,9)=dr_dy_from(y103,x103,y2000,x2000);
A(33,10)=dr_dx_from(y103,x103,y2000,x2000);
A(33,17)=-1;

A(34,9)=dr_dy_from(y103,x103,y101,x101);
A(34,10)=dr_dx_from(y103,x103,y101,x101);
A(34,5)=dr_dy_to(y103,x103,y101,x101);
A(34,6)=dr_dx_to(y103,x103,y101,x101);
A(34,17)=-1;
```

```
    %Normal matrix
    N = A'*P*A;


    %Vector of right hand side of normal equations
     n = A'*P*l;

    %Inversion of normal matrix / Cofactor matrix of the unknowns
     Q_xx = inv(N);


    %Solution of normal equation
     x_hat = Q_xx*n;

    %Adjusted unknowns
     X_hat = X_0+x_hat;

    %Update
    X_0 = X_hat;


    y3000 = X_0(1);
    x3000 = X_0(2);
    y100 = X_0(3);
    x100 = X_0(4);
    y101 = X_0(5);
    x101 = X_0(6);
    y102 = X_0(7);
    x102 = X_0(8);
    y103 = X_0(9);
    x103 = X_0(10);
    w1000 = X_0(11);
    w2000 = X_0(12);
    w3000 = X_0(13);
    w100 = X_0(14);
    w101 = X_0(15);
    w102 = X_0(16);
    w103 = X_0(17);

    %Check 1
     max_x_hat = max(abs(x_hat));

    %Update number of iterations
    iteration=iteration+1;

end

%Convert to [gon] and check the quadrants
gon = X_0(11:17,1)*200/pi;     %convert w to gon

gon = gon+400;

%gon(1) = gon(1)+400;          %check w1-w9 in X_0 >> check the quadrants
manually >> add +400 for the minus value
%gon(2) = gon(2)+400;
%gon(3) = gon(3)+400;
%gon(4) = gon(4)+400;
```

```
%Vector of residuals
v = A*x_hat-l;
v_gon = v(18:34,1)*200/pi;             %Convert to [gon]

%Objective function
vTPv = v'*P*v;

%Vector of adjusted observations
L_hat = L+v;
L_hat_d=L_hat(1:17,1);
L_hat_gon = L_hat(18:34)*200/pi;      %Convert to [gon]


%Final check for the linearization

    %Vector of reduced distances
    Phi_X_hat(1)=dis(y1000,x1000,y100,x100);
    Phi_X_hat(2)=dis(y1000,x1000,y102,x102);
    Phi_X_hat(3)=dis(y2000,x2000,y101,x101);
    Phi_X_hat(4)=dis(y3000,x3000,y100,x100);
    Phi_X_hat(5)=dis(y3000,x3000,y103,x103);
    Phi_X_hat(6)=dis(y100,x100,y1000,x1000);
    Phi_X_hat(7)=dis(y100,x100,y102,x102);
    Phi_X_hat(8)=dis(y100,x100,y2000,x2000);
    Phi_X_hat(9)=dis(y100,x100,y3000,x3000);
    Phi_X_hat(10)=dis(y101,x101,y103,x103);
    Phi_X_hat(11)=dis(y101,x101,y2000,x2000);
    Phi_X_hat(12)=dis(y102,x102,y1000,x1000);
    Phi_X_hat(13)=dis(y102,x102,y100,x100);
    Phi_X_hat(14)=dis(y102,x102,y2000,x2000);
    Phi_X_hat(15)=dis(y103,x103,y3000,x3000);
    Phi_X_hat(16)=dis(y103,x103,y2000,x2000);
    Phi_X_hat(17)=dis(y103,x103,y101,x101);

    %Vector of reduced directions
    Phi_X_hat(18)=direction(y1000,x1000,y100,x100,w1000);
    Phi_X_hat(19)=direction(y1000,x1000,y102,x102,w1000);
    Phi_X_hat(20)=direction(y2000,x2000,y103,x103,w2000);
    Phi_X_hat(21)=direction(y2000,x2000,y101,x101,w2000);
    Phi_X_hat(22)=direction(y3000,x3000,y100,x100,w3000);
    Phi_X_hat(23)=direction(y3000,x3000,y103,x103,w3000);
    Phi_X_hat(24)=direction(y100,x100,y1000,x1000,w100);
    Phi_X_hat(25)=direction(y100,x100,y2000,x2000,w100);
    Phi_X_hat(26)=direction(y100,x100,y3000,x3000,w100);
    Phi_X_hat(27)=direction(y101,x101,y103,x103,w101);
    Phi_X_hat(28)=direction(y101,x101,y2000,x2000,w101);
    Phi_X_hat(29)=direction(y102,x102,y1000,x1000,w102);
    Phi_X_hat(30)=direction(y102,x102,y100,x100,w102);
    Phi_X_hat(31)=direction(y102,x102,y2000,x2000,w102);
    Phi_X_hat(32)=direction(y103,x103,y3000,x3000,w103);
    Phi_X_hat(33)=direction(y103,x103,y2000,x2000,w103);
    Phi_X_hat(34)=direction(y103,x103,y101,x101,w103);



% Final Check
```

```
Finalcheck = max(abs(L_hat-Phi_X_hat'));

if Finalcheck<=delta
    disp('everything is fine!')
else
    disp('Something is wrong!')
end




%Empirical reference standard deviation
s_0 = sqrt(vTPv/r);

%VC matrix of adjusted unknowns
S_XX_hat = s_0^2*Q_xx;

%Standard deviation of the adjusted unknowns
s_X = sqrt(diag(S_XX_hat));
s_X_d=s_X(1:10,1);
s_X_gon = s_X(11:17,1)*200/pi;        %Convert to [gon]

%Cofactor matrix of adjusted observations
Q_LL_hat = A*Q_xx*A';

%VC matrix of adjusted observations
S_LL_hat = s_0^2*Q_LL_hat;

%Standard deviation of the adjusted observations
s_L_hat = sqrt(diag(S_LL_hat));
s_L_hat_d =s_L_hat(1:17,1);
s_L_hat_gon = s_L_hat(18:34)*200/pi;    %Convert to [gon]

%Cofactor matrix of the residuals
Q_vv = Q_LL-Q_LL_hat;

%VC matrix of residuals
S_vv = s_0^2*Q_vv;

%Standard deviation of the residuals
s_v = sqrt(diag(S_vv));
s_v_d=s_v(1:17,1);
s_v_gon = s_v(18:34,1)*200/pi;        %Convert to [gon]



%--------------------------------------------------------------------------
%  Global test
%--------------------------------------------------------------------------


%% Global test
Tx2 = (r*s_0^2)/sigma_0^2;

%for S = 95%
tx2u = chi2inv(0.975,r);  %1-a/2
tx2l = chi2inv(0.025,r);  %a/2
```

```
if tx2l<Tx2 && Tx2<tx2u
  disp('Fails to reject the H0')
else
  disp('Rejects the H0')
end


%-------------------------------------------------------------------------
% Internal and external reliability parameters
%-------------------------------------------------------------------------
% Parameters for internal
% Redundancy numbers
EV = diag(Q_vv*P);  %how much error has been transfered to residuals
EV_100 = EV*100;

% Standardised residuals
sigma_v = sigma_0^2*sqrt(diag(Q_vv));  %theoritical std dev that has no
blunders
NV = abs(v)./sigma_v;

% Potential magnitude of a blunder
GF = -v./(diag(Q_vv*P));
GF_d=GF(1:17,1);
GF_gon = GF(18:34,1)*200/pi;  %only the obs. directions

% Lower boundary value for blunders
GRZW = ones(no_n,1);

for i = 1:no_n
  GRZW(i,1) = sigma_0*4.13/(sqrt(EV(i,1)*P(i,i)));
end
GRZW_d=GRZW(1:17,1);
GRZW_gon = GRZW(18:34,1)*200/pi;

%Parameters for external reliability
P_diag = diag(P);
r_w = ones(17,1);

r_w(1)  = P_diag(18,1)/(P_diag(18,1)+P_diag(19,1));
r_w(2)  = P_diag(19,1)/(P_diag(18,1)+P_diag(19,1));
r_w(3)  = P_diag(20,1)/(P_diag(20,1)+P_diag(21,1));
r_w(4)  = P_diag(21,1)/(P_diag(20,1)+P_diag(21,1));
r_w(5)  = P_diag(22,1)/(P_diag(22,1)+P_diag(23,1));
r_w(6)  = P_diag(23,1)/(P_diag(22,1)+P_diag(23,1));
r_w(7)  = P_diag(24,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(8)  = P_diag(25,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(9)  = P_diag(26,1)/(P_diag(24,1)+P_diag(25,1)+P_diag(26,1));
r_w(10) = P_diag(27,1)/(P_diag(27,1)+P_diag(28,1));
r_w(11) = P_diag(28,1)/(P_diag(27,1)+P_diag(28,1));
r_w(12) = P_diag(29,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(13) = P_diag(30,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(14) = P_diag(31,1)/(P_diag(29,1)+P_diag(30,1)+P_diag(31,1));
r_w(15) = P_diag(32,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(16) = P_diag(33,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));
r_w(17) = P_diag(34,1)/(P_diag(32,1)+P_diag(33,1)+P_diag(34,1));

%dd = dist(:,3);
```

```
dd(1) = distances(1,3);
dd(2) = distances(2,3);
dd(3) = distances(15,3);
dd(4) = distances(3,3);
dd(5) = distances(4,3);
dd(6) = distances(5,3);
dd(7) = distances(6,3);
dd(8) = distances(8,3);
dd(9) = distances(9,3);
dd(10) = distances(10,3);
dd(11) = distances(11,3);
dd(12) = distances(12,3);
dd(13) = distances(13,3);
dd(14) = distances(14,3);
dd(15) = distances(15,3);
dd(16) = distances(16,3);
dd(17) = distances(17,3);

%Impact of the boundary value on the coordinates of the corresponding
%points
EGK = ones(34,1);

for i=1:17
  EGK(i) = (1-EV(i,1))*GRZW(i,1);
end

for i=18:34
  EGK(i) = (1-EV(i,1)-r_w(i-17,1))*GRZW(i,1)*dd(i-17);
end


EGK_d=EGK(1:17,1);
EGK_gon = EGK(18:34,1)*200/pi;

%Impact of a potential blunder on a point corresponding to the measurement
EP = ones(34,1);

for i=1:17
  EP(i,1) = (1-EV(i,1))*GF(i,1);
end

for i=18:34
  EP(i,1) = (1-EV(i,1)-r_w(i-17,1))*GF(i,1)*dd(i-17);
end


EP_d=EP(1:17,1);
EP_gon = EP(18:34,1)*200/pi;



results.L=[L(1:17,1); directions(:,3)];
results.v=[v(1:17,1); v_gon(:,1)];
results.L=[L_hat(1:17,1); L_hat_gon(:,1)];
results.s=[s_v(1:17,1); s_v_gon(:,1)];
results.s=[s_L_hat(1:17,1); s_L_hat_gon(:,1)];
results.GF=[GF(1:17,1); GF_gon(:,1)];
results.GRZW=[GRZW(1:17,1); GRZW_gon(:,1)];
results.EGK=[EGK(1:17,1); EGK_gon(:,1)];
```

```
results.EP=[EP(1:17,1); EP_gon(:,1)];

results= struct2table(results);
writetable(results, 'task3.xls');
```