

Ingenieurgeodäsie II Sommersemester 2022

Übung 12 (Einzelübung)

Regression zur Höhenberechnung

Eingang:

Eingang Wiedervorlage:

Ausgabe 10.05.2022

Abgabe 24.05.2022

Name	Vorname	Matrikelnummer	Unterschrift
Ho	Hsin-Feng	3378849	
Testat	Wiedervorlage	Abgabe bis:	

1. Beschreibung

Das amtliche Höhensystem in Deutschland basiert auf Normalhöhen H_N . Bezugsfläche dieses Höhensystems ist das Quasigeoid. Zur Übertragung von Höhen zwischen bekannten Punkten und Neupunkten wird klassischer Weise das Nivellement benutzt. Die rohen Höhenunterschiede aus dem Nivellement müssen, unter Hinzunahme von Schweremessungen, korrigiert werden, um für die Neupunkte Normalhöhen zu erhalten. Im unwegsamen Gelände oder bei großen Höhenunterschieden wird das Nivellement als Messverfahren sehr zeitaufwendig bis unmöglich. Aus diesem Grund möchte man mehr und mehr das flexiblere Messverfahren PDGPS auch für die Höhenbestimmung einsetzen. Die Bezugsfläche für die mittels GPS bestimmten Höhen ist das WGS84-Ellipsoid, weshalb es sich bei diesen Höhen um ellipsoidische Höhen h handelt.

Die Differenz zwischen Normalhöhen und ellipsoidischen Höhen wird Höhenanomalie ζ genannt. Es gilt: $\zeta = h - H_N$. Eine Bestimmung von Normalhöhen aus PDGPS-Messungen setzt folglich eine genaue Kenntnis der Höhenanomalie voraus. Da das WGS84-Ellipsoid eine Rechenfläche ist, ist seine Gestalt glatt und bekannt. Anders sieht es beim Quasigeoid aus. Die Bezugsfläche für die Normalhöhen ist physikalisch definiert und damit ist ihre Gestalt abhängig vom Schwerfeld der Erde. Somit ist die Höhenanomalie eine

1 a

Höhenanomalie

$$\zeta = h - H_N$$

wobei

- h : ellipsoidische Höhe
- H_N : Normalhöhe

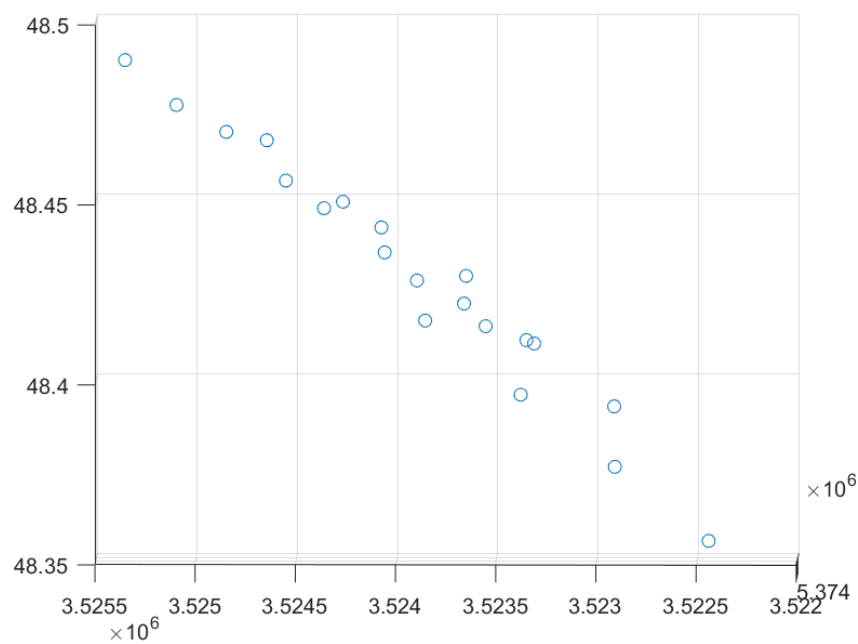
Höhenanomalien von Punkten 1-20:

Pkt.Nr	Höhenanomalie [m]	Pkt.Nr	Höhenanomalie [m]
1	48,3548	11	48,3946
2	48,3928	12	48,4203
3	48,4118	13	48,4420
4	48,4159	14	48,4556
5	48,4290	15	48,4695
6	48,3750	16	48,4148
7	48,4098	17	48,4483
8	48,4290	18	48,4659
9	48,4360	19	48,4762
10	48,4487	20	48,4890

Standardabweichung:

$$\sigma_\zeta = \sqrt{\sigma_h^2 + \sigma_{H_N}^2} = 0,0051 \text{ m}$$

Graphische Darstellung:



2 b

In dieser Aufgabe sind die Höhenanomalien ζ_i mit einem Flächenpolynom von Grad 2 zu approximieren.

$$\zeta_i = a_0 + a_1 y_i + a_2 x_i + a_3 y_i x_i + a_4 y_i^2 + a_5 x_i^2$$

Mit Gauß-Markov-Modell stellt man die folgenden Gleichungen.

$$\underbrace{\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{19} \\ \zeta_{20} \end{bmatrix}}_y = \underbrace{\begin{bmatrix} 1 & y'_1 & x'_1 & y'_1 \cdot x'_1 & y'^2_1 & x'^2_1 \\ 1 & y'_2 & x'_2 & y'_2 \cdot x'_2 & y'^2_2 & x'^2_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & y'_{19} & x'_{19} & y'_{19} \cdot x'_{19} & y'^2_{19} & x'^2_{19} \\ 1 & y'_{20} & x'_{20} & y'_{20} \cdot x'_{20} & y'^2_{20} & x'^2_{20} \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}}_x$$

$$\hat{x} = (A^T A)^{-1} A^T y = \begin{bmatrix} 48,4361 \\ 4,3398 \cdot 10^{-5} \\ -2,5563 \cdot 10^{-6} \\ 8,1539 \cdot 10^{-9} \\ -4,9588 \cdot 10^{-9} \\ -5,7264 \cdot 10^{-9} \end{bmatrix}$$

Schwerpunktkoordinaten:

$$x_s = \frac{1}{20} \sum_{i=1}^{20} x_i = 5375436,408 \text{ m} \quad y_s = \frac{1}{20} \sum_{i=1}^{20} y_i = 3523910,288 \text{ m}$$

3 c

Nun werden die Koeffizienten mit den signifikanten Tests überprüft. Und die nicht signifikant angezeigten Koeffizienten werden iterativ eliminiert. Die Koeffizienten erfolgen eine t-Verteilung.

Standardabweichung von Koeffizienten:

$$\sigma_a = \sqrt{\text{diag}(\Sigma_a)}$$

Testgröße für eine Koeffizient

$$\frac{|a_i - 0|}{\sigma_{a_i}}$$

Quantil ist $t_{97,5,r}^t$ in t-Verteilung. Mit $r = 20 - n$ (n=Anzahl der übrigen Koeffizienten)

Testgröße in jeder Schleife: Koeffizienten in jeder Schleife:

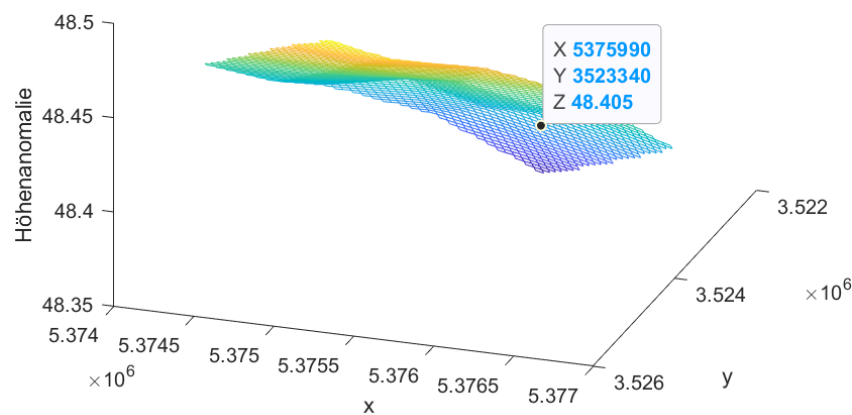
	Quantil	T_{a_0}	T_{a_1}	T_{a_2}	T_{a_3}	T_{a_4}	T_{a_5}
1	3.0688	0.0552	$6,657 \cdot 10^{-11}$	$3,645 \cdot 10^{-12}$	$2,472 \cdot 10^{-17}$	$6,628 \cdot 10^{-18}$	$7,002 \cdot 10^{-18}$
2	2.9467	$3,010 \cdot 10^4$	27,6830	1,8578	3,5185		2,0606
3	2,8131	$3,020 \cdot 10^4$	28,2715		3,7971		2,2270
4	2,6550	$4,105 \cdot 10^4$	28,5474		3,7364		

	a_0	a_1	a_2	a_3	a_4	a_5
1	48,4361	$4,3398 \cdot 10^{-5}$	$-2,5563 \cdot 10^{-6}$	$8,1539 \cdot 10^{-9}$	$-4,9588 \cdot 10^{-9}$	$-5,7264 \cdot 10^{-9}$
2	48,4323	$4,3178 \cdot 10^{-5}$	$-2,7177 \cdot 10^{-6}$	$1,1162 \cdot 10^{-8}$		$-3,4860 \cdot 10^{-9}$
3	48,4325	$4,3608 \cdot 10^{-5}$		$1,1940 \cdot 10^{-8}$		$-3,7536 \cdot 10^{-9}$
4	48,4301	$4,3887 \cdot 10^{-5}$		$1,1745 \cdot 10^{-8}$		

Das neue funktionale Modell:

$$\zeta = a_0 + a_1 y_i + a_3 y_i x_i$$

Graphische Darstellung:



4 d

Wir haben ein funktionales Modell für Höhenanomalie. Wir können die jeweilige Normalhöhen der Punkte 21 bis 30 berechnen.

Pkt.Nr	Höhenanomalie [m]	Normalhöhe [m]
21	48,4340	800,0020
22	48,4345	791,3315
23	48,4253	614,9617
24	48,4258	570,9782
25	48,4316	504,6484
26	48,4429	495,1981
27	48,4465	466,7145
28	48,4425	441,3225
29	48,4337	412,4743
30	48,4226	376,8034

5 e

Durch die Fehlerfortpflanzung kann man außerdem die Standardabweichung der Normalhöhen von Neupunkten ermitteln.

$$\Sigma_{neu} = F \cdot \Sigma_{ll} \cdot F'$$

wobei:

$$F = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & -1 & -y_{21} & -y_{21} \cdot x_{21} \\ 0 & 1 & 0 & \cdots & 0 & 0 & -1 & -y_{22} & -y_{22} \cdot x_{22} \\ 0 & 0 & 1 & \cdots & 0 & 0 & -1 & -y_{23} & -y_{23} \cdot x_{23} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & -1 & -y_{29} & -y_{29} \cdot x_{29} \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 & -y_{30} & -y_{30} \cdot x_{30} \end{bmatrix}$$

Damit ergeben sich die Standardabweichungen:

Pkt.Nr	Standardabweichung [mm]
21	5,1368
22	5,1329
23	5,1541
24	5,1437
25	5,1377
26	5,1553
27	5,2348
28	5,2199
29	5,1554
30	5,1358

6 Matlab-Code

```
1  % Inggeo Uebung 12
2  % Hsin-Feng Ho 3378849
3  clc
4  clear all
5  close all
6  %% Import Data
7  load data.mat
8
9  %% Aufgabe a
10 zeta_1 = data(1:20,4) - data(1:20,3); % Höhenanomalie 1 - 20
11 sigma_HN = 0.001; % Standardabweichung Normalhöhen
12 sigma_e = 0.005; % Standardabweichung Ellipsoid Höhe
13 sigma_zeta = sqrt(sigma_HN^2 + sigma_e^2); % Fehlerfortpflanzung
14 figure
15 scatter3(data(1:20,2), data(1:20,1), zeta_1)
16
17 s_x = mean(data(1:20,2));
18 s_y = mean(data(1:20,1));
19 % hold on
20 % scatter3(s_x, s_y, mean(zeta_1))
21 %% Aufgabe b
22 A = [ones(20,1), data(1:20,1) - s_y, data(1:20,2) - s_x, (data(1:20,1) -
23       s_y).*(data(1:20,2) - s_x), (data(1:20,1) - s_y).^2, (data(1:20,2) -
24       s_x).^2];
25 x_d = inv(A'*A)*A'*zeta_1;
26
27 %% Aufgabe c
28 r = 20 - length(x_d);
29
30 Sigma_a = sigma_zeta^2 \ (A' * A); % test
31
32 sigma_a = sqrt(diag(Sigma_a));
33 T = abs(x_d - 0) ./ sigma_a;
34 Q = tinv(1 - 0.025 / length(x_d), r); % Quantil
35 idx = find(T < Q);
36
37 a_list = cell(6,1);
38 T_list = cell(6,1);
39
40
41
42
43 %%
44 i = 1;
45 id = zeros(6,1) * NaN;
46
47 check = zeros(6,1) * NaN;
48 check_list = 1:6;
49
50 while ~isempty(idx)
51   a_list{i} = x_d;
52   T_list{i} = T;
53   id(i) = find(T == min(T));
54   check(i) = check_list(id(i));
```

```
55  check_list(id(i)) = [];  
56  A(:,id(i)) = [];  
57  x_d = inv(A'*A)*A' * zeta_1;  
  
59  
60  r = 20 - length(x_d);  
61  Sigma_a = sigma_zeta^2 * inv(A' * A);  
  
62  sigma_a = sqrt(diag(Sigma_a));  
63  T = abs(x_d - 0) ./ sigma_a;  
64  Q = tinv(1 - 0.025 / length(x_d), r);  
65  idx = find(T < Q);  
66  i = i + 1;  
67  end  
68  T_list{i} = T;  
69  a_list{i} = x_d;  
70  xq = min(data(1:20,2)):50:max(data(1:20,2));  
71  yq = min(data(1:20,1)):50:max(data(1:20,1));  
72  [xq,yq] = meshgrid(xq,yq);  
73  vq = griddata(data(1:20,2),data(1:20,1),zeta_1,xq,yq);  
74  figure,hold on  
75  mesh(xq,yq,vq)  
76  xlabel("x")  
77  ylabel("y")  
78  zlabel("Hoehenanomalie")  
79  
80  
81  %% Aufgabe d  
82  zeta_2 = x_d(1)+x_d(2).*(data(21:30,1)-s_y)+x_d(3).*(data(21:30,1)-s_y)  
83  .*(data(21:30,2)-s_x);  
84  NH_under = data(21:30,4) - zeta_2; % Normalhoehen 21 - 30  
85  data(21:30,3) = NH_under;  
  
86  
87  %% Aufgabe e  
88  F = [eye(10),-ones(10,1),-(data(21:30,1)-s_y),-(data(21:30,1)-s_y).*(data  
89  (21:30,2)-s_x)];  
90  [~,l] = size(F);  
91  Sigma_big = zeros(l,l);  
92  Sigma_big(1:10,1:10) = 0.005^2 * eye(10);  
93  Sigma_big(11:l,11:l) = Sigma_a;  
  
94  
95  Sigma_nh = F * Sigma_big * F';  
  
96  
97  sigma_nh = sqrt(diag(Sigma_nh))
```