



TECHNISCHE  
UNIVERSITÄT  
WIEN  
Vienna University of Technology

## DIPLOMARBEIT

# Robuste Ausgleichung geodätischer Netzmessungen

## Ein Vergleich von vier Ansätzen

ausgeführt am

Department für Geodäsie und Geoinformation  
Forschungsgruppe Geoinformation  
der Technischen Universität Wien

unter der Anleitung von

Priv.Doz. Dipl.-Ing. Dr.techn. Gerhard Navratil  
und  
Projektass. Dipl.-Math. Franz-Benjamin Mocnik

durch

Helene Kastner BSc.

Unterfeld 12  
4320 Perg

# Danksagung

Ich möchte mich an dieser Stelle bei Hr. DI Dr. Gerhard Navratil für seine freundliche und weitreichende Unterstützung, sowie für sein Engagement bedanken. Hr. Dipl.-Math. Franz-Benjamin Mocnik danke ich neben den wichtigen Beiträgen aus der Sicht eines Mathematikers für seine Zuvorkommenheit und Hilfe.

Auch möchte ich Hr. Ao.Univ.Prof. DI Dr. Peter Filzmoser für seinen wertvollen Hinweis bezüglich LTS danken.

Ebenso bedanke ich mich bei meinen Studienkollegen für den Zusammenhalt während des Studiums und darüber hinaus. Im Speziellen danke ich meiner Kollegin Fr. Sabine Horvath für den konstruktiven Gedankenaustausch in der Entstehungsphase dieser Arbeit.

Nicht zuletzt gebührt meinen Eltern besonderer Dank. Ohne ihre großartige Unterstützung und ihren Rückhalt wäre das Studium in dieser Form nicht möglich gewesen.

# Kurzfassung

Nicht erst seit der automatisierten Aufnahme und Verarbeitung von Messdaten haben robuste Verfahren in vielen naturwissenschaftlichen Fachgebieten einen hohen Stellenwert. Sie zeichnen sich durch ihre Fähigkeit aus, grobe Fehler zu lokalisieren und deren Einfluss zu reduzieren oder zu eliminieren. Vor allem die vorwiegende Nutzung der Methode der kleinsten Quadrate als zentrales geodätisches Auswertekonzept, welches Ausreißer nicht detektieren und den dadurch entstehenden Fehler nicht minimieren kann, erfordert die zusätzliche Verwendung von robusten Schätzern.

Die vorliegende Arbeit widmet sich vier robusten Methoden (L1-Norm, Biber-Schätzer, RAN-SAC und LTS) und erörtert sie unter Zuhilfenahme von terrestrischen Netzmessungen.

Neben den mathematischen Hintergründen wird die Implementierung der Algorithmen in Matlab behandelt. Den Schwerpunkt bildet die Ermittlung der Grenzen und Möglichkeiten jeder der vier robusten Methoden hinsichtlich der Aufdeckbarkeit von groben Fehlern und den Auswirkungen auf das Schätzergebnis. In diesem Zusammenhang werden grobe Fehler simuliert und in das Datenmaterial eingebracht. Des Weiteren wird der Einfluss der sukzessiven Reduktion der Gesamtredundanz untersucht.

Durch die Bewertung aller Verfahren wird eine Hilfestellung für zukünftige Anwendungen im Bereich der robusten Schätzung gegeben.

# Abstract

It's not just since the automatisisation of data gathering and processing that robust estimators play an important role in numerous natural specific fields. The ability of robust estimators is to localise gross errors and to reduce or eliminate the influence on the result. Especially the main use of the least squares adjustment as the major geodetic processing tool, which fails when the data contains outliers, makes it essential to use robust estimation additionally.

This work attends to four robust estimators (L1 norm, Biber estimator, RANSAC and LTS) and adapts them to a terrestrial geodetic network.

Both, the mathematical theory and the implementation of the algorithms in Matlab are discussed. The main focus lies on the determination of limits and possibilities of each method regarding the detection of gross errors and their effect on the results. In this context, outliers are simulated and added to the data. The influence of successive reduction of the redundancy of the system forms an additional research topic.

The work evaluates the pros and cons of the discussed robust estimators and enables assistance for future applications.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Mathematische und physikalische Grundlagen</b>	<b>3</b>
2.1	Einleitung . . . . .	3
2.2	Die Methode der kleinsten Quadrate . . . . .	4
2.2.1	Allgemeinfall der Ausgleichung . . . . .	5
2.2.2	Ausgleichung vermittelnder Beobachtungen . . . . .	7
2.2.3	Hauptprobe . . . . .	8
2.2.4	Datumsfestlegung und Freie Ausgleichung . . . . .	8
2.3	Robuste Schätzung . . . . .	9
2.3.1	Allgemeines . . . . .	10
2.3.2	Die L1-Norm . . . . .	12
2.3.3	Der Biber-Schätzer . . . . .	13
2.3.4	RANSAC . . . . .	17
2.3.5	Least Trimmed Squares . . . . .	19
2.4	Gegenüberstellung . . . . .	22
2.5	Streckenkorrektur . . . . .	24
2.5.1	Meteorologische Reduktion . . . . .	24
2.5.2	Geometrische Reduktion . . . . .	25
<b>3</b>	<b>Testdaten</b>	<b>27</b>
3.1	Richtungsmessungen . . . . .	29
3.1.1	Aufnahme . . . . .	29
3.1.2	Aufbereitung . . . . .	31
3.2	Streckenmessungen . . . . .	32
3.2.1	Aufnahme . . . . .	32
3.2.2	Streckenkorrektur und Aufbereitung . . . . .	33
<b>4</b>	<b>Umsetzung</b>	<b>36</b>
4.1	Simulationsnetz . . . . .	36
4.2	Grenzen der Verfahren . . . . .	37
4.3	Methode der kleinsten Quadrate . . . . .	38
4.4	L1-Norm . . . . .	41
4.5	Biber-Schätzer . . . . .	43
4.6	RANSAC . . . . .	45
4.7	LTS . . . . .	50
<b>5</b>	<b>Analyse</b>	<b>52</b>
5.1	Ausgangsmodell . . . . .	52
5.2	Modell mit einem groben Fehler . . . . .	57
5.3	Modell mit Fehler und reduzierter Redundanz . . . . .	67
5.3.1	Ein grober Fehler . . . . .	68

5.3.2	Zwei grobe Fehler . . . . .	75
5.3.3	Drei grobe Fehler . . . . .	77
5.3.4	Vier grobe Fehler . . . . .	78
5.4	Conclusio . . . . .	79
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>82</b>
	<b>Literaturverzeichnis</b>	<b>85</b>
	<b>Anhang</b>	<b>88</b>
A	Datenfiles . . . . .	88
A.1	naeherung_a.txt . . . . .	88
A.2	richtungen_acorr.txt . . . . .	88
A.3	sGK_acorr.txt . . . . .	89
A.4	strecken_acorr.txt . . . . .	90
B	Streckenkorrektur . . . . .	91
C	Methode der kleinsten Quadrate . . . . .	92
D	L1-Norm . . . . .	99
E	Biber-Schätzer . . . . .	106
F	RANSAC . . . . .	111
F.1	Hauptprogramm . . . . .	111
F.2	Funktion für die Verbesserungen . . . . .	125
G	LTS . . . . .	127
G.1	Hauptprogramm . . . . .	127
G.2	Funktion für die Verbesserungen . . . . .	136
G.3	Funktion für den Ausgleich nach der L2-Norm . . . . .	138

# Kapitel 1

## Einleitung

Im Allgemeinen wird bei geodätischen Aufgaben von einer redundanten Messanordnung ausgegangen. Dies bedeutet, dass mehr Bestimmungsstücke gemessen werden als für eine eindeutige Lösung notwendig sind. Diese Vorgehensweise ist erforderlich, da jede Messung unweigerlich fehlerbehaftet ist und es daher einer durchgreifenden Kontrolle bedarf. Die Aufzeichnung und Auswertung der Messgrößen erfolgt zunehmend automatisiert. Richtungen, Strecken, Koordinaten und andere Parameter werden im Zuge der Vermessung meist ohne Überprüfung auf ihre Plausibilität gespeichert. Vor allem grobe Fehler, die noch während der Aufnahme im Feld aufgedeckt und korrigiert werden könnten, bleiben oft unentdeckt und beeinflussen in der Folge wesentlich die Weiterverarbeitung. Aufgrund der überbestimmten Datenaufnahme wird die Ausgleichsrechnung als zentrales Berechnungsinstrument verwendet. Ihre Umsetzung erfolgt im Speziellen durch die Methode der kleinsten Quadrate. Bedingt durch ihre wahrscheinlichkeitstheoretische Konzeption schlagen sich hier die nicht entdeckten groben Fehler nieder und bewirken falsche Ergebnisse. An dieser Stelle setzen die robusten Verfahren an, welche in der Lage sind, den Einfluss fehlerhafter Messungen zu eliminieren oder zu reduzieren und etwaige Fehler aufzudecken.

Es existiert eine Vielzahl an robusten Methoden, wobei in dieser Arbeit vier davon vorgestellt werden. Es handelt sich dabei um die in der Informatik häufig genutzte L1-Norm, die in der Statistik vorwiegend verwendete Methode LTS, sowie den weit verbreiteten RANSAC-Algorithmus und den speziell für geodätische Anwendungen konzipierten Biber-Schätzer. Diese Verfahren werden hinsichtlich ihrer Fähigkeit, grobe Fehler zu detektieren, untersucht. Zudem erfolgt eine Betrachtung der Abweichungen, welche sich durch den Vergleich mit dem Ergebnis der Methode der kleinsten Quadrate ergeben. Die mathematischen Hintergründe der Methode der kleinsten Quadrate, sowie der robusten Schätzung im Allgemeinen und der vier robusten Verfahren im Einzelnen werden in Kapitel 2 beleuchtet und bilden die Grundlage für die weiteren Inhalte der Arbeit.

Zum besseren Verständnis und für realistische Abschätzungen werden die theoretischen Grundlagen anhand eines praktischen Beispiels umgesetzt. Ausgangspunkt bildet ein terrestrisch gemessenes geodätisches Netz, bei dem davon ausgegangen wird, dass die Messdaten keine groben Fehler enthalten. Der Aufbau des Netzes, die Testdaten, sowie unerlässliche Vorarbeiten werden in Kapitel 3 vorgestellt.

Kapitel 4 widmet sich der Implementierung der Algorithmen in Matlab und zeigt auf, welche Unterschiede sich ergeben, wenn alle fünf Rechenverfahren auf die Ausgangsdaten angewendet werden. Dabei wird einerseits untersucht welche Auswirkungen sich auf die in der Geodäsie üblichen Resultate - die Koordinaten - ergeben und andererseits die Effektivität, mit der die Algorithmen arbeiten, verglichen.

Das Hauptaugenmerk liegt auf der Erforschung der Grenzen und Möglichkeiten der robusten Verfahren und bildet Kapitel 5. Dafür wird ein grober Fehler simuliert und nacheinander an jedes Bestimmungsstück angebracht. Auf diese Art können Unterschiede deutlich gemacht und die Vor- und Nachteile jeder Methode diskutiert werden.

In weiterer Folge wird der Frage nachgegangen, inwieweit die Anzahl der Bestimmungsstücke Einfluss auf die Schätzung der Koordinaten nimmt. Es erfolgt eine schrittweise Redundanzreduktion bei gleichzeitiger Einführung eines groben Fehlers. Dabei wird besonders darauf geachtet, dass die Reduktion der Bestimmungsstücke so erfolgt, dass sie tatsächlich auftretenden Fällen aus der Praxis entspricht. Anhand der Abweichungen von der fehlerfreien Lösung werden die Auswirkungen beurteilt. Mit einer weiteren Erhöhung des Ausreißeranteils wird die Problemstellung erschwert und die Leistungsfähigkeit der Methoden getestet.

Kapitel 6 fasst die wichtigsten Erkenntnisse der Arbeit zusammen und gibt Empfehlungen und Beurteilungen auf Grundlage der analysierten Daten ab.



## Kapitel 2

# Mathematische und physikalische Grundlagen

In diesem Kapitel werden die verschiedenen Fehlerarten definiert, welche grundsätzlich auf Messungen Einfluss nehmen können. Die groben Fehler sind dabei von besonderer Wichtigkeit, da sie den Kern der späteren Untersuchungen bilden. Im Bereich der mathematischen Grundlagen wird nach der Abhandlung der Methode der kleinsten Quadrate und ihrer relevanten Besonderheiten, wie zum Beispiel der Datumsproblematik, auf die Grundzüge der robusten Schätzung eingegangen. Die Vorstellung der vier ausgewählten robusten Verfahren mit ihren Wirkungsweisen und mathematischen Hintergründen bildet den Hauptteil dieses Kapitels. Zu den physikalischen Grundlagen zählen die anzubringenden Streckenkorrekturen, die sich in einen meteorologischen und geometrischen Anteil aufspalten lassen und deren Notwendigkeit erläutert wird.

### 2.1 Einleitung

Die Ausgleichsrechnung beschäftigt sich mit der Lösung überbestimmter Gleichungssysteme. Das bekannteste und meist verwendete Verfahren ist die Methode der kleinsten Quadrate. Die dabei verarbeiteten Daten entstehen aus Messungen und sind deshalb zwangsläufig Einflüssen unterworfen, die zu einer Fehlerbehaftung jeder einzelnen Messung führen. Die Voraussetzung für die Anwendung der Methode der kleinsten Quadrate ist, dass die Daten nur zufällige Fehler enthalten, sowie frei von systematischen Einflüssen sind und keine groben Fehler aufweisen (Navratil, 2011, S.100).

Wieser (2011) gibt einen Überblick über die verschiedenen Fehlerarten:

*Zufällige Fehler* werden durch ungleichsinnig wirkende und unvermeidliche Effekte verursacht. In der Geodäsie sind sie vor allem der Unvollkommenheit der Messinstrumente (z.B. technisch begrenzte Möglichkeiten, thermisches Rauschen) und der menschlichen Sinne (z.B. visuelles Auflösungsvermögen) geschuldet. Auch nicht beeinflussbare Bedingungen im Messraum, wie z.B. atmosphärische Turbulenzen, führen zu zufälligen Abweichungen. Sie treten in etwa gleich oft positiv wie negativ auf und wirken auf jede Messung. Sie können nicht eliminiert, jedoch reduziert werden, indem die Aufnahme mehrmals wiederholt und im Anschluss der Mittelwert gebildet wird.

*Systematische Fehler* resultieren aus gleichsinnig wirkenden, nicht (richtig) modellierten Einflüssen auf das Messinstrument und das Messverfahren. Ihre Auswirkung kann ein Vielfaches der zufälligen Abweichungen betragen. Beispiele hierfür sind instrumentelle Justierfehler, Refraktion bei optischen Zielungen oder Veränderungen elektrischer Signale durch Umwelteinflüsse. Zur Reduktion und Aufdeckung solcher Fehler bietet sich die Prüfung, Justierung und Kalibrierung der Messinstrumente an. Ferner ist das Verständnis des Verfahrens und der Geräte und daraus resultierend, die Wahl des geeigneten Messmittels hilfreich.

*Grobe Fehler* haben verschiedene Ursachen. Sie treten etwa durch Unachtsamkeit des Beobachters (Punktverwechslung, Ablesefehler am Instrument) oder eine Fehlfunktion des Instruments auf. Sie betragen ein Vielfaches der zufälligen Messabweichungen und sind ansonsten nicht von zufälligen Fehlern unterscheidbar. Abhilfe kann wiederum durch sorgfältiges Arbeiten und fundierte Kenntnisse des Instrumentariums und Messverfahrens geschaffen werden. Zudem sind eine Plausibilitätskontrolle der Messwerte und eine redundante Aufnahme von Vorteil, um die Aufdeckbarkeit zu erhöhen. In der Praxis spricht man meistens dann von einem groben Fehler, wenn er deutlich erkennbar ist (z.B.  $> 10\sigma$ ), während in der Theorie oftmals schon das Überschreiten der  $3\sigma$ -Grenze ausreichend ist.

## 2.2 Die Methode der kleinsten Quadrate

Aus historischen Gründen und bedingt durch die mathematisch leichte Umsetzbarkeit, hat sich die Methode der kleinsten Quadrate zur Standardmethode für die Ausgleichsrechnung entwickelt. Mit ihr können alle Messmethoden einfach eingebunden werden, überdies ist die Abschätzung von Zuverlässigkeits- und Präzisionsmaßen noch vor der eigentlichen Messung in Form eines a priori-Ausgleichs möglich. Die Feststellung der tatsächlich erreichten Qualitätsmaße nach der Auswertung (a posteriori) zählt ebenfalls zu den Vorteilen dieser Methode (vgl. Navratil, 2011, S.5).

Die Hauptaufgabe der Methode der kleinsten Quadrate besteht darin, die wahrscheinlichsten Werte für die unbekanntes Größen  $\mathbf{X}$  zu bestimmen. Um dies zu erreichen, müssen auch die Verbesserungen  $v_i$  mit der größten Wahrscheinlichkeit gefunden werden. Diese  $v_i$  beschreiben den Betrag, um den eine Beobachtung verändert (verbessert) werden muss, um den Erwartungswert zu erhalten (Navratil, 2008, S.84). Sie resultieren demnach aus den im vorigen Abschnitt vorgestellten Fehlern. Für die Verwendung der Methode der kleinsten Quadrate ist es notwendig, dass die Verbesserungen normalverteilt und die Messungen nur mit zufälligen Fehlern behaftet sind. Nur dann ist die Gauß'sche Normalverteilung (Dichtefunktion  $f(v_i)$ ) für die Verbesserungen und zufälligen Abweichungen anwendbar und gültig.

$$f(v_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \cdot e^{-\frac{v_i^2}{2\sigma_i^2}} \quad (2.1)$$

Es werden jene Verbesserungen gesucht, für welche die Gesamtwahrscheinlichkeitsdichte  $\Omega$  maximal wird (vgl. Navratil, 2008, S.131).

$$\Omega = \left( \prod_{i=1}^n \frac{1}{\sigma_i \sqrt{2\pi}} \right) \cdot e^K \quad \text{mit } K = \frac{1}{2} \sum_{i=1}^n \frac{v_i^2}{\sigma_i^2} \quad (2.2)$$

Das tritt genau dann ein, wenn  $K$  minimal wird. Mit unterschiedlicher Gewichtung der Verbesserungen lässt sich die zentrale Extremwertaufgabe formulieren als

$$\sum_{i=1}^n p_i v_i^2 \rightarrow \min \quad (2.3)$$

bzw.

$$\mathbf{v}^T \mathbf{P} \mathbf{v} \rightarrow \min. \quad (2.4)$$

Je nach Aufgabenstellung ergeben sich in der Auflösung dieser Problemstellung leichte Unterschiede wodurch mehrere Fälle der Ausgleichung differenziert werden können. Nachfolgend wird zunächst der Allgemeinfeld abgehandelt, bevor auf die Ausgleichung vermittelnder Beobachtungen eingegangen wird, welche für die vorliegende Arbeit von Bedeutung ist. Die Notation und der Aufbau des Kapitels 2.2 werden von Niemeier (2008) und Navratil (2011) übernommen.

### 2.2.1 Allgemeinfeld der Ausgleichung

Ausgangspunkt für die Berechnung bildet die Messung von  $n$  Beobachtungen zur Bestimmung von  $u$  Unbekannten. In diesem Zusammenhang wird der  $(n,1)$ -Beobachtungsvektor  $\mathbf{L}$  eingeführt, der alle Beobachtungen  $L_1, L_2, \dots, L_n$  enthält. Dieser Vektor ist eine Näherung für den wahren Wert  $\tilde{\mathbf{L}}$ , der nicht direkt feststellbar ist, aber durch die Methode der kleinsten Quadrate geschätzt werden kann. Dieser Schätzwert  $\hat{\mathbf{L}}$  ergibt sich aus den tatsächlichen Beobachtungen  $\mathbf{L}$  und den Verbesserungen  $\mathbf{v}$ .

$$\hat{\mathbf{L}} = \mathbf{L} + \mathbf{v} \quad (2.5)$$

Die Unbekannten  $X_1, X_2, \dots, X_n$  werden im  $(u,1)$ -Unbekanntenvektor  $\mathbf{X}$  zusammengefasst, welcher wiederum eine Näherung für den wahren Wert  $\tilde{\mathbf{X}}$  darstellt. Der Schätzwert  $\hat{\mathbf{X}}$  für den wahren Wert  $\tilde{\mathbf{X}}$  berechnet sich aus dem genäherten Parametervektor  $\mathbf{X}_0$  und dem gekürzten Parametervektor  $\mathbf{x}$ . Näherungswerte für die Unbekannten können beispielsweise beschafft werden, indem mit einem Teil der vorhandenen Beobachtungen die Problemstellung nicht überbestimmt gelöst wird. Der gekürzte Parametervektor  $\mathbf{x}$  wird im Zuge der Ausgleichung berechnet.

$$\hat{\mathbf{X}} = \mathbf{X}_0 + \mathbf{x} \quad (2.6)$$

Die ausgeglichenen Größen erfüllen das funktionale Modell  $\varphi$ , welches den funktionalen Zusammenhang zwischen den Beobachtungen und Unbekannten herstellt. Es besteht aus  $r$  Funktionen, die Beobachtungsgleichungen genannt werden. Der Nullvektor  $\mathbf{o}$  in Formel (2.7) deutet an, dass die ausgeglichenen Werte der Beobachtungen und Unbekannten die Beobachtungsgleichungen genau erfüllen.

$$\varphi(\hat{\mathbf{L}}, \hat{\mathbf{X}}) = \mathbf{o} \quad (2.7)$$

Wird versucht das funktionale Modell mit den Beobachtungen und den Näherungswerten für die Unbekannten zu erfüllen, entsteht ein Widerspruch

$$\mathbf{w} = \varphi(\mathbf{L}, \mathbf{X}_0). \quad (2.8)$$

Aus den Überlegungen, welche Werte die Beobachtungen annehmen müssten, damit das Gleichungssystem (2.8) keinen Widerspruch aufweist, resultiert der genäherter Beobachtungsvektor  $\mathbf{L}_0$ . Er wird aus den Näherungswerten für die Unbekannten gewonnen. Die tatsächlichen Beobachtungen in  $\mathbf{L}$  weichen von den genähernten Beobachtungen in  $\mathbf{L}_0$  um den gekürzten Beobachtungsvektor  $\mathbf{l}$  voneinander ab.

$$\mathbf{L} = \mathbf{L}_0 + \mathbf{l} \quad (2.9)$$

Gemäß der Annahme, dass die Elemente des gekürzten Parametervektors  $\mathbf{x}$  bzw. des Verbesserungsvektors  $\mathbf{v}$  relativ klein gegenüber den Elementen des genähernten Parametervektors  $\mathbf{X}_0$  bzw. des Beobachtungsvektors  $\mathbf{L}$  sind und ein stetiger Verlauf gegeben ist, darf das funktionale Modell (2.7) mit Hilfe der Taylorentwicklung linearisiert und nach den Gliedern 1. Ordnung (1.Ableitung) abgebrochen werden. Die partiellen Ableitungen nach den Beobachtungen werden in der Designmatrix  $\mathbf{A}$  und jene nach den Beobachtungen in der Matrix  $\mathbf{B}$  zusammengefasst. Mit diesen Definitionen und unter Berücksichtigung der Gleichungen (2.5), (2.6) und (2.8) lässt sich (2.7) umschreiben auf

$$\mathbf{Ax} + \mathbf{Bv} + \mathbf{w} = \mathbf{o}. \quad (2.10)$$

Für die Lösung der Extremwertaufgabe (2.4) mit der Nebenbedingung (2.10) sind noch die Gewichte  $p_i$  festzulegen. Eine Gewichtung ist deswegen sinnvoll, da weniger genaue Messungen einen kleineren Einfluss auf das Ergebnis nehmen sollen als solche mit einer hohen Genauigkeit. Dazu ist die Kovarianzmatrix  $\Sigma_{\mathbf{LL}}$  zu erstellen, welche auf der Hauptdiagonalen die Varianzen der einzelnen Beobachtungen enthält. Bestehen zwischen den Beobachtungen stochastische Abhängigkeiten, sind die jeweiligen Korrelationen in den entsprechenden Zeilen und Spalten außerhalb der Hauptdiagonale einzutragen.

$$\Sigma_{\mathbf{LL}} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{pmatrix} \quad (2.11)$$

Die Varianz der Gewichtseinheit a priori  $\sigma_0$  legt fest, welcher Beobachtung das Gewicht Eins zugeteilt wird und das Genauigkeitsniveau für alle anderen Beobachtungen bestimmt. Sie erlaubt einen Vergleich zwischen der theoretischen und praktischen Messgenauigkeit und ermöglicht gemeinsam mit der Varianz der Gewichtseinheit a posteriori  $s_0$  eine Überprüfung der über das Genauigkeitsniveau getroffenen Annahmen. Es resultiert die Kofaktormatrix

$$\mathbf{Q}_{LL} = \frac{1}{\sigma_0^2} \boldsymbol{\Sigma}_{LL}. \quad (2.12)$$

Die Gewichtsmatrix  $\mathbf{P}$  sorgt schließlich dafür, dass weniger genaue Beobachtungen (große Varianzen) niedriger gewichtet werden, während genaueren Beobachtungen (kleine Varianzen) ein höheres Gewicht verliehen wird.

$$\mathbf{P} = \mathbf{Q}_{LL}^{-1} \quad (2.13)$$

Für die Lösung der Ausgleichungsaufgabe hat sich Multiplikatormethode von Lagrange bewährt. Dabei werden anstelle der Lagrange'schen Multiplikatoren die in der Geodäsie äquivalenten Korrelatoren  $k_i$  verwendet und in einem Vektor  $\mathbf{k}$  zusammengefasst. Die Anzahl der Korrelatoren richtet sich nach der Anzahl der Funktionen in  $\varphi$ . Nach Bildung der ersten Ableitung der zu minimierenden Funktion

$$F(\mathbf{v}, \mathbf{x}) = \mathbf{v}^T \mathbf{P} \mathbf{v} - 2\mathbf{k}^T (\mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{v} + \mathbf{w}) \quad (2.14)$$

und Nullsetzen ebendieser, was dem Lösungsvorgang einer Extremwertaufgabe entspricht, resultiert nach einigen Umformungen das zu lösende Gleichungssystem

$$\begin{pmatrix} \mathbf{k} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{B} \mathbf{P}^{-1} \mathbf{B}^T & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} -\mathbf{w} \\ \mathbf{o} \end{pmatrix}. \quad (2.15)$$

### 2.2.2 Ausgleichung vermittelnder Beobachtungen

Diese Art der Ausgleichung zeichnet sich dadurch aus, dass das funktionale Modell durch  $r = n$  Gleichungen beschrieben wird, wobei jede Gleichung jeweils nur eine Beobachtung enthält und nach dieser aufgelöst werden kann. Gleichung (2.7) kann demnach umgeformt werden auf

$$\hat{\mathbf{L}} = \varphi(\hat{\mathbf{X}}) \quad \text{bzw.} \quad \varphi(\hat{\mathbf{X}}) - \hat{\mathbf{L}} = \mathbf{o}. \quad (2.16)$$

Dadurch resultiert bei der Bildung der partiellen Ableitungen nach den Beobachtungen die negative Einheitsmatrix

$$\mathbf{B} = -\mathbf{I}. \quad (2.17)$$

Infolgedessen ergeben sich aus (2.10) die Verbesserungsgleichungen

$$\mathbf{v} = \mathbf{A} \mathbf{x} - \mathbf{l}. \quad (2.18)$$

Die Umformung in (2.16) ermöglicht weitere Vereinfachungen, die zum Widerspruchsvektor

$$\mathbf{w} = -\mathbf{l} \quad (2.19)$$

führen und das zu lösende Gleichungssystem (2.15) simplifizieren, sodass dieses lautet:

$$\begin{pmatrix} \mathbf{k} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{P}^{-1} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{1} \\ \mathbf{o} \end{pmatrix}. \quad (2.20)$$

Davon abgeleitet, lässt sich die explizite Lösung für  $\mathbf{x}$  angeben als

$$\mathbf{x} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \mathbf{l}. \quad (2.21)$$

Für die Kontrolle der a priori getätigten Annahmen über das Genauigkeitsniveau, berechnet sich die Gewichtseinheit a posteriori unter Einbeziehung des Freiheitsgrades  $n_f$  aus

$$s_0^2 = \frac{\mathbf{v}^T \mathbf{P} \mathbf{v}}{n_f}. \quad (2.22)$$

Der interessierte Leser sei für eine detaillierte Herleitung auf die eingangs erwähnte Literatur verwiesen.

### 2.2.3 Hauptprobe

Nach dem Ausgleichungsprozess sind die ausgeglichenen Beobachtungen und Unbekannten auf ihre Richtigkeit zu kontrollieren. Nachdem das funktionale Modell (2.7) die Bedingung beim Lösen des Gleichungssystems darstellt, ist es notwendig dessen Einhaltung zu überprüfen. Zu diesem Zweck sind die Größen  $\hat{\mathbf{L}}$  und  $\hat{\mathbf{X}}$  in die Funktionen  $\varphi$  einzusetzen. Das funktionale Modell gilt als erfüllt, wenn sich die Differenzen im Bereich der Rechengenauigkeit bewegen.

Sollten die Differenzen größer als erwartet sein, kann dies mehrere Gründe haben:

1. Die zum Linearisieren benötigten Näherungswerte waren nicht ausreichend genau, sodass die Taylorreihe zu früh abgebrochen wurde.
2. Das funktionale Modell ist fehlerhaft.
3. Es ist ein Rechenfehler aufgetreten.

Bei Zutreffen des ersten Falls ist die Lösung iterativ zu berechnen. Die Ergebnisse der ersten Ausgleichung dienen als Näherungswerte für die nächste Iteration. Es wird sowohl eine neue Designmatrix, als auch ein neuer gekürzter Beobachtungsvektor berechnet. Die Iteration wird solange durchgeführt, bis ein Abbruchkriterium erfüllt ist. Zum einen kann dies das ausreichende Erfüllen der Hauptprobe sein, zum anderen das zu Null Werden des gekürzten Parametervektors.

### 2.2.4 Datumsfestlegung und Freie Ausgleichung

Das geodätische Datum umfasst die Lage, die Orientierung und den Maßstab eines Vermessungsnetzes. Wenn keine absoluten Werte (Koordinaten), sondern nur relative Größen (Richtungen, Strecken) gemessen werden, fehlt es an der eindeutigen Festlegung der Datumparameter. Als

Folge besitzt die Designmatrix  $\mathbf{A}$  einen Rangdefekt und ist singulär, das Gleichungssystem (2.20) kann nicht gelöst werden.

Zur Abhilfe können unter anderem zusätzliche, den Rangdefekt kompensierende, Bedingungen eingeführt werden. Diese Vorgehensweise wird bei der freien Ausgleichung genutzt. Der Datumsdefekt  $d$  wird durch die gleiche Anzahl an Bedingungen behoben. Eine weitere Eigenschaft dieser Variante ist, dass auch die Datumspunkte an der Ausgleichung teilnehmen und für sie Präzisionsmaße bestimmt werden können. Tragen alle Netzkpunkte zur Datumsbestimmung bei, spricht man von der Gesamtspurminimierung, andernfalls von der Teilspurminimierung (vgl. Navratil, 2011, S.40f.).

Ein Lagenetz kann beispielsweise nur einen Rang- bzw. Datumsdefekt von maximal vier aufweisen, da es bezüglich der Translation in x- und y-Richtung, der Rotation um die z-Achse und dem Maßstabsfaktor unbestimmt sein kann. Die entsprechenden Bedingungen ergeben sich aus der Forderung, dass „die Summe der Koordinatenzuschläge zu ihren Näherungskoordinaten Null wird; mit anderen Worten: der Schwerpunkt aus den Näherungskoordinaten entspricht dem Schwerpunkt der ausgeglichenen Koordinaten“ (Kraus, 1996, S.101). Dabei bezeichnen  $dx_i$  und  $dy_i$  die differentiellen Zuschläge zu den Näherungskoordinaten  $X_{0i}$ ,  $Y_{0i}$  (Niemeier, 2008, S.234).

$$\begin{aligned}
 \text{Translation in x:} & \quad \sum dx_i = 0 \\
 \text{Translation in y:} & \quad \sum dy_i = 0 \\
 \text{Rotation um z:} & \quad \sum (Y_{0i} dx_i - X_{0i} dy_i) = 0 \\
 \text{Maßstab:} & \quad \sum (X_{0i} dx_i + Y_{0i} dy_i) = 0.
 \end{aligned} \tag{2.23}$$

Durch Ableitung der Gleichungen (2.23) nach den differentiellen Zuschlägen ergibt sich bei Einführung der Parameter in der Reihenfolge y, x die Ränderungsmatrix

$$\mathbf{G}^T = \begin{pmatrix} 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & \dots & 1 & 0 \\ -X_{01} & Y_{01} & \dots & -X_{0m} & Y_{0m} \\ Y_{01} & X_{01} & \dots & Y_{0m} & X_{0m} \end{pmatrix}. \tag{2.24}$$

Das Gleichungssystem (2.20) ändert sich dementsprechend auf

$$\begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{A} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{k} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{l} \\ \mathbf{0} \end{pmatrix}. \tag{2.25}$$

## 2.3 Robuste Schätzung

Für das Verständnis der einzelnen Verfahren ist es notwendig, die Grundzüge der robusten Schätzung zu verstehen. Aus diesem Grund wird im Folgenden ein Einblick in die Thematik gegeben.

### 2.3.1 Allgemeines

Die Definition der robusten Schätzung beinhaltet die weitgehende Unbeeinflussbarkeit der Ergebnisse durch Ausreißer. In der geodätischen Anwendung sind diese Ausreißer grobe Fehler, durch welche die Normalverteilung der Beobachtungen nicht mehr eingehalten werden kann und die Methode der kleinsten Quadrate versagt. Robuste Verfahren sind darauf spezialisiert, grobe Fehler aufzufinden und deren Einfluss auf das Schätzergebnis zu reduzieren oder eliminieren. Im Allgemeinen kommt es dabei nicht zu einem Ersatz einer Methode durch eine andere, vielmehr ergibt sich eine parallele Anwendung beider Systeme, „um je nach Modellkenntnissen und Qualität der Daten die bestmöglichen Ergebnisse zu erzielen“ (Navratil, 2011, S.103). Zunächst wird die robuste Schätzung zur Grobfehlersuche genutzt, damit im Anschluss die Methode der kleinsten Quadrate auf den bereinigten Datensatz angewendet werden kann.

Bei vielen Aufnahmen, vor allem im Bereich der Katastervermessung, werden die Messungen nur mit geringer Überbestimmung durchgeführt. Hier ist es von besonderer Wichtigkeit, auf ein zuverlässiges Berechnungsverfahren vertrauen zu können.

Robuste Schätzer sollen eine Vielzahl an Forderungen erfüllen (Caspary, 1988):

- Verteilungsrobustheit: Fehlerhafte Annahmen im stochastischen Modell sollen das Ergebnis möglichst wenig beeinflussen.
- Datenrobustheit: Der Schätzer soll auch dann noch zuverlässige Ergebnisse liefern, wenn die Ausgangsdaten grobe Fehler enthalten. Die fehlerbehafteten Daten sollen einen geringeren Einfluss auf das Ergebnis haben.
- Modellrobustheit: Die Ergebnisse sollen hauptsächlich aus modellkonformen Daten resultieren. Grobe Fehler, die dem Modell nicht entsprechen, sollen sich wiederum möglichst wenig auswirken.
- Hohe Trennfähigkeit: Modellkonforme Beobachtungen (kleine Verbesserungen) sollen sich deutlich von modellwidersprechenden Beobachtungen (große Verbesserungen) unterscheiden.
- Optimale Schätzergebnisse: Die Ergebnisse sollen den Ergebnissen der Methode der kleinsten Quadrate entsprechen, wenn die entsprechenden Voraussetzungen dafür erfüllt sind.

Je nach Anwendungsfall sind diesen Forderungen unterschiedliche Prioritäten zuzuordnen, da das gleichzeitige Erfüllen aller Punkte sehr schwierig ist (Wicki, 1998, S.58).

Für das Verständnis der weiteren Arbeit sind folgende Begriffe zu klären:

**Schätzfunktion  $t$ :** Oft resultieren die Parameter nicht direkt aus den Beobachtungen, sondern werden aus Funktionen der Messgrößen bestimmt. Zum Beispiel ergeben sich Koordinaten aus einer Funktion von Richtungen und Strecken. Mithilfe einer solchen Schätzfunktion  $t$  wird aus mehreren  $n$  Zufallsvariablen  $X_{i=1-n}$  der Schätzwert  $T_n$  für einen unbekannt Parameter ermittelt ("geschätzt"). Die Schätzfunktion  $t$  ist eine reelle Funktion der Messungen  $x_1, x_2, \dots, x_n$  (vgl. Niemeier, 2008, S.203).



**Verlustfunktion  $\rho(\mathbf{v}_i)$ :** Diese gibt „die Abweichung einer Schätzfunktion von dem gesuchten optimalen Schätzergebnis“ (Niemeier, 2008, S.207) an. Die Verlustfunktion wird für die Bestimmung der Verbesserungen  $v_i$  eingeführt, da diese die Abweichungen der Beobachtungen von dem gesuchten Schätzergebnis darstellen (vgl. Formel (2.5)).

**Einflussfunktion  $\Psi$ :** Sie beschreibt, wie sich eine infinitesimale Änderung in den Daten auf die Schätzfunktion und somit auf das Schätzergebnis auswirkt. Für robuste Schätzer muss die Einflussfunktion beschränkt sein, sodass ein grober Fehler nur einen begrenzten Einfluss auf das Ergebnis haben kann (Herleitung siehe (ebd., S.204ff.)). Die Einflussfunktion ist die erste Ableitung der Verlustfunktion nach den Verbesserungen (ebd., S.211).

**Bruchpunkt:** Der Anteil der groben Fehler im gesamten Datenmaterial wird in Prozent angegeben. Der Bruchpunkt gibt jenen Prozentsatz an groben Fehlern an, bis zu dem es möglich ist, eine zuverlässige Lösung zu erhalten. „Ein Bruchpunkt von 10% bedeutet etwa, dass bis zu 10% der Messungen grob falsch sein dürfen und das Ergebnis immer noch brauchbar ist. Im Fall von exakt 10% ist das Ergebnis jedoch bereits nicht mehr zuverlässig.“ (Navratil, 2011, S.116).

**Redundanzanteil  $r_i$ :** Nach Navratil (2011, S.76) beschreibt der Redundanzanteil „wie viel Prozent eines Beobachtungsfehlers sich in der zugehörigen Verbesserung zeigt“. Die Summe der Redundanzanteile ergibt die Gesamtredundanz  $r$  des Ausgleichungssystems.

$$\text{tr}(\mathbf{Q}_{\mathbf{v}\mathbf{v}}\mathbf{Q}_{\mathbf{ll}}^{-1}) = \sum r_i = r = n - u \quad (2.26)$$

Die  $r_i$  können Werte zwischen Null und Eins annehmen. Für  $r_i = 0$  ist die zugehörige Beobachtung komplett unkontrolliert und ein Fehler würde sich unbemerkt in seiner vollen Größe in den Unbekannten niederschlagen. Mit  $r_i = 1$  ist die Beobachtung zwar vollständig kontrolliert und ein Fehler aufdeckbar, sie ist aber auch überflüssig und sollte im Zuge einer Netzoptimierung eliminiert werden. Meist wird ein Wert im Bereich von  $0,3 < r_i < 0,7$  angestrebt.

**Minimal detektierbare Ausreißer  $\nabla_0 l_i$ :** Es gilt (ebd., S.77f.): Der Grenzwert der inneren Zuverlässigkeit  $\nabla_0 l_i$  gibt an, wie groß ein Ausreißer mindestens sein muss, um gerade noch aufgedeckt werden zu können. Er berechnet sich aus

$$\nabla_0 l_i = \frac{\sigma_{l_i}}{\sqrt{r_i}} \cdot \delta_0. \quad (2.27)$$

Der Dezentralitätsparameter  $\delta_0$  gibt das Zentrum der Normalverteilung der standardisierten Verbesserungen  $w_i$  an. Bei Vorliegen eines groben Fehlers entspricht  $\delta_0$  jenem Betrag, um den die Wahrscheinlichkeitsverteilung der betroffenen Beobachtung gegenüber jener der fehlerfreien Beobachtungen verschoben ist. Der Parameter wird mit einem frei wählbaren Signifikanzniveau  $\alpha_0$  und der Macht  $\beta_0$  festgelegt. Üblicherweise wird für  $\alpha_0 = 0,1\%$  und  $\beta_0 = 80\%$  angenommen, was in  $\delta_0 = 4,13$  resultiert. Oft wird auch der Begriff minimal detektierbarer Bias (MDB) verwendet.

### 2.3.2 Die L1-Norm

Bevor auf die Lösung der Methode eingegangen wird, folgt zunächst ein Einblick in den theoretischen Hintergrund und den Zusammenhang mit der Methode der kleinsten Quadrate.

#### Ansatz

Die L1-Norm zählt zu den L-Schätzern bzw. den Ls-Norm-Schätzern, die sich allgemein durch folgende Verlustfunktion auszeichnen (Navratil, 2011, S.122f.):

$$\rho(v_i) = |v_i|^s \quad \text{mit} \quad 1 \leq s \leq \infty. \quad (2.28)$$

Im Unterschied zur Methode der kleinsten Quadrate (auch L2-Norm, da  $s = 2$ ), wird hier die gewichtete Summe der absoluten Verbesserungen minimiert. Daher ergibt sich mit  $s = 1$ ,

$$\mathbf{p}^T |\mathbf{v}| = \sum_{i=1}^n p_i |v_i| \rightarrow \min. \quad (2.29)$$

Zur Minimierung wird die erste Ableitung der Verlustfunktion gebildet, es resultiert die robuste bzw. beschränkte Einflussfunktion

$$\Psi(v_i) = \text{sign}(v_i). \quad (2.30)$$

#### Lösung

In der Fachliteratur wird meist der Simplex-Algorithmus, der das Problem mithilfe der linearen Optimierung löst, für die Berechnung der L1-Norm vorgeschlagen. Amiri-Simkooei führt die Umformung des ursprünglichen Ausgleichungsproblems inklusive rangdefizitärer  $\mathbf{A}$ -Matrix in ein System der Form  $Ax = b$  an, welches als Ausgangspunkt für den Simplex-Algorithmus dient (Amiri-Simkooei, 2003).

Eine Alternative bietet die Umsetzung mittels iterativer Gewichtsanzpassung. Dabei wird der Berechnungsablauf der Methode der kleinsten Quadrate verwendet und in jeder Iteration  $k$  die Gewichtsmatrix neu definiert. Die Vorgehensweise lautet wie folgt (Neitzel, 2004, S.57; Schlossmacher, 1973):

1. Ansetzen der Gewichtsmatrix als Einheitsmatrix:  $\mathbf{P}^{k=1} = \mathbf{E}$
2. Berechnung der Verbesserungen  $v_1^k, \dots, v_n^k$  nach der Methode der kleinsten Quadrate
3. Bildung der neuen Gewichte auf Basis der Verbesserungen:  $p_i^{k+1} = 1/|v_i^k|$ , wobei  $p_i^{k+1} = 0$ , wenn  $v_i^k \approx 0$
4. Wiederholung der Schritte zwei und drei bis  $|v_i^{k+1} - v_i^k| < \epsilon$

Auf diese Weise erhalten Beobachtungen mit einer großen Verbesserung, die ein Indikator für einen möglichen groben Fehler ist, ein geringes Gewicht. Dementsprechend werden „fehlerfreie“

Beobachtungen aufgewichtet. Nach Erreichen des Abbruchkriteriums in Schritt vier, können anhand der kleinsten Gewichte  $p_i$  Kandidaten für grobe Fehler gefunden werden.

In Untersuchungen von Neitzel (2004, S.57ff.) hat sich gezeigt, dass ein zu Null Setzen der Gewichte problematisch ist. Aus diesem Grund wird entweder diese Vereinfachung außer Acht gelassen oder der Ansatz von Krarup et al. (1980) verwendet, bei dem die Gewichte folgendermaßen gebildet werden:

$$p = \frac{1}{|v| + c} \quad \text{mit } c = \text{const.} \quad (2.31)$$

In diesem Fall ist mit einer langsameren Konvergenz zu rechnen, da die Gewichte sehr groß werden.

### 2.3.3 Der Biber-Schätzer

In den frühen 90er Jahren entwickelte der schweizer Geodät Fridolin Wicki den Biber-Schätzer, indem er den in der Geodäsie verwendeten Huber-Schätzer verbesserte und ihn für die Ausgleichung großer Netze modifizierte. Seit seiner Veröffentlichung ist dieses robuste Verfahren in nahezu allen Vermessungsbüros der Schweiz im Einsatz. Sein Name leitet sich von seiner Eigenschaft ab, den Einfluss von Ausreißern anhand ihrer standardisierten Verbesserungen zu begrenzen: **bounded influence by standardized residuals**.

Der Biber-Schätzer beruht auf den von Huber entwickelten M-Schätzern und ist eng mit dem Schweppe-Schätzer verwandt. Seine Vorteile liegen in:

- seiner einfachen Struktur,
- seiner zuverlässigen Lokalisierung grober Fehler,
- seinen fast identischen Resultaten zu den Ergebnissen der Methode der kleinsten Quadrate, wenn fehlerfreie Daten vorliegen und
- seiner problemlosen Integration in bestehende Ausgleichsprogramme.

Er erfüllt alle an einen robusten Schätzer gestellten Forderungen (siehe Abschnitt 2.3.1) (vgl. Wicki, 1998, S.3, 83f.).

#### Ansatz

Im Gegensatz zum Huber-Schätzer, bei dem die Residuen (Verbesserungen) unabhängig von ihrer Qualität behandelt werden, bezieht der Biber-Schätzer diese mit ein, indem er die Standardabweichungen der Verbesserungen  $\sigma_{v_i}$  als zusätzliches Gewicht verwendet (Hossainali, 2006, S.76), (Wicki, 1998, S.87).

$$\sum_{i=1}^n \sigma_{v_i} \Psi_c \left( \frac{v_i}{\sigma_{v_i}} \right) a_{ij} = 0 \quad (2.32)$$

Sowohl das Netzdesign, als auch die Gewichte der Beobachtungen werden berücksichtigt (Wicki, 1998, S.87), da sich die  $w_i$  wie folgt berechnen (Kraus, 1996, S.225):

$$w_i = \frac{v_i}{\sigma_{v_i}} \quad (2.33)$$

$$\text{mit } \sigma_{v_i} = \sqrt{(\mathbf{Q}_{\mathbf{v}\mathbf{v}})_{ii}} \sigma_0 = \sqrt{(\mathbf{Q}_{\mathbf{v}\mathbf{v}}\mathbf{P}_{\mathbf{II}})_{ii}(\mathbf{P}_{\mathbf{II}})_{ii}^{-1}} \sigma_0 = \sqrt{(\mathbf{Q}_{\mathbf{v}\mathbf{v}}\mathbf{P}_{\mathbf{II}})_{ii}} \sigma_{l_i} = \sqrt{r_i} \sigma_{l_i} \quad (2.34)$$

$$\text{und } \sigma_{l_i} = \sqrt{(\mathbf{Q}_{\mathbf{II}})_{ii}} \sigma_0. \quad (2.35)$$

Weiters schlägt Wicki (1998, S.87ff.) vor, als Verlustfunktion  $\rho_c$  und Einflussfunktion  $\Psi_c$  die an Huber angelehnten Funktionen zu wählen.

$$\rho_c(w_i) = \begin{cases} \frac{1}{2\sigma_{v_i}} v_i^2 & \text{für } |w_i| < c \\ \frac{c}{\sigma_{v_i}} |v_i| - \frac{1}{2} c^2 & \text{für } |w_i| \geq c \end{cases} \quad (2.36)$$

$$\Psi_c(w_i) = \begin{cases} w_i & \text{für } |w_i| < c \\ \text{sign}(w_i)c & \text{für } |w_i| \geq c \end{cases} \quad (2.37)$$

„Da die  $\sigma_{v_i}$  immer positiv sind, gilt:  $\text{sign}(w_i) = \text{sign}(v_i)$ “ (ebd., S.88).

Der Grenzwert  $c$  entscheidet über die Zuordnung der Beobachtungen zu einem der drei Intervalle in (2.38). In der Praxis hat sich für den Grenzwert ein Wert im Bereich von  $2,5 \leq c \leq 4$  bewährt. „Dies entspricht bei normalverteilten Beobachtungen der zweieinhalb bis vierfachen Standardabweichung der  $w_i$ “ (ebd.). Die Intervalle I und III werden als „robuste Intervalle“ bezeichnet, denn ihnen wird genau dann eine Beobachtung zugewiesen, wenn der Verdacht auf einen groben Fehler besteht. Das mittlere Intervall II wird „Methode der kleinsten Quadrate-Intervall“ genannt, da hier jede zugeteilte Beobachtung so behandelt wird, als wenn nach der Methode der kleinsten Quadrate ausgeglichen würde.

$$\begin{array}{lcl} \text{I} & : & w_i \leq -c \\ \text{II} & : & -c < w_i < c \\ \text{III} & : & w_i \geq c \end{array} \quad (2.38)$$

Die Definition (2.37) kann auch nur mit den Verbesserungen  $v_i$  angeschrieben werden, wie Formel (2.39) zeigt. Die dabei verwendeten Grenzwerte  $k_i$  definieren ebenfalls die drei Intervalle aus (2.38). Sie sind für jede Verbesserung  $v_i$  verschieden und werden vor Anwendung des Biber-Schätzers mit den  $\sigma_{v_i}$  aus der Lösung mit der Methode der kleinsten Quadrate berechnet und für die gesamte Parameterschätzung konstant gehalten. Zwar könnten nach jeder Iteration neue Grenzwerte berechnet werden, dies würde aber der Definition des modifizierten M-Schätzers (2.32) widersprechen, der auf den gegebenen zusätzlichen Gewichten aufgebaut ist (ebd., S.102).

$$\Psi_k(v_i) = \begin{cases} v_i & \text{für } |v_i| < k \\ \text{sign}(v_i)k & \text{für } |v_i| \geq k \end{cases} \quad (2.39)$$

Der Zusammenhang mit (2.37) ist dann gegeben durch

$$\Psi_k(v_i) = \sigma_{v_i} \Psi_c(w_i) \quad (2.40)$$

und

$$k_i = c \sigma_{v_i}. \quad (2.41)$$

### Lösung

Der von Wicki (1998, S.92f.) vorgeschlagene Lösungsansatz wird im Folgenden dargelegt. Die betreffenden Normalgleichungen lauten

$$\sum_{i=1}^n \Psi_{k_i}(v_i) a_{ij} = 0. \quad (2.42)$$

Zu deren Lösungen ist ein iteratives Vorgehen notwendig, bei dem alle Beobachtungen eindeutig einem der drei Intervalle zugeordnet werden müssen. Ein Iterieren ist deswegen essentiell, da ein grober Fehler auch bei einer korrekten Beobachtung eine große Verbesserung hervorrufen kann und eine falsche Intervallzuordnung nach sich zieht. Dieses Maskieren des groben Fehlers bzw. dieser „Verschmierungseffekt“ wird durch folgende Vorgehensweise gelöst:

#### Schritt 1: Ausgleichung nach der Methode der kleinsten Quadrate

Da grundsätzlich angenommen wird, dass nur ein kleiner Teil der Beobachtungen grob fehlerhaft ist, wird eine Ausgleichung nach der Methode der kleinsten Quadrate durchgeführt. Es wird davon ausgegangen, dass alle Beobachtungen dem mittleren Intervall angehören.

#### Schritt 2: Auswahl der Beobachtung mit dem maximalen $w_i$

Oftmals enthält die Beobachtung mit der betragsmäßig größten standardisierten Verbesserung einen groben Fehler. Aus diesem Grund wird nur diese Beobachtung dem entsprechenden robusten Intervall zugeordnet, die restlichen  $|w_i| \geq c$  verbleiben im mittleren Intervall.

#### Schritt 3: Berechnung der neuen Unbekannten und Verbesserungen

Es werden neue Unbekannte  $x_{\text{ROB}}$  und Verbesserungen  $v_{\text{ROB}}$  berechnet, allerdings hat darauf die im zweiten Schritt aussortierte Beobachtung nur einen reduzierten Einfluss.

Der reduzierte Einfluss kann mit einer der drei Möglichkeiten erreicht werden, nämlich durch Änderung

1. des Anteils an den Normalgleichungen ( $\mathbf{A}^T \mathbf{P} \mathbf{A}$ ) oder
2. der entsprechenden Beobachtung  $l_i$  oder
3. des entsprechenden Gewichts  $p_i$

Die Schritte zwei und drei werden so lange wiederholt, bis alle Verbesserungen eindeutig einem Intervall zugeordnet sind.

Bei großen Netzen mit vielen Beobachtungen sind die erste und dritte Möglichkeit sehr aufwendig und rechenintensiv, da sie in jedem Schritt ein Lösen des Gleichungssystems unter Einbeziehung der modifizierten Normalgleichungsmatrix erfordern. Es wird daher im Allgemeinen und in der vorliegenden Arbeit die Modifikation der Beobachtungen bevorzugt.

### Modifikation der Beobachtungen

Anhand von (2.39) wird die Absicht des Biber-Schätzers klar: Eine Beobachtung mit der Verbesserung  $v_i$ , welche den Grenzwert  $k_i$  betragsmäßig übersteigt und somit im Verdacht steht mit einem groben Fehler behaftet zu sein, soll denselben Einfluss auf das Ergebnis ausüben, wie eine Beobachtung mit einer Verbesserung der Größe  $k$  (Wicki, 1998, S.96). Ziel ist es, den Einfluss bzw. die Verbesserungen der Beobachtungen in den robusten Intervallen auf  $\pm k$  zu begrenzen.

Die konkrete Vorgehensweise wird anhand Wicki (1998, S.96ff.) erläutert: Allen voran wird eine Ausgleichung nach der Methode der kleinsten Quadrate durchgeführt, um „Startwerte“ für den Biber-Schätzer zu erhalten. Aus der L2-Lösung können die Grenzwerte  $k_i$  für alle Verbesserungen berechnet werden. Aus allen, den robusten Intervallen zugeordneten, Beobachtungen wird jene mit der größten  $|w_i|$  ausgewählt und so modifiziert, dass ihre Verbesserung bei der erneuten Ausgleichung genau den Wert  $k_i$  annimmt. Die modifizierte Beobachtung  $l_i$  beeinflusst die geschätzten Unbekannten somit in einem geringeren Maße als die ursprüngliche.

Im Wesentlichen wird die Änderung der betroffenen Beobachtung mit dem Differenzvektor  $\Delta \mathbf{d}$  angebracht. Das entsprechende Element wird berechnet aus

$$\Delta d_i = \frac{d_i}{-r_i} \quad (2.43)$$

mit

$$d_i = v_i - \text{sign}(v_i)k_i. \quad (2.44)$$

Mit diesen Definitionen werden neue, robuste Unbekannte und Verbesserungen berechnet, wobei  $t$  den Index für die Iterationen symbolisiert. Der gekürzte robuste Parametervektor  $\mathbf{x}_{\text{ROB}}$  wird immer in Bezug auf die ursprünglichen Zuschläge zu den Unbekannten  $\mathbf{x}$  aus der Ausgleichung nach der Methode der kleinsten Quadrate berechnet.

$$\mathbf{x}_{\text{ROB}}^{(t)} = \mathbf{x} - (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \begin{pmatrix} 0 \\ \vdots \\ \frac{d_i}{-r_i} \\ \vdots \\ 0 \end{pmatrix}, \quad (2.45)$$

$$\mathbf{v}_{\text{ROB}}^{(t)} = \mathbf{A}\mathbf{x}_{\text{ROB}}^{(t)} - \left( \mathbf{1} - \begin{pmatrix} 0 \\ \vdots \\ \frac{d_i}{-r_i} \\ \vdots \\ 0 \end{pmatrix} \right). \quad (2.46)$$

Mit den neuen Verbesserungen und ursprünglichen Grenzwerten ergeben sich neue standardisierte Verbesserungen, deren Zuordnung in die Intervalle untersucht wird. Wird wiederum eine Beobachtung einem der beiden robusten Intervalle zugeteilt, wird der zuvor erstellte Differenzvektor um das neue Element erweitert, siehe Formel (2.47). Dieser Ablauf wird solange wiederholt, bis keine Beobachtung mehr ihren Grenzwert überschreitet.

$$\Delta d_i = \frac{1}{-r_i} \sum_{t=1}^{s-1} d_i^{(t)} \quad \text{mit } (s-1) = \text{Anzahl der Iterationen} \quad (2.47)$$

Die Ergebnisse des Biber-Schätzers sind zum einen der gekürzte Parametervektor  $\mathbf{x}_{\text{ROB}}$ , der an den Näherungsvektor  $\mathbf{X}_0$  angebracht werden muss, um die ausgeglichenen Unbekannten  $\hat{\mathbf{X}}$  zu erhalten und zum anderen der Vektor  $\mathbf{v}_{\text{TOT}}$ , der gemeinsam mit dem Beobachtungsvektor  $\mathbf{L}$  die ausgeglichenen Beobachtungen  $\hat{\mathbf{L}}$  ergibt. Im Vektor  $\mathbf{v}_{\text{TOT}}$  befinden sich die Verbesserungen, die direkt an die gemessenen Beobachtungen anzubringen sind, während im Vektor  $\mathbf{v}_{\text{ROB}}$  für jene Beobachtungen, die den robusten Intervallen zuzuordnen sind, der Grenzwert steht.

Es folgt

$$\mathbf{v}_{\text{TOT}} = \mathbf{A}\mathbf{x}_{\text{ROB}} - \mathbf{1}, \quad (2.48)$$

mit dem Zusammenhang

$$\mathbf{v}_{\text{TOT}} = \mathbf{v}_{\text{ROB}} - \Delta \mathbf{d}. \quad (2.49)$$

### 2.3.4 RANSAC

Einen etwas anderen Ansatz verfolgt die RANSAC-Methode von Fischler und Bolles (1981): Statt das überbestimmte Problem in Einem zu lösen, wird dieses in viele eindeutig lösbare Fälle aufgeteilt. Diese eindeutigen Fälle zeichnen sich durch eine Redundanz von Null aus, es werden jeweils genau soviele Beobachtungen miteinbezogen, wie Unbekannte vorhanden sind. Durch das Vorliegen von redundanten Beobachtungen sind viele Kombinationen möglich, um jeweils dasselbe Set von Unbekannten zu berechnen. Die Beobachtungen werden zufällig ausgewählt, weshalb es auch möglich ist, dass Kombinationen auftreten, die nicht lösbar sind. Vor allem bei geodätischen Netzmessungen mit ihren komplexen Beobachtungsplänen ist nicht jede zufällige Auswahl an Messungen auch in der Lage alle Unbekannten zu bestimmen.

Wurden mehrere Ergebnisse berechnet wurden, ist die „richtige“ Lösung zu ermitteln. Dafür wird untersucht, wieviele Beobachtungen aus der Gesamtheit der Messungen eine jeweilige Lösung unterstützen. Diejenige mit den meisten unterstützenden Beobachtungen wird verwendet, um die endgültige Lösung mit allen unterstützenden Beobachtungen zu berechnen. Da an diesem Punkt

davon ausgegangen wird, dass die unterstützenden Beobachtungen nicht mehr mit groben Fehlern behaftet sind, kann die endgültige Lösung mit der Methode der kleinsten Quadrate geschätzt werden.

Zusammengefasst ergibt sich folgender Ablauf, wobei die Notation aus Fischler und Bolles (1981) übernommen wird:

1. Zufällige Auswahl eines Subsets  $S$ , bestehend aus  $u$  Beobachtungen, aus der Gesamtmenge  $P$  aller vorliegenden Beobachtungen.
2. Berechnung einer eindeutigen Lösung bzw. eines Modells  $M$ .
3. Mithilfe von  $M$ , Festlegung eines *Consensus Set*  $S^*$ , welches nur jene Beobachtungen aus  $P$  beinhaltet, die das Modell unterstützen. Das *Consensus Set* ist mithilfe einer selbst definierten Fehlertoleranz zu finden.
4. Liegt  $S^*$  über einem Schwellenwert  $t$ , wird mit allen unterstützenden Beobachtungen (und der Methode der kleinsten Quadrate) das finale Modell  $M^*$  berechnet. Liegt  $S^*$  unter dem Schwellenwert  $t$ , werden die obigen Schritte wiederholt.

Von der zufälligen Auswahl an Stichproben, die eine möglichst hohe Übereinstimmung aufweisen sollen, leitet sich auch der Name dieser Methode ab: **RAN**dom **SAM**ple **C**onsensus.

Nachfolgend werden jene drei Parameter vorgestellt, die bei dieser Methode wesentlich sind.

### **Fehlertoleranz**

Die Fehlertoleranz entscheidet im Berechnungsprozess darüber, welche Beobachtungen ein Modell  $M$  unterstützen und zur Bildung von  $S^*$  beitragen. Der Grenzwert muss einen groben Fehler anhand seiner Größe identifizieren und von der weiteren Berechnung ausschließen. Wird die Toleranz zu niedrig gewählt, können nicht genügend ausreißerfreie Beobachtungen gefunden werden und das Modell wird trotz seiner Richtigkeit nicht als solches angesehen und eventuell verworfen. Wird die Toleranz zu groß gewählt, ist die Wahrscheinlichkeit, dass ein Ausreißer zu den unterstützenden Beobachtungen gezählt wird, höher. Die Anzahl der modellkonformen Beobachtungen kann dann größer werden, als bei einem Modell, das eigentlich ausreißerfrei ist.

In der Geodäsie wird die Fehlertoleranz meist aus der a priori Standardabweichung abgeleitet, wobei das  $3\sigma$ -Intervall ausreichend ist (Navratil, 2011, S.126). Für andere Anwendungen kann der Grenzwert empirisch bestimmt werden.

### **Anzahl der Iterationen $k$**

Es ist zu klären, wieviele Subsets zusammengestellt werden müssen, bis genügend Informationen vorliegen, um ein endgültiges Modell berechnen zu können. Würden alle möglichen Kombinationen in Betracht gezogen, müssten  $\frac{n!}{u!(n-u)!}$  Subsets gebildet werden. Um diesem enormen Aufwand zu entgehen, werden nur  $k$  Kombinationen berechnet. Die Wahl von  $k$  hängt von folgenden Faktoren ab, nämlich (Schindler, 2011, S.7)



- der Anzahl  $u$  an ausgewählten Beobachtungen pro Subset  $S$
- dem relativen Anteil an ausreißerfreien Beobachtungen:  $w$
- der Wahrscheinlichkeit, dass ein ausgewähltes Subset keinen Ausreißer enthält:  $w^u$
- der Wahrscheinlichkeit, dass bei  $k$  Versuchen mindestens ein Subset gewählt wird, welches frei von groben Fehlern ist:  $z = (1 - w^u)^k$

Aus diesen Definitionen folgt (ebd., S.7)

$$k = \frac{\log(1 - z)}{\log(1 - w^u)}. \quad (2.50)$$

Zur Veranschaulichung und Abschätzung des Aufwands ist in Tabelle 2.1 ein Beispiel für die Wahrscheinlichkeit von  $z = 0,99$  gegeben.

Tabelle 2.1: Beispiel für  $z = 0,99$  und den daraus resultierenden Werten für  $k$  (ebd., S.8)

u \ 1-w	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588

### Schwellenwert $t$

Der Schwellenwert  $t$  kann über ein vorzeitiges Abbrechen der RANSAC-Methode entscheiden. Er definiert die Größe eines Consensus Sets  $S^*$ , die erforderlich ist, um daraus das finale Modell  $M^*$  zu berechnen und das Verfahren zu terminieren. Sollte auch nach  $k$  Iterationen bei gegebenem Schwellenwert kein  $S^* > t$  gebildet werden können, wird entweder das größte  $S^*$  für die finale Lösung herangezogen oder die Berechnung abgebrochen.

Eine mögliche Näherung für  $t$  lautet (Collins, 2007, S.27)

$$t = (1 - e) \cdot n, \quad (2.51)$$

wobei  $e$  der relative Anteil an groben Fehlern und  $n$  die Anzahl aller Beobachtungen ist.

### 2.3.5 Least Trimmed Squares

Die Least Trimmed Squares Methode (LTS) stammt aus dem Gebiet der linearen Regressionsanalyse und ist im Bereich der Statistik eine bewährte und robuste Alternative zur klassischen Parameterschätzung (Čížek und Víšek, 2000, S.1). Sie ist zu jenen Verfahren zu zählen, die

sich durch einen hohen Bruchpunkt auszeichnen und eigens dafür entwickelt wurden (Čížek und Víšek, 2000, S.4). Bis zu 50% der Daten dürfen bei LTS mit groben Fehlern kontaminiert sein um dennoch ein zuverlässiges Ergebnis zu erhalten.

Den Vorteilen wie zum Beispiel dem asymptotisch normalen Verhalten oder dem hohen Bruchpunkt stehen auch Nachteile gegenüber. Erwähnenswert ist die „Hohe Subsample Sensitivität“ (Čížek und Víšek, 2000, S.11), die robuste Schätzer mit hohem Bruchpunkt betrifft. Sie resultiert aus der Tatsache, dass diese Art von robusten Verfahren ein „Kernsubset“ suchen, das einem bestimmten Modell am besten folgt, ohne dafür alle Daten einzubeziehen. Eine (kleine) Änderung in den Beobachtungen kann zu einer großen Veränderung in der Zusammensetzung dieses Kernsubsets führen. Dies kann zum Beispiel passieren wenn den Daten mehrere Verteilungen zugrunde liegen (Untersuchungen hierzu siehe Benáček et al. (1998)). In diesem Fall folgt das Schätzergebnis einer anderen zugrundeliegenden Regressionsfunktion als zuvor.

Für eine genauere Auseinandersetzung mit LTS wird auf Rousseeuw und Leroy (1987) und Rousseeuw (1985) verwiesen.

### Ansatz

Zwischen konkreten Messungen  $x_i$  und  $y_i$  soll ein linearer Zusammenhang in Form von Koeffizienten  $\beta_i$  gefunden werden. Ein Intercept-Term  $\beta_0$  ist dabei optional. „Die Beziehung gilt nicht exakt, sondern ist durch zufällige Störungen  $\epsilon$  überlagert“ (Fahrmeir et al., 2009, S.60). Die Residuen  $\epsilon_i$  beschreiben folglich die Abweichungen der einzelnen Messpunkte von der geschätzten Lösung. Ein (multiples) lineares Regressionsproblem in allgemeiner Form lautet (Fahrmeir et al., 2009, S.60)

$$y_i = \beta_0 + \beta_1 \cdot x_{i1} + \dots + \beta_k \cdot x_{ik} + \epsilon_i \quad \text{mit } i = 1, \dots, n \text{ und } k = 1, \dots, u. \quad (2.52)$$

Durch Überführen in die Matrixschreibweise, wobei der Intercept-Term vernachlässigt wird und anschließender Anwendung der bisher verwendeten Notation, lässt sich Formel (2.52) umformulieren auf (Fahrmeir et al., 2009, S.60-61)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad (2.53)$$

bzw.

$$\mathbf{l} = \mathbf{A}\mathbf{x} + \mathbf{v} \quad \Leftrightarrow \quad \mathbf{v} = \mathbf{A}\mathbf{x} - \mathbf{l}. \quad (2.54)$$

LTS minimiert ebenfalls die Residuen  $v_i$ , dabei werden aber nur  $h$  der insgesamt  $n$  Messungen miteinbezogen. Die Wahl der  $h$  Beobachtungen geschieht auf Basis der geordneten Residuen. Die Grundaufgabe kann demnach definiert werden als (Čížek und Víšek, 2000, S.6)

$$\beta^{LTS} = \min \sum_{i=1}^h v_i^2, \quad (2.55)$$

wobei von den aufsteigend gereihten Residuen

$$|v_1| \leq |v_2| \leq \dots \leq |v_h| \leq |v_{h+1}| \leq \dots \leq |v_n| \quad (2.56)$$

ausgegangen wird. Die dabei verwendete Trimmingkonstante  $h$  muss im Bereich

$$\frac{n}{2} < h \leq n \quad (2.57)$$

liegen und definiert den Bruchpunkt (ebd., S.9): Kleine Werte von  $h$  führen zu einem hohen Bruchpunkt, während große Werte die Effizienz verbessern, da mehr Daten in die Schätzung miteinbezogen werden (vorausgesetzt das Datenmaterial ist nur gering kontaminiert). Der maximale Bruchpunkt kann mit

$$h = \frac{n + u + 1}{2} \quad (2.58)$$

erreicht werden.

Čížek und Víšek (2000, S.10) empfehlen, die Werte für  $h$  zu variieren und das Schätzergebnis dabei zu beobachten. Auf diese Weise kann unter Umständen auf den Anteil an groben Fehlern rückgeschlossen werden.

## Lösung

Der Berechnungsablauf in der vorliegenden Arbeit orientiert sich am Modell von Rousseeuw und van Driessen (2006), das speziell für große Datensätze entworfen wurde und eine schnelle Lösung des Problems ermöglicht. Die Vorgehensweise lautet demnach:

1. Festlegung des Parameters  $h$
2. für Datensätze bis  $n \leq 600$  sind 500 Wiederholungen nachfolgender Schritte (a-f) ausreichend. Für einen größeren Datenumfang wird in der erwähnten Literatur eine dementsprechend rechenzeitparende Alternative vorgeschlagen.
  - (a) Bildung eines initialen Subsets  $H_1$ , indem eine eindeutig bestimmte Lösung berechnet wird
  - (b) Bestimmung der Residuen  $v_n$  aller Beobachtungen zur Initiallösung
  - (c) Ordnen der Residuen gemäß  $|v_{1:n}| \leq |v_{2:n}| \leq \dots \leq |v_{n:n}|$
  - (d) Berechnen der Quadratsumme  $Q_{1-3}$  aus den  $h$  kleinsten Residuen

- (e) Mit den zugehörigen  $h$  Beobachtungen ( $= H_2$ ) die Lösung überbestimmt ermitteln
- (f) Zweimalige Wiederholung der Schritte (a) - (e) bis  $Q_3$

3. Auswählen der zehn kleinsten  $Q_3$

4. Ausgehend von der jeweils letzten Lösung als „Initiallösung“, Ausführen der Schritte (b)-(e) bis das Verfahren konvergiert

5. Die endgültige Lösung entspricht jener mit dem kleinsten konvergierten  $Q$

Die Schritte (a) bis (e) werden als „ $C$ -Schritte“ (nach: concentration) bezeichnet, da die Summe der kleinsten Residuen kleiner ist, als im Schritt zuvor. Der Beweis hierfür liegt in Formel (2.4), da die Methode der kleinsten Quadrate die Residuen minimiert, sodass gilt

$$Q_{i+1} \leq Q_i . \tag{2.59}$$

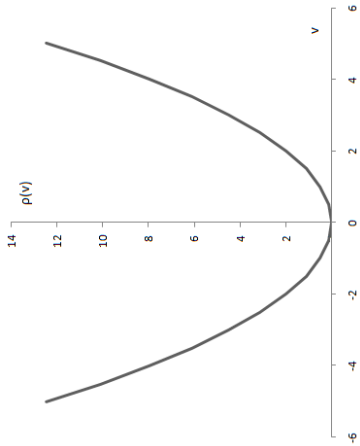
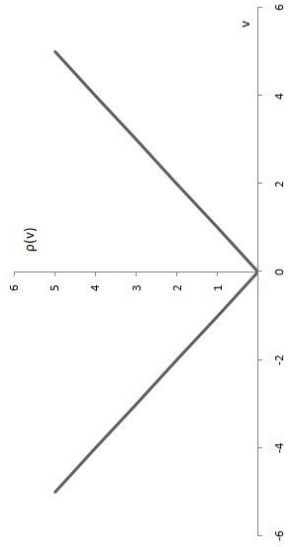
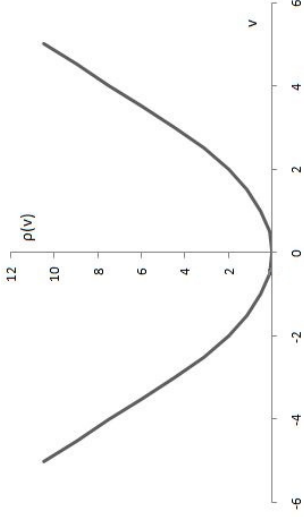
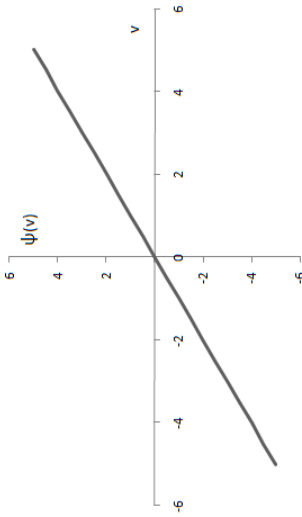
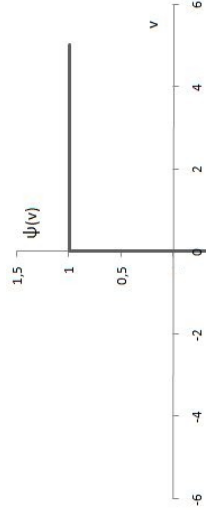
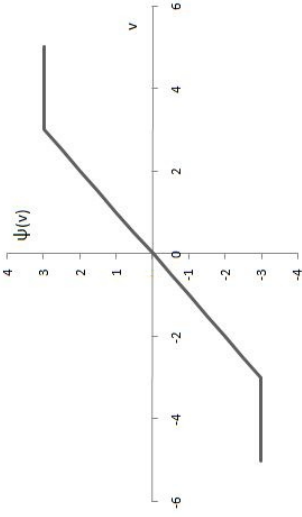
Eine der Vereinfachungen des obigen Algorithmus im Vergleich zur üblichen Vorgehensweise bezieht sich darauf, dass für jede der 500 aufgestellten  $H_1$  nur drei  $C$ -Schritte berechnet werden müssen. Aus mehreren Untersuchungen wurde die Erkenntnis gewonnen, dass die Werte  $Q_3$  für die Subsets  $H_3$ , die zu einer verfälschten Lösung führen, deutlich größer sind. Somit ist ein Vergleich nach dem dritten  $C$ -Schritt ausreichend, um nur mehr mit den Daten der zehn kleinsten  $Q_3$  weiterzurechnen und ein (relativ) fehlerfreies Ergebnis zu erhalten.

## 2.4 Gegenüberstellung

In Tabelle 2.2 werden in Anlehnung an Jäger et al. (2005) der Methode der kleinsten Quadrate zwei robuste Alternativen exemplarisch gegenübergestellt. Durch den visuellen Vergleich sind die Unterschiede erkennbar und die Systematiken nachvollziehbar.

Die Verlust- bzw. Schätzfunktion des Biber-Schätzers wird anhand des zugrunde liegenden Huber-Schätzers dargestellt. Der betreffende Ausdruck kann mittels (2.41) auf (2.36) umgeformt werden. Für die entsprechende Abbildung wurde  $k = 3$  angenommen. Je nach Wahl des Grenzwertes  $k$  werden die Verlust- bzw. Einflussfunktion flacher oder steiler. Die robusten Verfahren (L1-Norm und Biber-Schätzer) besitzen eine begrenzte Einflussfunktion, was durch den horizontalen Funktionsverlauf repräsentiert wird. Unabhängig von der absoluten Größe einer Verbesserung wird bei der L1-Norm der Einfluss auf  $\pm 1$  festgesetzt. Beim Biber-Schätzer erfolgt bis zur Grenze  $k$  ein an den Verbesserungsbetrag angepasster Verlauf. Der Einfluss jener Daten, die  $k$  überschreiten, wird auf den Grenzwert eingeschränkt. Die Einflussfunktion der Methode der kleinsten Quadrate unterliegt keiner Restriktion;  $\Psi(v_i)$  verläuft linear und geht bei wachsenden Verbesserungen gegen Unendlich, ein grober Fehler beeinflusst das Ergebnis im vollen Umfang.

Tabelle 2.2: Gegenüberstellung einiger behandelter Verfahren (vgl. Jäger et al., 2005, S.117, 125ff.)

	Methode der kleinsten Quadrate	L1-Norm	Biber-Schätzer
Verlustfunktion $\rho$	 $\rho(v_i) = \frac{1}{2}  v_i ^2 = \frac{1}{2} v_i^2$	 $\rho(v_i) =  v_i $	 $\rho_c(v_i) = \begin{cases} \frac{1}{2} v_i^2 & \text{für }  v_i  < k \\ k v_i  - \frac{1}{2} k^2 & \text{für }  v_i  \geq k \end{cases}$
Einflussfunktion $\Psi$	 $\Psi(v_i) = \text{sign}(v_i) \cdot  v_i  = v_i$	 $\Psi(v_i) = \frac{v_i}{ v_i } = \text{sign}(v_i)$	 $\Psi_k(v_i) = \begin{cases} v_i & \text{für }  v_i  < k \\ \text{sign}(v_i) \cdot k & \text{für }  v_i  \geq k \end{cases}$

## 2.5 Streckenkorrektur

Im Folgenden werden die relevanten Formeln für die Reduktion der Streckenbeobachtungen vorgestellt. Korrekturen sind notwendig, um die Messungen von äußeren Einflüssen zu befreien und geometrisch richtig zu stellen.

### 2.5.1 Meteorologische Reduktion

Die Distanzmessung in der Geodäsie erfolgt mittels elektromagnetischer Wellen im Bereich des nahen Infrarots oder sichtbaren Lichts. Während der Ausbreitung dieses Signals kommt es zu einer Beeinflussung durch die Atmosphäre. Die dort herrschenden Bedingungen (Temperatur, Luftdruck, Wasserdampfdruck) bewirken eine Veränderung von Geschwindigkeit, Phasenlage und Amplitude des Signals und bedingen eine zu lange oder zu kurze Distanzmessung (Kahmen, 2006, S.170).

Die meteorologische Korrektur erfolgt unter anderem anhand der Formel von Barrel und Sears (2.60), mit welcher der Gruppenbrechungsindex  $n_{Gr}$  des Lichtes in trockener Luft und Normalatmosphäre berechnet wird (ebd., S.171). Dabei ist  $\lambda$  die Wellenlänge der Trägerwelle des Distanzmessgerätes, welche in  $\mu m$  einzusetzen ist.

$$(n_{Gr} - 1)10^6 = 287,604 + 3\frac{1,6288}{\lambda^2} + 5\frac{0,0136}{\lambda^4} \quad (2.60)$$

Mittels der Formel von Kohlrausch (2.61) ist  $n_{Gr}$  des Weiteren auf den Luftzustand bei der Messung (Temperatur  $t$  in °C, Luftdruck  $p$  in Torr) umzurechnen (Joeckel und Stober, 1991, S.73). Es wird hierbei der Ausdehnungskoeffizient der Luft  $\alpha$  benötigt, welcher den Wert 0,003661 annimmt. Der Partialdruck des Wasserdampfes  $e$  ist in der Einheit Torr einzuführen und errechnet sich anhand Formel (2.62) (Kahmen, 2006, S.171). Die Konstante  $D$  ist entsprechend dem Anwendungsfall auf Messungen über Wasser oder über Eis zu beziehen. Die Feuchttemperatur wird mit  $t'$  bezeichnet,  $E'$  stellt den Sättigungsdampfdruck bei der Temperatur des feuchten Thermometers dar.

$$n_L = 1 + \frac{n_{Gr} - 1}{(1 + \alpha t)} \frac{p}{760} - \frac{5,5 \cdot 10^{-8}}{1 + \alpha t} e, \quad (2.61)$$

$$e = E' - Dp(t - t'). \quad (2.62)$$

Die anzubringende Korrektur  $m_{met.}$  resultiert aus Formel (2.63). In Anlehnung an (2.61) wird der Brechungsindex  $n'_L$  mit den, das jeweilige Gerät betreffenden, Normbezugswerten  $t_B, p_B$  berechnet, wobei der letzte Term aufgrund seines geringen Einflusses vernachlässigt werden kann.

$$m_{met.} = \frac{n'_L}{n_L} \quad (2.63)$$

Die meteorologisch reduzierte Strecke  $D$  ergibt sich durch Anbringen des Reduktionswertes  $m_{met.}$  an die gemessene Strecke  $s_{gem.}$ .

$$D = s_{gem.} \cdot m_{met.} \quad (2.64)$$

## 2.5.2 Geometrische Reduktion

Die als Bogen angenommene Strecke  $D$  ist um einige Einflüsse zu korrigieren. Wegen der Bahnkrümmung ist die Korrektur  $r_K$  anzubringen, um auf die schräge Sehne  $S^R$  zwischen  $P_1$  und  $P_2$  zu gelangen (siehe Abb. 2.1). Aufgrund der Neigung und Höhe über der Bezugsebene, wird die Reduktion  $r_H$  angewendet und die Sehne  $S^0$  des im Bezugshorizont verlaufenden Bogens der Erdoberfläche erreicht. Der Erdkrümmung wird mit der Korrektur  $r_E$  Rechnung getragen. Dieser Faktor beschreibt den Übergang von  $S^0$  auf  $S$ . Für Strecken unter zehn Kilometer Länge sind die Reduktionen  $r_K$  und  $r_E$  nicht zwingend anzubringen (Kahmen, 2006, S.173). Zur Berechnung der Korrekturen ist vor allem die Kenntnis des Radius  $R$  der Bezugskugel erforderlich, für die Reduktion  $r_K$  ist der Refraktionskoeffizienten  $k$  von Bedeutung.

$$r_K = -k^2 \frac{D^3}{24R^2}, \quad (2.65)$$

$$S^R = D + r_K, \quad (2.66)$$

$$r_H = \left( \sqrt{\frac{1 - \left(\frac{\Delta H}{S^R}\right)^2}{\left(1 + \frac{H_1}{R}\right)\left(1 + \frac{H_2}{R}\right)}} - 1 \right) S^R, \quad (2.67)$$

$$r_E = \frac{D^3}{24R^2}. \quad (2.68)$$

Die reduzierte Strecke ergibt sich aus

$$S = D + r_K + r_H + r_E. \quad (2.69)$$

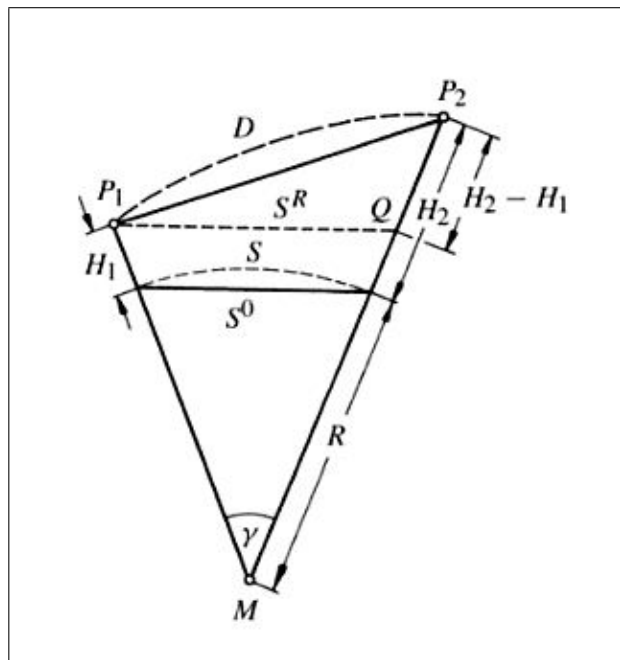


Abbildung 2.1: anzubringende Reduktionen an die gemessenen Schrägstrecken (ebd., S.174)

Durch die Verwendung von Gauß-Krüger-Koordinaten muss die Strecke mit der nachfolgenden Formel entsprechend angepasst werden. Die geometrisch (und meteorologisch) vollständig reduzierte Strecke ergibt sich zu (Joeckel und Stober, 1991, S.172)

$$s_{\text{GK}} = S \cdot \left( 1 + \frac{1}{6R^2} (y_1^2 + y_1 y_2 + y_2^2) \right), \quad (2.70)$$

wobei  $y_1, y_2$  den Gauß-Krüger-Koordinaten des Stand- und Zielpunktes in y-Richtung entsprechen.



# Kapitel 3

## Testdaten

Zur empirischen Untersuchung der robusten Verfahren werden terrestrische Netzmessungen verwendet. Diese wurden von 28.09. - 5.10.1993 im Zuge der Landesvermessung Feldübung von Studenten des damaligen Instituts für Geoinformation an der Technischen Universität Wien im Gebiet der Hohen Wand, welches sich westlich von Wiener Neustadt (NÖ) befindet, aufgenommen.

Es handelt sich dabei um 124 Richtungs- und 55 Distanzmessungen, sowie um drei Höhenunterschiede, die mittels Nivellement bestimmt wurden. Die Anforderungen an das zu messende Präzisionsnetz bestanden in der Einhaltung einer Lagegenauigkeit von  $\sigma_{x,y} = 12\text{mm}$ . Um diesem Anspruch gerecht zu werden, wurden die Richtungen in zwei Kreislagen und vier Sätzen beobachtet, wobei drei verschiedene Messgeräte eingesetzt wurden. Die Streckenmessungen erfolgten mit zwei Geräten und in vierfacher Ausführung. Einen Überblick über die verwendeten Messgeräte und deren erreichbare Genauigkeit schafft Tabelle 3.1 (Wild-Heerbrugg, 2014; Angst et al., 1994). Die Richtungs- und Streckenmessungen liegen nur analog in Form von Messprotokollen vor und sind für die weitere Verarbeitung zu digitalisieren.

Tabelle 3.1: Verwendete Geräte mit Genauigkeitsangaben

Gerät	Messgröße	Genauigkeit
Wild T1610	Hz, V	0,5mgon
Wild T2000	Hz, V	0,15mgon
Wild T2002	Hz, V	0,15mgon
Wild DI1600	Strecke	3mm + 2ppm
Wild DI2000	Strecke	1mm + 1ppm

Das Netz besteht aus acht Netzknoten, von denen fünf als Standpunkte genutzt wurden. Es besitzt eine Ausdehnung von mehreren Kilometern. Abbildung 3.1 zeigt die Geometrie des Netzes und seine ungefähre Lage in der Natur. Magentafarbene Linien symbolisieren die gemessenen Strecken, blaue Pfeile die Richtungsbeobachtungen.

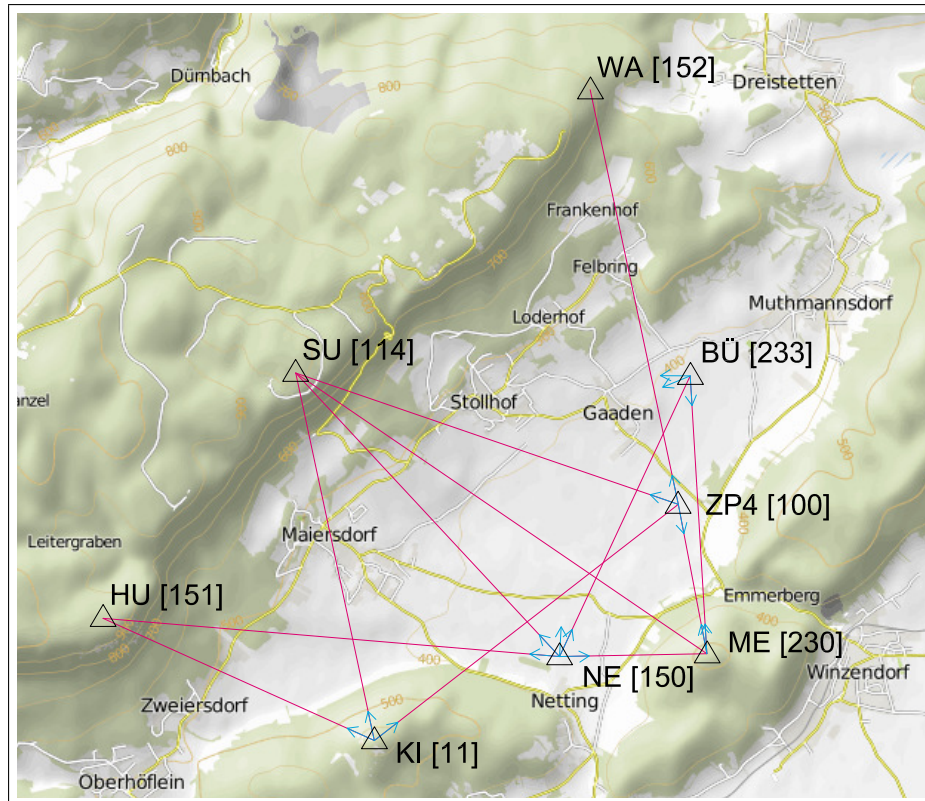


Abbildung 3.1: Ungefähre Lage des Vermessungsnetzes  
(Kartengrundlage: Österreich Werbung Wien, 2014)

Tabelle 3.2 listet alle Netzpunkte inklusive ihrer Position (Gauß-Krüger-Koordinaten, M34) und Verwendung auf. Sieben der acht Punkte waren zum Zeitpunkt der Vermessung amtlich eingetragene Triangulierungspunkte mit entsprechend bekannten Koordinaten und Höhenangaben. Punkt ZP-4 wurde eigens für die Feldübung geschaffen und besitzt daher zunächst nur Näherungskordinaten. Die Höhen von ZP-4, ME und BÜ wurden per Nivellement bestimmt und sind nicht dem amtlichen Koordinatenverzeichnis entnommen. Die Spalte „Bez.“ beinhaltet die Kurzbezeichnungen der Netzpunkte, so wie sie bei den Berechnungen in Matlab und in den nachfolgenden Kapiteln verwendet werden.

Tabelle 3.2: Netzpunkte

Punktnr.	Abk.	Bez.	Y [m]	X [m]	Höhe [m]	Verwendung
KT 11-75	KI	11	-20 629,91	5 296 245,76	642,700	Standpunkt
KT 549-76	ZP4	100	-18 280,00	5 298 068,00	359,538	Standpunkt
KT 114-75	SU	114	-21 239,35	5 299 087,38	867,820	nur Zielpunkt
KT 150-75	NE	150	-19 197,40	5 296 894,93	378,880	Standpunkt
KT 151-75	HU	151	-22 728,15	5 297 191,77	925,940	nur Zielpunkt
KT 152-75	WA	152	-18 958,94	5 301 272,04	839,930	nur Zielpunkt
KT 230-76	ME	230	-18 056,34	5 296 909,84	457,299	Standpunkt
KT 233-76	BÜ	233	-18 185,52	5 299 063,71	415,105	Standpunkt

### 3.1 Richtungsmessungen

Die zugrundeliegenden Richtungsmessungen aus der Feldübung wurden manuell protokolliert. Durch die heutige ausschließliche Nutzung der digitalen Speicherung von Messwerten ist diese Vorgehensweise obsolet und vielfach nicht mehr bekannt. Um Unklarheiten vorzubeugen, wird nachfolgend auf die händischen Aufzeichnungen eingegangen und die Datenaufbereitung für die weitere Nutzung mit der Auswertesoftware erläutert.

#### 3.1.1 Aufnahme

Abbildung 3.2 zeigt einen Ausschnitt aus den damals üblichen Messprotokollen. Exemplarisch sind die dokumentierten Größen für eine komplette Satzmessung ausgewählt.

✓ 30.09. ST. 2  
30.09.1993

M 1		GRUNDLAGENMESSUNG				GRUPPE... B	INSTRUMENT WILD T. 3510				BEOBACHTER H. INTERMITZ	
		RICHTUNGEN				ZENITDISTANZEN				SEITEN		ANMERKUNG
VISUR		KREISLAGE	RICHTUNG	± SATZSCHL	REDUZIERTES	REDUZIERTES R.	ABLESUNG	L + R	1. MESSUNG			
VON	NACH			1/2 (L+R)	MITTEL	ORIENTIERTE R.	AM	L - R	2. MESSUNG			
			L + R			HÖHENKREISE	1/2 (L - R)	SCHIEF	HORIZONTAL			
			g : c cc	c : cc	c : cc	g : c cc	g : c cc	g : c cc	g : c cc			
2P4	ME	L	188 : 0349	± 0		0 : 0000					TEMPERATUR 9°C	
		R	388 : 0324	03 : 22	03 : 22							
2P4	4V	L	287 : 8627	± 0		89 : 8307					LUFDRUCK 976	
		R	87 : 8631	86 : 29	86 : 29							
2P4	SU	L	321 : 3785	- 1		133 : 3477	89 : 7711	400 : 0057				
		R	121 : 3814	38 : 00	37 : 99		310 : 2346	89 : 7683				
2P4	WA	L	386 : 9691	- 1		198 : 9364	90 : 7762	400 : 0072				
		R	186 : 9683	86 : 87	86 : 86		309 : 2310	90 : 7726				
2P4	BU	L	6 : 3201	- 1		218 : 2882	96 : 4975	400 : 0073				
		R	206 : 3209	32 : 05	32 : 04		303 : 5098	86 : 4838				
2P4	GR	L	65 : 4149	- 2		277 : 3825						
		R	265 : 4148	41 : 49	41 : 47							
2P4	ME	L	188 : 0317	- 2		0 : 0000						
		R	388 : 0330	03 : 24	03 : 22							

Abbildung 3.2: Satzmessung für Richtungen. Auszug aus dem Messprotokoll.

Unter „Visur“ ist links der Standpunkt und rechts der Zielpunkt angeführt. Die Visur zum Anfangspunkt, hier ME, wird auch am Ende des Satzes durchgeführt, um eine Verdrehung des Geräts bzw. Stativs oder Horizontalkreises aufzudecken. In der Spalte „Kreislage“ wird als „L“ die erste Kreislage und „R“ dementsprechend als zweite Kreislage aufgefasst. Die Bezeichnungen L (links) und R (rechts) leiten sich von der Lage des Vertikalteilkreises in Bezug zum Fernrohr ab. Die Spalte „Richtung“ enthält die Messwerte für die Horizontalrichtungen in der Einheit Gon.

Die ersten beiden Nachkommastellen werden als Neuminuten (<sup>c</sup>) bezeichnet, die dritte und vierte Nachkommastelle als Neusekunden (<sup>cc</sup>). Bei fehlerfreien Messungen sollten sich die erste und zweite Kreislage exakt um 200<sup>g</sup> unterscheiden. Durch unvermeidliche Messfehler tritt dies selten ein, trotzdem bietet der Vergleich der beiden Kreislagen eine erste Überprüfung auf grobe Fehler. Ist die Differenz zu groß, sollte die Messung wiederholt werden. In der anschließenden Spalte sind Kontrollrechnungen zu finden. In der dritten Zeile werden die entsprechenden Nachkommastellen der gemessenen Horizontalrichtungen addiert und in der zweiten Zeile die Summe halbiert, wodurch das Mittel gebildet wird. Dies wird für alle Messungen eines Satzes durchgeführt, damit der Satzschluss abgelesen werden kann. Dieser ergibt sich aus der Differenz zwischen dem Mittel des Anfangspunktes und jenem des Endpunktes. Ist diese Differenz nicht Null, so weist dies auf eine oben erwähnte Verdrehung hin. Um den Effekt zu korrigieren, ist er an die Messungen anzubringen. Dazu muss der Differenzwert kumulativ auf alle den Satz beinhaltenden Messungen aufgeteilt werden. Die Größe der einzeln angebrachten Korrekturen kann dabei selbst festgelegt werden und ist in die erste Zeile in Spalte fünf einzutragen.

Für den obigen Ausschnitt bedeutet dies:

$$\begin{array}{rcl}
 \text{von ZP4 nach ME:} & \text{L+R :} & 319^{\text{cc}} + 324^{\text{cc}} = 643^{\text{cc}} \\
 & \frac{\text{L+R}}{2} : & \frac{643^{\text{cc}}}{2} \approx 322^{\text{cc}} \\
 \text{von ZP4 nach HU:} & \text{L+R :} & 8627^{\text{cc}} + 8631^{\text{cc}} = 17258^{\text{cc}} \\
 & \frac{\text{L+R}}{2} : & \frac{17258^{\text{cc}}}{2} \approx 8629^{\text{cc}} \\
 & & \vdots \\
 \text{von ZP4 nach ME:} & \text{L+R :} & 317^{\text{cc}} + 330^{\text{cc}} = 647^{\text{cc}} \\
 & \frac{\text{L+R}}{2} : & \frac{647^{\text{cc}}}{2} \approx 324^{\text{cc}} \\
 \text{Satzschluss:} & & 324^{\text{cc}} - 322^{\text{cc}} = 2^{\text{cc}}
 \end{array}$$

Das reduzierte Mittel in Spalte sechs errechnet sich aus dem Mittelwert der beiden Kreislagen (Spalte fünf, Zeile zwei)  $\pm$  dem Satzschluss (Spalte fünf, Zeile eins). Die Vorkommastellen werden aus der ersten Kreislage übernommen, die Nachkommastellen resultieren aus der Mittelbildung und dem Satzschluss. Für die reduzierten Mittel ergibt dies:

$$\begin{array}{rcl}
 \text{von ZP4 nach ME:} & 322^{\text{cc}} \pm 0 & = 322^{\text{cc}} \\
 \text{von ZP4 nach HU:} & 8629^{\text{cc}} \pm 0 & = 8629^{\text{cc}} \\
 & & \vdots \\
 \text{von ZP4 nach ME:} & 324^{\text{cc}} - 2^{\text{cc}} & = 322^{\text{cc}}
 \end{array}$$

Optional kann die reduzierte Richtung gebildet werden. Hierfür wird das reduzierte Mittel des ersten anvisierten Punktes als reduzierende Richtung verwendet, auf die sich alle anderen Messungen im jeweiligen Satz beziehen. In diesem konkreten Beispiel wird die erste Visur zu ME um sein eigenes reduziertes Mittel reduziert, was die Nullrichtung ergibt. Die reduzierte Mittel zum zweiten Punkt HU wird um das reduzierte Mittel von ME reduziert, etc.

Für die vorliegende Arbeit werden die reduzierten Richtungen nicht verwendet, der Vollständigkeit halber sei ihre Berechnung dennoch angeführt.

reduzierende Richtung:  $188,0322^g$

reduzierte Richtungen:

$$\begin{aligned}
 \text{von ZP4 nach ME: } & 188,0322^g - 188,0322^g = 0,0000^g \\
 \text{von ZP4 nach HU: } & 287,8629^g - 188,0322^g = 99,8307^g \\
 & \vdots \\
 \text{von ZP4 nach ME: } & 188,0322^g - 188,0322^g = 0,0000^g
 \end{aligned}$$

Die orientierte Richtung, ebenfalls in Spalte sieben, kann erst nach Anbringen der Orientierungsunbekannte ausgefüllt werden und ist für die Aufzeichnung im Feld nicht relevant.

Die entsprechenden Spalten für die Zenitdistanzmessungen sind für die vorliegende Arbeit nicht von Bedeutung, da die Horizontierung der Streckenmessungen über den Höhenunterschied zwischen Stand- und Zielpunkt erfolgt. Unbedingt sollte aber das jeweils verwendete Gerät auf dem Protokoll vermerkt werden, um beim Aufstellen der Gewichtsmatrix den Beobachtungen die entsprechenden Standardabweichungen zuordnen zu können. Der Übersichtlichkeit halber und um etwaige Fragen bezüglich des Messvorgangs klären zu können, ist es vorteilhaft, die einzelnen Protokolle mit dem Namen des Beobachters und dem Datum zu versehen.

### 3.1.2 Aufbereitung

Als ersten Schritt ist es unerlässlich, die eingetragenen Werte zu kontrollieren und auf allfällige Rechen- und Schreibfehler zu überprüfen.

Tabelle 3.3: Aufbau der Textdatei „richtungen.txt“

Spalte	Inhalt
1	Standpunkt
2	Zielpunkt
3	Kreislage I [gon]
4	Kreislage II [gon]
5	Richtungsmessgenauigkeit [mgon]
6	reduziertes Mittel [gon]
7	Satznummer

Für die Weiterverarbeitung in Matlab wird das Textfile „richtungen.txt“ erstellt und die wichtigsten Spalten des Messprotokolls übernommen. In Tabelle 3.3 ist der Aufbau der Datei dargestellt. Die Satznummer in Spalte 7 ist für die richtige Berechnung der (Orientierungs-)Unbekannten und die Zuordnung innerhalb des Auswerteprogramms wichtig. Richtungsbeobachtungen, die demselben Satz zugehören, wird dieselbe Satznummer zugeordnet.

### 3.2 Streckenmessungen

Analog zu den Richtungsmessungen wird der Inhalt der Messprotokolle, sowie die Aufbereitung der Daten besprochen. Hinzu kommen die Berechnung und das Anbringen der erforderlichen Korrekturen.

#### 3.2.1 Aufnahme

Ein Abriss der Streckenmessprotokolle ist in Abbildung 3.3 dargestellt. Neben den Kurzbezeichnungen für Stand- und Zielpunkt sind in der vierten Spalte die dazugehörigen Höhen eingetragen. Die Beschriftung „i“ steht hierbei für die Instrumentenhöhe am Standpunkt, „z“ für die Zielhöhe des Zielpunktes. Die Einheit für beide Höhen ist Zentimeter. In den Spalten fünf und sechs sind die meteorologischen Parameter Temperatur in der Einheit Celsius und Luftdruck in mbar zu finden, die jeweils während der Durchführung der Streckenmessung an Stand- und Zielpunkt gemessen wurden. Diese Werte finden bei der Streckenkorrektur Verwendung. In der vorletzten Spalte ist die gemessene Schrägstrecke in Metern angeführt. Aus denselben Gründen wie auch bei den Richtungsmessungen, ist es notwendig, das verwendete Gerät am Protokoll zu vermerken.

GRUPE B    Distanzer: DI 1600    513 30.09  
Operat.....

V109a Feldbuch für Streckenmessungen-..... Nr.....							Beobachter: MILTER MATR	
Datum	Standpunkt	Bezugs- punkt	i	Temperatur	Luftdruck	atm. Korr.c	schräg gem. Strecke s'	S hor.
	Zielpunkt		z			s.c		S'atm.korr.
27.09.2009	ZP4		1,724	10°	976	m bar	1002,675	
	BÜ		1,421	9,5	✓			
	ZP4		1,724	10°			1002,676	
	BÜ		1,421	9,5				
	ZP4		1,724	10°			1002,675	
	BÜ		1,421	9,5				
ZP4		1,724	10°			1002,675		
BÜ	1002 675		1,421	9,5				

Abbildung 3.3: Auszug aus dem Messprotokoll für Strecken



### 3.2.2 Streckenkorrektur und Aufbereitung

Bevor die rohen Streckenbeobachtungen in die Ausgleichung eingeführt werden können, sind sowohl meteorologische, als auch geometrische Reduktionen anzubringen. Zum einen werden dadurch äußere Einflüsse korrigiert, zum anderen werden die gemessenen Schrägstrecken in ebene Strecken umgerechnet. In Kapitel 2.5 wurden die notwendigen Formeln bereits vorgestellt.

Für die Reduktion und in Folge für die Ausgleichung, wird eine Datei „strecken.txt“ mit allen streckenrelevanten Parametern erstellt. Ihr Inhalt ist Tabelle 3.4 zu entnehmen.

Tabelle 3.4: Aufbau der Textdatei „strecken.txt“

Spalte	Inhalt
1	Standpunkt
2	Zielpunkt
3	gemessene Strecke [m]
4	Temperatur (Standpunkt) [°C]
5	Temperatur (Zielpunkt) [°C]
6	Druck (Standpunkt) [mbar]
7	Druck (Zielpunkt) [mbar]
8	Instrumentenhöhe (Standpunkt) [m]
9	orthometrische Höhe (Standpunkt) [m]
10	Instrumentenhöhe (Zielpunkt) [m]
11	orthometrische Höhe (Zielpunkt) [m]
12	GK y-Koordinate (Standpunkt) [m]
13	GK y-Koordinate (Zielpunkt) [m]
14	Streckenmessgenauigkeit [mm]
15	Streckenmessgenauigkeit [ppm]

Anhand eines Beispiels wird die Streckenreduktion im Folgenden im Detail besprochen.

1	100	233	1002.675	10	9.5	976	976	1.724	359.538	1.421	415.105	-18280	-18185.52	3	2
2	100	233	1002.676	10	9.5	976	976	1.724	359.538	1.421	415.105	-18280	-18185.52	3	2

Abbildung 3.4: Auszug aus der Inputdatei „strecken.txt“

Ausgehend von den Daten in Abbildung 3.4 werden die Werte für Temperatur und Druck von Stand- und Zielpunkt gemittelt. Da die Reduktionsformeln den Druck in Torr verlangen, ist der Mittelwert dementsprechend umzurechnen:

$$\text{gemittelte Temperatur: } \frac{10 + 9,5}{2} = 9,75^\circ\text{C}$$

$$\text{gemittelter Druck: } \frac{976 + 976}{2} = 976\text{mbar}$$

$$\text{Umrechnung von mbar auf Torr: } \frac{976}{1,33322} = 732,0732\text{Torr}$$

Der Gruppenbrechungsindex für Licht in Normalatmosphäre nach Barrel und Sears errechnet sich mit der Wellenlänge von 850nm für beide Streckenmessgeräte (Leica, 2014; Rüeger, 1996, S.236) und (2.60) zu

$$(n_{Gr} - 1)10^6 = 287,604 + 3\frac{1,6288}{0,85^2} + 5\frac{0,0136}{0,85^4} \Rightarrow n_{Gr} = 1,000294.$$

Für die Formel von Kohlrausch (2.61) wird der Partialdruck des Wasserdampfes  $e$  benötigt. In (2.62) wird  $E' = 9,426$  gesetzt (Angst et al., 1994), die Konstante  $D$  beträgt für diese Anwendung 0,000662 (Kahmen, 2006, S.171). Die Feuchttemperatur  $t'$ , die aufgrund ihres geringen Einflusses nicht gemessen wurde, wird  $t' = t$  gesetzt. Der Partialdruck des Wasserdampfes vereinfacht sich daher zu  $e = E'$ , für den Brechungsindex  $n_L$  folgt somit

$$n_L = 1 + \frac{1,0003 - 1}{(1 + 0,003661 \cdot 9,75)} \frac{976}{760} - \frac{5,5 \cdot 10^{-8}}{1 + 0,003661 \cdot 9,75} 9,426 = 1,000273.$$

Mit den Normbezugswerten  $t_B = 12^\circ\text{C}$  und  $p_B = 760\text{Torr}$  (Angst et al., 1994) ergibt sich

$$n'_L = 1 + \frac{1,000294 - 1}{(1 + 0,003661 \cdot 12)} \frac{760}{760} = 1,000282.$$

Die meteorologisch reduzierte Strecke beträgt

$$D = 1002,675 \cdot \frac{1,000282}{1,000273} = 1002,684\text{m}.$$

Für die geometrische Reduktion ist der Radius für die Bezugskugel (Schmieggugel) zu bestimmen. Aufgrund der Genauigkeitsabschätzungen in Angst et al. (1994) reicht es aus, diesen mittels der Näherungsformel  $\sqrt{M \cdot N}$  zu berechnen. Die erforderlichen Größen, Meridiankrümmungshalbmesser  $M$  und Querkrümmungshalbmesser  $N$ , sind aus der großen und kleinen Halbachse des zugrunde liegenden Bessel Ellipsoid abzuleiten (Joeckel und Stober, 1991, S.159).

$$\begin{aligned} c &= \frac{a^2}{b} = \frac{6377397,155^2}{6356078,963} = 6398786,848\text{m}, \\ e'^2 &= \frac{a^2 - b^2}{b^2} = 0,00672\text{m}, \\ V^2 &= 1 + e'^2 \cdot \cos^2 B = 1,00291\text{m} \quad \text{mit geographischer Breite } B = 48,83887^\circ, \\ M &= \frac{c}{V^2} = 6370950,140\text{m}, \\ N &= \frac{c}{V} = 6389494,457\text{m}, \\ R &= \sqrt{M \cdot N} = 6380215,561\text{m}. \end{aligned}$$



Mit dem Näherungswert für den Refraktionskoeffizient  $k = 0,13$  folgt gemäß den Formeln (2.65) und (2.66)

$$\begin{aligned}r_K &= -1,74 \cdot 10^{-8}\text{m}, \\S^R &= 1002,684\text{m}.\end{aligned}$$

In Formel (2.67) ist die Höhe von Stand- und Zielpunkt, sowie deren Höhendifferenz gefordert. Hierbei ist zu beachten, dass sich diese Höhen aus der orthometrischen Punkthöhe und der Instrumentenhöhe zusammensetzen.

$$\begin{aligned}H_1 &= 359,538 + 1,724 = 361,262\text{m}, \\H_2 &= 415,105 + 1,421 = 416,526\text{m}, \\ \Delta H &= H_2 - H_1 = 55,264\text{m}, \\r_H &= -1,585\text{m}.\end{aligned}$$

Gemeinsam mit der Reduktion  $r_E = 1,03 \cdot 10^{-6}\text{m}$  aus (2.68) ergibt sich die Strecke

$$S = 1001,099\text{m}.$$

Nach Anbringen der Korrektur für die Gauß-Krüger-Projektion aus (2.70), resultiert die geometrisch und meteorologisch reduzierte Strecke

$$\begin{aligned}y_1 &= -18280\text{m}, \\y_2 &= -18152,52\text{m}, \\s_{\text{GK}} &= 1001,104\text{m}.\end{aligned}$$

Die Outputdatei „sGK\_a.txt“ der Streckenreduktion (siehe Anhang B) enthält für die weitere Verwendung die Standpunkts- und Zielpunktsnummern, sowie die korrigierten Strecken.

# Kapitel 4

## Umsetzung

Die Implementierung der Algorithmen und alle weiteren Analysen werden mit Matlab (Version R2009a) auf einem Notebook mit Intel(R) Core(TM) i5-2410M CPU @ 2.30Ghz, 4 GB RAM, 64 Bit-Betriebssystem und Windows 7 Home Premium durchgeführt. Der Entwurf der Programmcodes erfolgt anhand eines selbst erstellten Simulationsnetzes, in dem die Parameter als bekannt vorausgesetzt sind und welches zu Beginn dieses Kapitels vorgestellt wird. Die vollständigen Quellcodes aller Verfahren sind den Anhängen C-G zu entnehmen, während im Folgenden nur jene Aspekte erläutert werden, welche für die Umsetzung von besonderer Bedeutung sind.

### 4.1 Simulationsnetz

Das simulierte Netz besteht aus fünf Netzpunkten, die in Anlehnung an das zu untersuchende Netz, durch Richtungs- und Streckenbeobachtungen miteinander verknüpft sind und zur Gänze an der Ausgleichung teilnehmen (freier Ausgleich). Die Koordinaten und „Beobachtungen“ des in AutoCAD erstellten Netzes werden mit dem softwareeigenen Tool abgemessen und als fehlerfreie Beobachtungen angesehen. Beim Entwerfen des Simulationsnetzes wird darauf geachtet, dass die Redundanzanteile einen Wert von mind. 0,5 annehmen. Es ergibt sich die folgende Bilanz:

$$\begin{aligned}n_r &= 55 \text{ Richtungsbeobachtungen,} \\n_s &= 28 \text{ Streckenbeobachtungen,} \\u_k &= 10 \text{ (5 Netzpunkte zu je 2 unbekannte Koordinaten),} \\u_o &= 16 \text{ (Anzahl an Richtungssätzen bzw. Orientierungsunbekannten),}\end{aligned}$$

$$r = n - u + d = (n_r + n_s) - (u_k + u_o) + d = 83 - 26 + 3 = 60. \quad (4.1)$$

In Abbildung 4.1 ist das Netz überblicksmäßig dargestellt. Die blauen Pfeile stellen Richtungsbeobachtungen dar, die grünen Linien Streckenmessungen. Die roten Linien symbolisieren die Lage des Horizontalteilkreises und markieren somit Standpunkte.

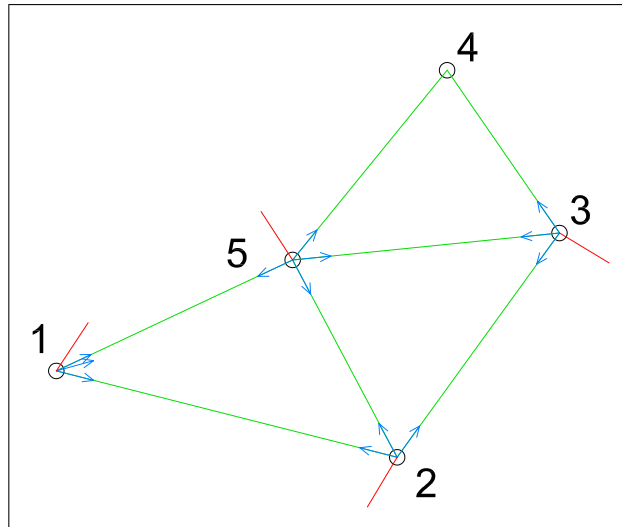


Abbildung 4.1: Simulationsnetz

## 4.2 Grenzen der Verfahren

Oftmals wird auf den Einsatz robuster Schätzer verzichtet und die Verbesserungen aus der Methode der kleinsten Quadrate als Anhaltspunkt genommen, um eine erste Einschätzung auf Vorliegen eines groben Fehlers zu treffen. Messungen, deren Verbesserungen relativ hoch sind, werden dann als Kandidaten für grobe Fehler angesehen. Dieses Urteil ist in vielen Fällen aber nicht gesichert, da sich mehrere fehlerhafte Messungen gegenseitig maskieren können und so die Verbesserung einer fehlerfreien Beobachtung erhöhen. Es bedarf somit robuster Verfahren, die solche Verschmierungseffekte auflösen. Im Extremfall sind aber auch robuste Schätzer nicht in der Lage diesen Effekt aufzudecken. Dies passiert dann, wenn die Messungen so verfälscht werden, dass die Beziehungen untereinander zwar stimmig sind, das System an sich aber nicht mehr der Wirklichkeit entspricht.

Im Zuge der Umsetzung stößt man auf eine weitere Problematik, welche für alle Verfahren ein unlösbares Hindernis darstellt: Besteht ein Richtungssatz nur aus zwei (gleichartigen) Messungen und ist eine davon grob falsch, ist es nicht möglich die fehlerhafte Messung zu detektieren. Die Begründung hierfür liegt in der gemeinsamen Orientierungsunbekannte, durch welche die beiden Messungen miteinander verknüpft sind und sie voneinander abhängig macht. Die Orientierungsunbekannte ist der (nicht robuste) Mittelwert der beiden Einzelorientierungen. Durch das Fehlen eines weiteren Bestimmungsstücks in der betreffenden Satzmessung, fehlt die nötige Information (Redundanz), um eine richtige Entscheidung bezüglich der Lokalisierung des groben Fehlers treffen zu können. Infolge von softwarebedingten Rundungsfehlern oder Rechenungenauigkeiten und dem damit einhergehenden minimalen Betragsunterschied in den beiden Verbesserungen, wählt der Algorithmus jene Richtung mit dem größeren Betrag als Kandidaten für einen groben Fehler aus. Dabei kann die Auswahl zufällig korrekt sein, in gleich vielen Fällen aber auch ebenso falsch. Streckenmessungen sind von dieser Problematik nicht betroffen, da ihre Beobachtungsgleichungen unabhängig von anderen Messungen definiert sind.

## 4.3 Methode der kleinsten Quadrate

Für die Ausgleichung des tatsächlich gemessenen Netzes werden die in Kapitel 3.1 gewonnenen reduzierten Mittel als Richtungsbeobachtungen verwendet. Gemeinsam mit den reduzierten Strecken bilden sie die Gruppe der  $n$  Beobachtungen. Die zu bestimmenden  $u$  Unbekannten setzen sich aus den Lagekoordinaten der acht Netzkpunkte und den Orientierungsunbekannten der Richtungssätze zusammen. Daraus ergibt sich folgende Bilanz bzw. Gesamtredundanz:

$$\begin{aligned}n_r &= 124 \text{ Richtungsbeobachtungen,} \\n_s &= 55 \text{ Streckenbeobachtungen,} \\u_k &= 16 \text{ (8 Netzkpunkte zu je 2 unbekanntem Koordinaten),} \\u_o &= 26 \text{ (Anzahl an Richtungssätzen bzw. Orientierungsunbekannten),}\end{aligned}$$

$$r = n - u = (n_r + n_s) - (u_k + u_o) = 179 - 42 = 137. \quad (4.2)$$

### 4.2.1 Bestimmung von Näherungswerten

Für die Lösung des Ausgleichungsproblems ist es notwendig, Näherungswerte für die Unbekannten zu kennen, um die Beobachtungsgleichungen linearisieren und das Gleichungssystem aufstellen zu können. Als Näherungen der Netzkpunkte werden die Koordinaten aus dem amtlichen Punktverzeichnis herangezogen. Mit diesen Informationen wird das Inputfile „naeherung\_a.txt“, dessen Aufbau in Tabelle 4.1 schematisch dargestellt ist, erstellt (die Endung \_a bezieht sich auf die Ausgangsdaten).

Tabelle 4.1: Aufbau von „naeherung\_a.txt“

Spalte	Inhalt
1	Punktnummer
2	Y-Koordinate [m]
3	X-Koordinate [m]

Für die spätere Verringerung der Redundanz ist zu beachten, dass die nicht verwendeten Messungen oder Punkte vor dem Einlesen der entsprechenden Dateien aus ebendiesen manuell entfernt werden müssen. Zudem muss bei einer Änderung in der Inputdatei der Richtungen darauf geachtet werden, dass die Nummerierung der Sätze pro Standpunkt fortlaufend bleibt.

Die Beschaffung von Näherungswerten für die Orientierungsunbekannten ist aufwendiger. Mit der Beobachtungsgleichung für die Richtungen (4.3) wird deren Berechnung aus den Näherungskordinaten und den reduzierten Mitteln ersichtlich.

$$R_{12} = t_{12} - o = \tan^{-1} \left( \frac{y_2 - y_1}{x_2 - x_1} \right) - o \quad \rightarrow \quad o = t_{12} - R_{12} \quad (4.3)$$

Bei der Berechnung des Richtungswinkels  $t$  ist die Quadrantenzuordnung aus Tabelle 4.2 zu berücksichtigen. Der reine Ergebniswert  $t^*$  nach der Quotientenbildung und Anwendung der Tangensfunktion ist um  $200^{\text{g}}$  bzw.  $400^{\text{g}}$  zu korrigieren. Ist die Zieleinheit wie in der vorliegenden Arbeit nicht das Bogenmaß, muss ein Umrechnungsfaktor  $\rho$  an die Berechnung des Richtungswinkels angehängt werden. Für die Einheit Neugrad ergibt sich für  $\rho$  der Faktor  $\frac{200}{\pi}$ .

Tabelle 4.2: Quadrantenzuordnung bei der Richtungswinkelberechnung (Witte und Sparla, 2011, S.184)

$\Delta y$	$\Delta x$	Quadrant	Ergebnis
+	+	I	$t = t^*$
+	-	II	$t = t^* + 200^{\text{g}}$
-	-	III	$t = t^* + 200^{\text{g}}$
-	+	IV	$t = t^* + 400^{\text{g}}$

Für jede einzelne Richtungsbeobachtung ist eine Einzelorientierung zu berechnen, um damit satzweise eine aus den jeweiligen Einzelorientierungen gemittelte Orientierungsunbekannte angegeben zu können. Diese bilden gemeinsam mit den Näherungskordinaten aus „naeherung\_a.txt“ den genäherten Parametervektor  $\mathbf{X}_0$ , wobei die Parameter in der folgenden Reihenfolge eingeführt werden.

$$\mathbf{X}_0^T = \underbrace{(Y_{11} \ X_{11} \ Y_{100} \ X_{100} \ \cdots \ Y_{233} \ X_{233})}_{\text{Koordinaten mit aufsteigender Punktnummer}} \underbrace{o_1 \ o_2 \ o_3 \ o_4 \ o_1 \ \cdots \ o_8 \ \cdots \ o_1 \ \cdots \ o_2}_{\text{Orientierungsunbekannte mit aufsteigender Standpunktnummer}}$$

Der genäherte Beobachtungsvektor  $\mathbf{L}_0$  wird anhand der Werte aus  $\mathbf{X}_0$  gebildet. Die genäherten Richtungen berechnen sich durch Einsetzen in (4.3), für die Strecken wird die nachfolgende Beobachtungsgleichung benützt.

$$s_{12} = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \quad (4.4)$$

In  $\mathbf{L}_0$  werden die Richtungen und Strecken in nachstehender Abfolge eingeführt.

$$\mathbf{L}_0^T = (R_{0,1} \ \cdots \ R_{0,124} \ s_{0,1} \ \cdots \ s_{0,55})$$

#### 4.2.2 Aufstellen des Gleichungssystems

Bei der Erstellung des Beobachtungsvektors  $\mathbf{L}$  werden für die Richtungen die reduzierten Mittel verwendet. Es wäre ebenso möglich, die erste Kreislage und die um  $200^{\text{g}}$  bzw.  $400^{\text{g}}$  angepasste zweite Kreislage getrennt in die Ausgleichung einzuführen. Die Reihenfolge, in der die Richtungen und Strecken in  $\mathbf{L}$  eingeführt werden, orientiert sich an jener in  $\mathbf{L}_0$ .

Die  $\mathbf{A}$ -Matrix enthält die partiellen Ableitungen der Beobachtungsgleichungen (4.3) und (4.4) nach den Unbekannten in  $\mathbf{X}_0$ . Für die Richtungsmessungen ergeben sich die Ableitungen zu

$$\frac{dR_{12}}{dy_1} = \frac{x_1 - x_2}{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \quad \frac{dR_{12}}{dx_1} = \frac{y_2 - y_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \quad (4.5)$$

$$\frac{dR_{12}}{dy_2} = \frac{x_2 - x_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \quad \frac{dR_{12}}{dx_2} = \frac{y_1 - y_2}{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \quad (4.6)$$

$$\frac{dR_{12}}{do} = -1 \quad (4.7)$$

und für die Strecken zu

$$\frac{ds_{12}}{dy_1} = \frac{y_1 - y_2}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}, \quad \frac{ds_{12}}{dx_1} = \frac{x_1 - x_2}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}, \quad (4.8)$$

$$\frac{ds_{12}}{dy_2} = \frac{y_2 - y_1}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}, \quad \frac{ds_{12}}{dx_2} = \frac{x_2 - x_1}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}. \quad (4.9)$$

Auch hier gilt: Ist die Zieleinheit nicht Radiant, muss ein Umrechnungsfaktor  $\rho$  an die Richtungsableitungen angebracht werden. Die Platzierung der Ableitungen in  $\mathbf{A}$  richtet sich nach der vorgegebenen Reihenfolge in  $\mathbf{X}_0$  und  $\mathbf{L}$ .

Da alle Netzpunkte an der Ausgleichung teilnehmen, handelt es sich um einen freien Ausgleich, genauer gesagt um eine Gesamtspurminimierung. Nachdem Richtungs- und Streckenbeobachtungen vorliegen und sich die Ausarbeitung auf ein Lagenetz beschränkt, ist lediglich der Maßstabsfaktor durch die Messungen alleine bestimmt. Die  $\mathbf{A}$ -Matrix weist einen Rangdefekt von  $d = 3$  auf, verursacht durch die freien Parameter: Translation in x- und y-Richtung und Rotation um die z-Achse (Niemeier, 2008, S.233). Die zugehörige Ränderungsmatrix  $\mathbf{G}^T$  beinhaltet demnach die ersten drei Zeilen von (2.24) und die Gesamtredundanz erhöht sich auf

$$r = n - u + d = 179 - 42 + 3 = 140. \quad (4.10)$$

Für die Gewichtsmatrix  $\mathbf{P}$  ist zunächst die Kovarianzmatrix  $\mathbf{\Sigma}_{\mathbf{LL}}$  zu bilden. Mit den Richtungen in Radiant und den Strecken in Meter, sind auch die Varianzen auf diese Einheiten umzurechnen. Bei den Distanzmessungen ist weiters zu beachten, dass ein streckenabhängiger Anteil anzubringen ist. Anhand konkreter Genauigkeitsangaben wird in Tabelle 4.3 die Berechnung der Varianzen gezeigt.

Tabelle 4.3: Berechnung der Varianzen

Messgenauigkeit	Varianz
Richtungen: 0,5mgon	$\sigma_r^2 = \left(\frac{0,15}{1000}\right)^2$
Strecken: 3mm + 2ppm	$\sigma_s^2 = \left(3^2 + \left(\frac{s_{GK}}{1000} \cdot 2\right)^2\right) \cdot 10^{-6}$

Die Gewichtseinheit a priori  $\sigma_0^2$  für die Kofaktormatrix  $\mathbf{Q}_{LL}$  wird gleich Eins gesetzt, „da dies methodisch einfach ist und Fehler bei der Bewertung des Genauigkeitsniveaus ausgeschaltet werden können. Der früher übliche Ansatz, als Varianzfaktor die Varianz einer der Beobachtungsgruppen zu verwenden, ist rechentechnisch nicht mehr erforderlich“ (Niemeier, 2008, S.157).

### 4.2.3 Lösung des Gleichungssystems

Die Ausgleichung erfolgt nach einem iterativen Schema. Die Anzahl der angegebenen Nachkommastellen der Richtungen und Strecken wird um eine Stelle erhöht und daraus folgende Grenzwerte abgeleitet: Die Unbekannten werden solange neu bestimmt, bis sich der gekürzte Parametervektor  $\mathbf{x}$  von einer Iteration zur nächsten um weniger als  $10^{-4}\text{m}$  bzw. für Richtungen um weniger als  $10^{-5}\text{gon}$  (das entspricht  $0,01\text{mgon}$ ) ändert. Dies wird erreicht, indem am Ende jeder Iteration die ausgeglichenen Unbekannten  $\hat{\mathbf{X}}$  als Näherungsvektor für die Unbekannten  $\mathbf{X}_0$  für den nächsten Durchlauf gesetzt werden. Aus ihnen wird ein neuer gekürzter Beobachtungsvektor  $\mathbf{l}$ , sowie eine neue Designmatrix  $\mathbf{A}$  berechnet. In jedem Durchlauf wird außerdem die Hauptprobe gelöst und ihre maximalen Diskrepanzen in den Richtungs- und Streckenbeobachtungen beobachtet. Um die einzelnen Verfahren später unter verschiedenen Aspekten vergleichen zu können, wird zudem die Anzahl der Iterationen dokumentiert.

## 4.4 L1-Norm

Die Umsetzung der L1-Norm entspricht im Wesentlichen jener der Methode der kleinsten Quadrate. Der einzige Unterschied besteht in der Bildung der Gewichtsmatrix  $\mathbf{P}$ . Während im ersten Iterationsschritt für alle  $p_i = 1$  gesetzt wird, erfolgt die Gewichtsdefinition in den restlichen Iterationen nach Formel (2.31).

Für das Abbruchkriterium  $\epsilon$  der iterativen Gewichtsanzpassung und die Konstante  $c$  bei der Gewichtsbestimmung werden die Erfahrungswerte von Neitzel (2004) übernommen, nämlich  $\epsilon = 10^{-5}$  und  $c = 10^{-9}$ .

Um eine detaillierte Analyse zu ermöglichen, werden die Hauptdiagonalelemente der  $\mathbf{P}$ -Matrix und die geschätzten Parameter jedes Iterationsschrittes in eine eigene Matrix übernommen. So kann der Verlauf der Gewichtung und der davon abgeleiteten Ergebnisse nachvollzogen werden.

Durch die Überprüfung des Algorithmus anhand des Simulationsnetzes können nachfolgende limitierende Faktoren erkannt und dementsprechend berücksichtigt bzw. vermieden werden.

### Wahl der Einheit

Die möglichen Einheiten für Berechnungen mit Richtungen sind beispielsweise das Bogenmaß, Altgrad und Neugrad. Aufgrund der im Vermessungswesen üblichen Verwendung von Neugrad, wird dieses Winkelmaß für die vorliegende Arbeit gewählt.

## Verwendung von unterschiedlichen Beobachtungsarten

Um einen realistischen Vergleich mit den tatsächlichen Messdaten zu schaffen, werden die in der Praxis unvermeidbaren zufälligen Fehler (max. 0,5mgon bzw. 5mm - entsprechen  $1\sigma$ ) an die simulierten Beobachtungen angebracht. Wird mit diesen Daten die L1-Norm-Schätzung berechnet, fallen die Gewichte der Strecken deutlich kleiner aus als jene der Richtungen. Die Streckengewichtung reagiert demnach sensibler auf Abweichungen. Bereits bei zufälligen Fehlern, die im Bereich der Messgenauigkeit liegen, werden die Strecken stark abgewichtet. Diese Abgewichtung würde ohne weitere Untersuchungen des Datenmaterials zu einem falschen Schluss bezüglich des Vorhandenseins von groben Fehlern führen. Zur Abhilfe müssen die Verbesserungen vergleichbar bzw. aneinander angepasst werden, da sich aus ihnen die Gewichte berechnen, siehe (2.31). Die Größe der Anpassung folgt aus der Gegenüberstellung des (robusten) Medians der absoluten Verbesserungen nach dem ersten Iterationsschritt.

$$\text{med } v_r = 2,7 \cdot 10^{-6} \text{ gon} \quad (4.11)$$

$$\text{med } v_s = 1,4 \cdot 10^{-5} \text{ m} \quad (4.12)$$

Der Vergleich der beiden Werte spiegelt die Unverhältnismäßigkeit wider. Die Angleichung besteht darin, die Streckenverbesserungen vor der Berechnung der Gewichte durch die ungefähre Differenz der beiden Mediane, in diesem Fall 10, zu dividieren und so in die Größenordnung der Richtungsverbesserungen zu überführen.

## Beurteilung der Gewichte

Zusätzlich zu den zufälligen Abweichungen werden grobe Fehler im Bereich von  $4\sigma$  angebracht. Mit diesen Veränderungen kann ein Grenzwert gefunden werden, der eine Abschätzung der zulässigen Gewichtsgröße erlaubt, bis zu der eine Beobachtung noch nicht als grober Fehler einzustufen ist. Es zeigt sich, dass eine Beobachtung erst dann als kritisch anzusehen ist und näher untersucht werden sollte, wenn deren Gewicht einen Wert unter 1000 annimmt.

## Abhängigkeiten in Richtungsmessungen

Bei einem Richtungssatz mit drei oder mehr Beobachtungen ist es wesentlich, wieviele Messungen fehlerhaft sind und welches Vorzeichen der grobe Fehler besitzt. Je mehr Messungen betroffen sind, umso näher wandert der Mittelwert aus den Einzelorientierungen zu den falschen Orientierungswerten und die eigentlich richtigen Werte erhalten große Verbesserungen mit entsprechend kleinen Gewichten. Der Effekt wird noch zusätzlich verstärkt, wenn alle groben Fehler dasselbe Vorzeichen besitzen. In diesem Fall streuen die Einzelorientierungen und die Verbesserungen zum Mittelwert sind für alle Beobachtungen etwa gleich groß. Im Extremfall kann dies dazu führen, dass der ganze Satz als grob falsch eingestuft wird.



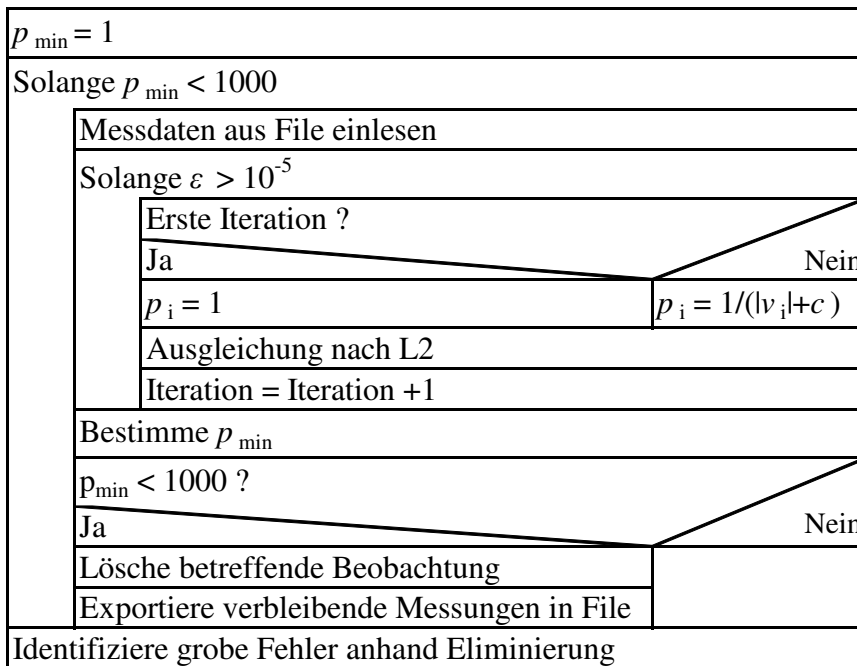


Abbildung 4.2: Struktogramm der L1-Norm

### Iterative Vorgehensweise

Um auf Verschmierungseffekte reagieren zu können, sollte die L1-Norm nicht nur einmal berechnet werden: Nach Erfüllen des Abbruchkriteriums ist es ratsam, nur die Beobachtung mit dem kleinsten Gewicht auszuschließen und den Algorithmus erneut zu starten. Diese Abfolge sollte solange wiederholt werden, bis kein Gewicht mehr unter den Grenzwert fällt.

Abseits dieser Faktoren ist anzumerken, dass jede eliminierte Beobachtung durch ihre Zeilennummer identifiziert wird, welche in einen eigenen Vektor übergeben wird. Am Ende der Berechnung weist dessen Inhalt auf die groben Fehler hin und fasst die Erkenntnisse zusammen. Zur Veranschaulichung ist in Abbildung 4.2 der Ablauf der L1-Norm vereinfacht dargestellt.

## 4.5 Biber-Schätzer

Das Struktogramm in Abbildung 4.3 zeigt überblicksmäßig die Vorgehensweise des umgesetzten Algorithmus. Ausgangspunkt ist eine Ausgleichung nach der Methode der kleinsten Quadrate. Daraus ergibt sich für die Kofaktormatrix der Unbekannten (Niemeier, 2008, S.140f.)

$$\mathbf{Q}_{\mathbf{xx}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} = \mathbf{N}^{-1} \tag{4.13}$$

und daraus abgeleitet für die Kofaktormatrix der ausgeglichenen Beobachtungen

$$\mathbf{Q}_{\mathbf{ff}} = \mathbf{A} \mathbf{N}^{-1} \mathbf{A}^T \tag{4.14}$$

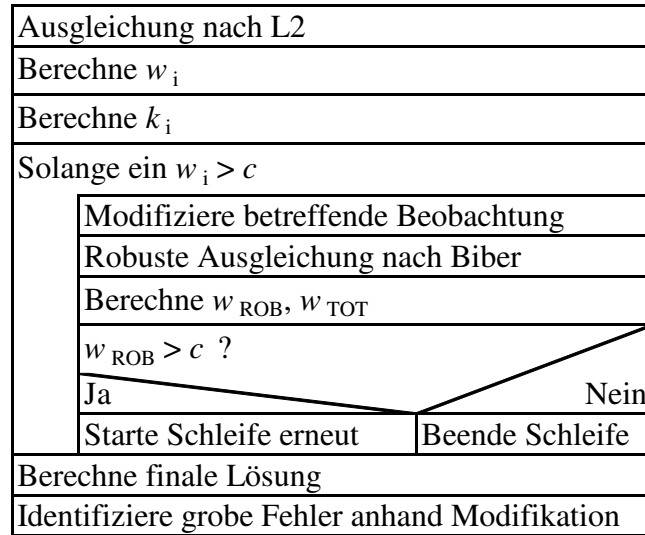


Abbildung 4.3: Struktogramm des umgesetzten Biber-Schätzers

und die Kofaktormatrix der Verbesserungen

$$\mathbf{Q}_{\mathbf{v}\mathbf{v}} = \mathbf{Q}_{\mathbf{I}\mathbf{I}} - \mathbf{Q}_{\mathbf{I}\mathbf{J}} \tag{4.15}$$

Die Hauptdiagonale von  $\mathbf{Q}_{\mathbf{v}\mathbf{v}}$  enthält die Redundanzanteile  $r_i$  (siehe Formel (2.26)). Mit den  $r_i$  und (2.35) - (2.37) können die standardisierten Verbesserungen und individuellen Grenzwerte  $k_i$  berechnet werden.

Für jede Beobachtung wird geprüft, ob der Absolutbetrag der zugehörigen standardisierten Verbesserung größer als der Grenzwert  $c$  (hier:  $c = 3$ , entspricht dem  $3\sigma$ -Bereich) ist. Ist dies der Fall, wird in einem *Test*-Vektor in der entsprechenden Zeile der Wert  $|w_i|$  eingefügt, ansonsten bleibt die Zeile leer.

Nach dieser Überprüfung wird aus allen, den Grenzwert überschreitenden, Beobachtungen jene mit dem größten Wert  $|w_{\text{max}}|$  gesucht und die zugehörige Zeilennummer gespeichert. Sollten mehrere Beobachtungen denselben maximalen Wert aufweisen, so wird die zuerst gefundene Beobachtung für die weitere Bearbeitung verwendet. Dieses Auswahlkriterium ist frei wählbar, es könnte ebenso die Messung, die als zweites oder drittes gefunden wird, ausgesucht werden.

Die zu  $|w_{\text{max}}|$  gehörige Beobachtung wird dazu verwendet, den Differenzvektor mit (2.43) und (2.44) zu erstellen. Die korrespondierende Zeilennummer dient als Identifikator, um die entsprechenden Elemente aus den Matrizen und Vektoren zu extrahieren. Gemeinsam mit der Designmatrix  $\mathbf{A}$ , der Gewichtsmatrix  $\mathbf{P}$  und dem gekürzten Beobachtungsvektor  $\mathbf{I}$  aus der anfangs berechneten Ausgleichung nach der Methode der kleinsten Quadrate, kann eine robuste Ausgleichung in der Form (2.45) und (2.46) durchgeführt werden.

Mit den daraus resultierenden  $\mathbf{v}_{\text{ROB}}$  und  $\mathbf{v}_{\text{TOT}}$  und den davon abgeleiteten absoluten und totalen standardisierten Verbesserungen wird wiederum ein mögliches Überschreiten des Grenzwertes untersucht und die Vorgehensweise wiederholt. Wichtig dabei ist, dass der Differenzvektor gemäß (2.47) erweitert wird. Sollte eine Beobachtung, deren Einfluss schon einmal beschränkt wurde, erneut dem robusten Intervall zugeordnet werden, so sind sowohl die erste, als auch die zweite Korrektur zu berücksichtigen. Zudem werden alle angebrachten  $d_i$  in ein und denselben Differenzvektor  $\Delta \mathbf{d}$  eingetragen. Nach mehreren Iterationen können demzufolge auch mehrere Zeilen besetzt sein.

Es wird solange iteriert, bis der *Test*-Vektor keinen nicht verschwindenden Eintrag mehr enthält, also dessen Norm gleich Null ist. In diesem Fall wurde der Einfluss aller Beobachtungen, die mit einem groben Fehler behafteten sind, auf  $k_i$  eingeschränkt. Bei dieser Behauptung muss allerdings berücksichtigt werden, dass die Definition eines groben Fehlers auf die eingangs festgelegt Größe des Grenzwertes  $c$  zurückgeht und keine allgemeine Gültigkeit besitzt. Je nach Aufgabenstellung und Problematik kann die Größe eines groben Fehlers abgeschätzt und eine individuelle Definition festgelegt werden.

Das zuletzt berechnete  $\mathbf{x}_{\text{ROB}}$  und die entsprechenden  $\mathbf{v}_{\text{TOT}}$  müssen für das robuste Schlussergebnis auf  $\mathbf{X}_0$  und  $\mathbf{L}$  gerechnet werden.

Während der gesamten Berechnung läuft ein Zähler für die Anzahl an durchgeführten Iterationen. Zudem werden die Zeilennummern der modifizierten Beobachtungen in einem eigenen *Fehler*-Vektor aufgelistet. Mit diesem sind nach Ablauf des Algorithmus die Kandidaten für einen groben Fehler auf einen Blick erkennbar.

## 4.6 RANSAC

Ähnlich des in Kapitel 2.3.4 vorgestellten Konzepts, ist das implementierte Rechenprinzip im Ablaufdiagramm in Abbildung 4.4 dargestellt. Die einzelnen Schritte werden nun im Folgenden erläutert.

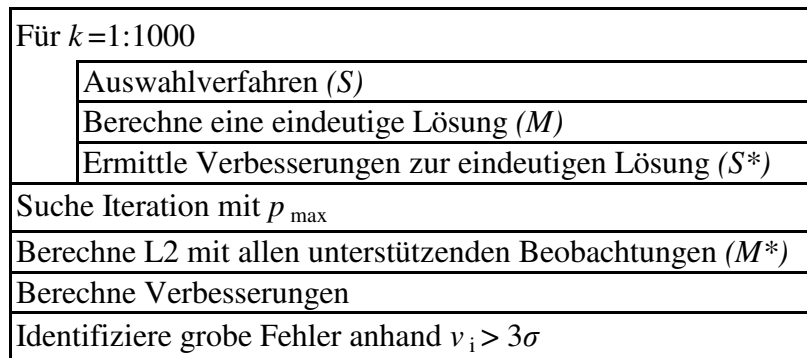


Abbildung 4.4: Struktogramm des implementierten RANSAC-Algorithmus

## Auswahlverfahren

Die automatische Auswahl von  $u = 42$  Beobachtungen zur Berechnung einer eindeutigen Lösung gestaltet sich bei geodätischen Aufgaben schwierig. Zum einen soll gewährleistet sein, dass jeder Netzpunkt bestimmt werden kann, was voraussetzt, dass jeder Punkt zumindest einmal in einer Beobachtungsgleichung enthalten ist. Zum anderen sollte sich die Anzahl der Unbekannten, im Speziellen der Orientierungsunbekannten, nicht ändern, damit die Verfahren untereinander vergleichbar bleiben. Durch diese Voraussetzungen wird ersichtlich, dass nicht jede beliebige Kombination an Beobachtungen auch zu einem lösbaaren Gleichungssystem führt. Aus diesem Grund werden gewisse Restriktionen eingeführt, um geeignete Beobachtungen auszuwählen:

- Würden nur Richtungsmessungen die Lösung bestimmen, würde dies zu einem Datumsdefekt von  $d = 4$  führen, wodurch die Ränderungsmatrix verändert werden müsste und die Verfahren untereinander nicht mehr unter denselben Bedingungen vergleichbar wären. Bei einem reinen Streckenausgleich würde der Datumsdefekt auf  $d = 3$  belassen, aufgrund der geometrischen Verteilung der Streckenmessungen wäre die  $\mathbf{A}$ -Matrix jedoch unzureichend konditioniert und die Ergebnisse wären unzuverlässig. Daraus folgt, dass sowohl Richtungs- als auch Streckenmessungen für die Lösung berücksichtigt werden müssen.
- Damit die Zahl der Orientierungsunbekannten von  $u_o = 26$  beibehalten werden kann, muss aus jedem Richtungssatz zumindest eine Messung ausgewählt werden.
- Um die beiden Beobachtungsgruppen entsprechend gleich zu behandeln, wird für die Strecken ein ähnliches Auswahlverfahren verwendet, wie es bereits bei den Richtungen eingesetzt wird. Es wird eine Art Satzzugehörigkeit gebildet.
- Nach der Bestimmung von 26 Richtungen und 13 Strecken, werden aus den verbleibenden Beobachtungen die fehlenden drei Messungen zufällig ausgewählt.

Die Satzzugehörigkeit betreffend sei erwähnt, dass für RANSAC ein anderes Inputfile für die korrigierten Strecken verwendet wird, als bei den anderen Verfahren. Damit die Distanzmessungen entsprechend ausgewählt werden können, wird die Datei „sGK\_a.txt“ um eine vierte Spalte erweitert („sGK\_ar.txt“), in welcher die „Satznummer“ zu finden ist. Diese wird so vergeben, dass an einem Standpunkt allen Messungen zu ein und demselben Zielpunkt die gleiche Satzbezeichnung zugewiesen wird.

Konkret erfolgen die Auswahl und das zugehörige Aufstellen des eindeutigen Gleichungssystems nach folgendem Prinzip:

Aus der anfänglichen Ausgleichung nach der Methode der kleinsten Quadrate werden mehrere Größen übernommen. Es handelt sich dabei um die Designmatrix  $\mathbf{A}$ , den (genäherten) Beobachtungsvektor  $\mathbf{L}_0$  bzw.  $\mathbf{L}$ , die Hauptdiagonalelemente der  $\Sigma_{\mathbf{LL}}$ -Matrix, sowie um die Einzelorientierungen und Richtungswinkel aller Richtungsbeobachtungen. All diese Größen werden um eine Spalte erweitert, in der die fortlaufende Zeilennummer steht. Auf diese Weise erhält jede Beobachtung einen festen Index. In jedem Satz wird entsprechend der obig festgesetzten Restriktionen eine Beobachtung zufällig ausgewählt. Jede dieser Messungen wird anhand ihrer

Zeilennummer identifiziert und mit dem Wert Eins markiert, wodurch sie sich von den restlichen Beobachtungen unterscheidet. Je nach Kennzeichnung werden die aus der L2-Norm übernommenen Variablen in zwei neue Größen aufgespalten ( $\mathbf{A}_1, \mathbf{L}_1, \dots$  bzw.  $\mathbf{A}_0, \mathbf{L}_0, \dots$ ). Aus den verbliebenen Beobachtungen werden nach demselben Prinzip, diesmal aber ohne Unterscheidung in Richtungen und Strecken, die restlichen drei Messungen ausgewählt. Die Zeilen der nicht verwendeten Beobachtungen werden aus den Größen  $\mathbf{L}_0, \mathbf{A}_0, \dots$  gelöscht. Die Variablen aus beiden Auswahlrunden werden jeweils zu einer Größe zusammengefügt. Dabei wird die eingangs angebrachte Kennzeichnung genutzt, um die Messungen wieder in die aufsteigende Reihenfolge nach fortlaufender Zeilennummer zu bringen. Dies ist von großer Wichtigkeit, da ansonsten eine falsche Zuordnung der Matrizen- und Vektorelemente geschieht.

Zum besseren Verständnis stellt Abbildung 4.5 den Auswahlvorgang unter Nutzung der Kennzeichnungen einfach dar. Die vorletzte Spalte in den Matrizen symbolisiert die Markierung der Messungen entsprechend ihrer Verwendung für die Berechnung der eindeutigen Lösung. In der letzten Spalte befindet sich die fortlaufende Zeilennummer, die vom Stand der Ausgangsmatrix ausgeht und fixiert wird. Dieser Vorgang wird analog für alle aus der L2-Ausgleichung übernommenen Größen wiederholt.

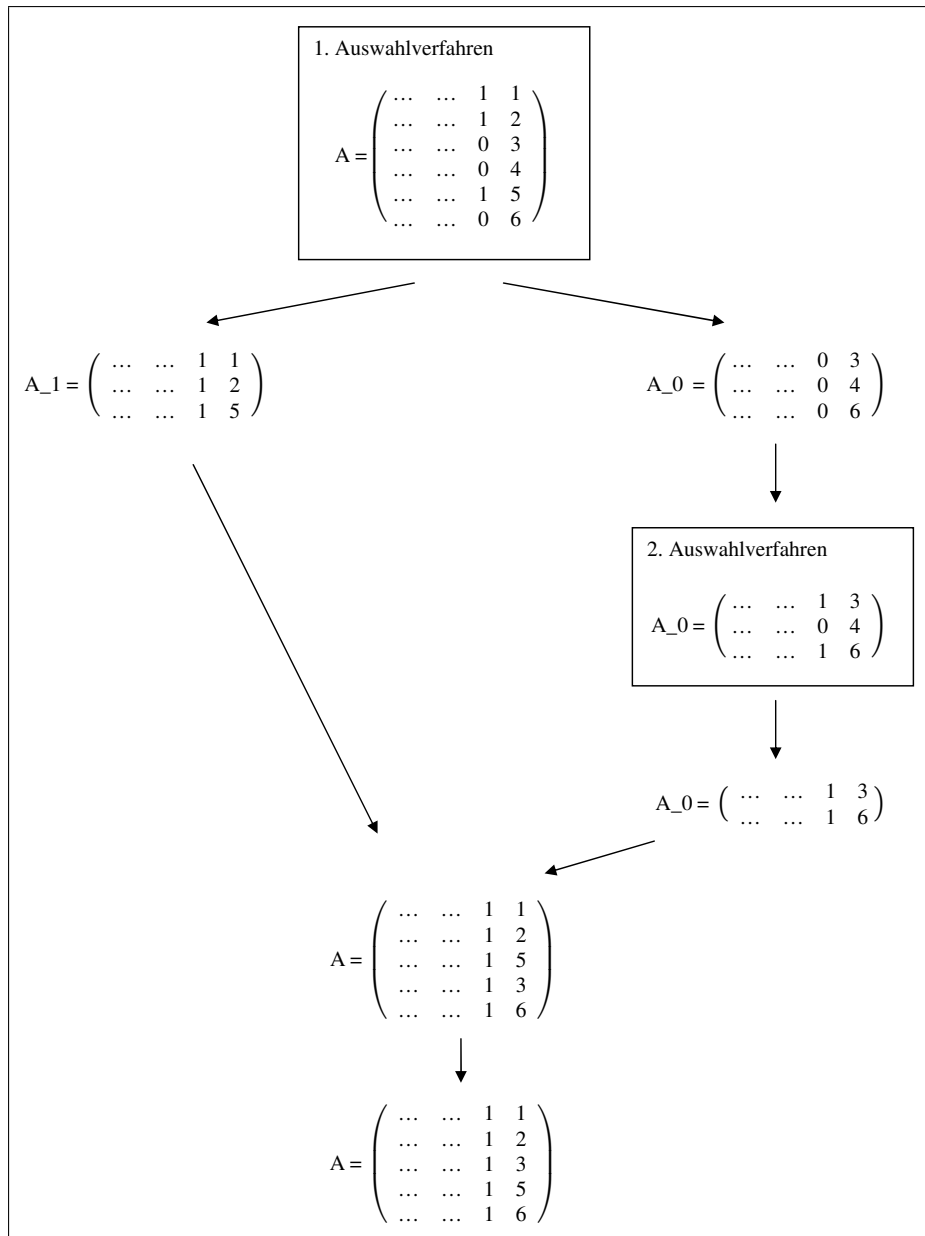


Abbildung 4.5: Schematische Darstellung des Auswahlvorgangs am Beispiel der Designmatrix

### Eindeutige Lösung $M$

Für die eindeutige Lösung wird das lineare Gleichungssystem aufgelöst. Durch die Teilnahme aller Netzpunkte an der Ausgleichung ist die Ränderungsmatrix  $\mathbf{G}$  einzuführen und das Prinzip des freien Ausgleichs anzuwenden. Weiters ist zu beachten, dass mit den verringerten Vektoren aus dem vorigen Abschnitt, vor der Berechnung von  $M$ , neue Näherungswerte für die Orientierungsunbekannten und Richtungsbeobachtungen berechnet werden müssen, da sich das auszugleichende System nun aus weniger Messungen zusammensetzt.

## Consensus Set $S^*$

Bei der Bestimmung gilt: Jede Beobachtung, deren Verbesserung zur aktuellen Lösung innerhalb der Fehlertoleranz liegt, zählt zum Consensus Set. Mit den eindeutig bestimmten Unbekannten werden durch Einsetzen in die beiden Beobachtungsgleichungen alle Messungen aus dem Ausgangsdatensatz berechnet (vgl. Vorgehensweise der Hauptprobe). Die Differenz zwischen dem erhaltenen Ergebnis und der ursprünglicher Messung ergibt die Verbesserung

$$\mathbf{V} = \varphi(\hat{\mathbf{X}}) - \mathbf{L}. \quad (4.16)$$

Die Fehlertoleranz wird wie üblich als die dreifache Standardabweichung definiert. Dazu werden die a priori Genauigkeitsangaben, welche in die Kovarianzmatrix  $\Sigma_{\mathbf{LL}}$  einfließen, verdreifacht und als Grenzwert gespeichert. Für jede Beobachtung wird getestet, ob ihre Absolutverbesserung aus (4.16) unter diesen Wert fällt, die aktuelle Lösung dementsprechend unterstützt oder nicht. Im positiven Fall wird die betroffene Messung in einem eigenen Vektor mit dem Wert 1 markiert, im negativen Fall mit dem Wert 0. Dieser Vektor wird für jede Iteration erstellt und für die Bestimmung des finalen Modells gespeichert.

## Iterationen $k$

Die Schritte eins bis drei werden  $k$ -mal wiederholt, um eine ausreichend große Menge an Lösungen zu berechnen. Im vorliegenden Ausgangsfall, bei dem davon ausgegangen wird, dass die Daten keinen groben Fehler aufweisen, müsste theoretisch eine Iteration genügen, da immer alle Beobachtungen eine beliebige Lösung unterstützen. Für alle anderen Fälle ist der Aufwand  $k$  mit Formel (2.50) abzuschätzen. Für jede neue Iteration ist es unbedingt notwendig, die übergebenen und im Laufe des Verfahrens veränderten Größen auf den Stand der Übergabe rückzusetzen.

## Schwellenwert $t$

In dieser Arbeit wird der Schwellenwert  $t$  als nachfolgender Quotient definiert. Er gibt ein Maß für die Wahrscheinlichkeit von  $S^*$  an (Navratil, 2011, S.126).

$$p = \frac{\text{Anzahl der unterstützenden Beobachtungen}}{\text{Anzahl aller Beobachtungen } n} \quad (4.17)$$

Er errechnet sich aus der Summe der Einträge des Vektors, der aus der Bildung von  $S^*$  entstammt, geteilt durch die Anzahl aller Beobachtungen  $n$ . Anhand der Iterationsnummer, zu dem der jeweilige Quotient abgespeichert wird, kann auf die unterstützenden Beobachtungen zugegriffen werden.

Für die Annahme von ausreißerfreien Ausgangsdaten ist der maximale Quotient von  $p = 1$  zu erwarten und das endgültige Modell wird aus allen Beobachtungen berechnet. Bei fehlerbehafteten Daten wird für die Berechnung von  $M^*$  jenes Consensus Set ausgewählt, das den größten Quotienten besitzt. Sollten zwei oder mehr  $S^*$  einen gleich hohen Wert  $p_{\max}$  erreichen, wird das zuerst berechnete  $S^*$  für die Endrechnung herangezogen.

## Finales Modell $M^*$

Mit der Zeilenkennzeichnung der unterstützenden Beobachtungen können ähnlich wie beim Auswahlvorgang, die betroffenen Zeilen aus den übergebenen Größen extrahiert und eine überbestimmte Ausgleichung nach der Methode der kleinsten Quadrate berechnet werden. Dabei ist wiederum zu beachten, dass die Näherungswerte für die Orientierungsunbekannten und Richtungsbeobachtungen neu berechnet werden müssen.

## Bestimmung der fehlerhaften Beobachtungen

Nachdem das primäre Ziel von RANSAC ist, eine möglichst fehlerfreie Schätzung der Parameter zu ermöglichen und nicht, grobe Fehler zu lokalisieren, muss diese Information selbständig extrahiert werden. Zu diesem Zweck werden nach Berechnung der finalen Lösung, die Verbesserungen aller Beobachtungen zu diesem Ergebnis berechnet. Jene Messungen, die nicht innerhalb der  $3\sigma$ -Fehlertoleranz liegen, werden als (grobe) Fehler angesehen.

Im Unterschied zur Ermittlung der Verbesserungen bei der Bildung der Consensus Set, werden die unterstützenden Beobachtungen mit dem Wert Null markiert und die Kandidaten für fehlerhafte Messungen mit dem Wert Eins. Durch Bildung der Summe des Vektors kann sofort auf die Anzahl der Fehler geschlossen werden. Anhand der Zeilennummer, die der Beobachtungsnummer entspricht, ist die betreffende Messung identifizierbar.

## 4.7 LTS

Der LTS-Algorithmus ist jenem von RANSAC ähnlich (siehe Abbildung 4.4), weshalb das Auswahlverfahren für die Bestimmung der eindeutigen Lösung und die Funktion zur Berechnung der Verbesserungen übernommen werden können und teilweise nur leicht modifiziert werden müssen. Die Trimmingkonstante  $h$  wird so gewählt, dass der maximale Bruchpunkt erreicht wird:

$$h = \frac{n + p + 1}{2} = \frac{179 + 42 + 1}{2} = 111. \quad (4.18)$$

Bei der Berechnung aller Initiaillösungen ist gewährleistet, dass für alle Beobachtungen eine Verbesserung ermittelt werden kann, da aufgrund des Auswahlverfahrens die Anzahl der Unbekannten auf 42 bestehen bleibt. In den nachfolgenden  $C$ -Schritten ist es je nach Reihung der Residuen möglich, dass die Zahl der Orientierungsunbekannten abnimmt, was bei der Bildung der Verbesserungen berücksichtigt werden muss (siehe Anhang G.2). Nach Ermittlung der Verbesserungen werden diese aufsteigend sortiert und die Beobachtungsnummern (entsprechen den Zeilennummern aus dem Ausgangsdatensatz) der ersten 111 Messungen als Input für eine neuerliche Ausgleichung nach der Methode der kleinsten Quadrate übergeben. Wie auch bei der L1-Norm müssen die Verbesserungen der Strecken um einen Faktor 10 dividiert werden, um in die Größenordnung der Richtungsresiduen überführt zu werden. Im dritten  $C$ -Schritt, also der Ermittlung und Reihung der Residuen, sowie der Bildung der Quadratsumme, werden die Beobachtungsnummern der 111 Messungen in einer eigenen Matrix gesammelt.



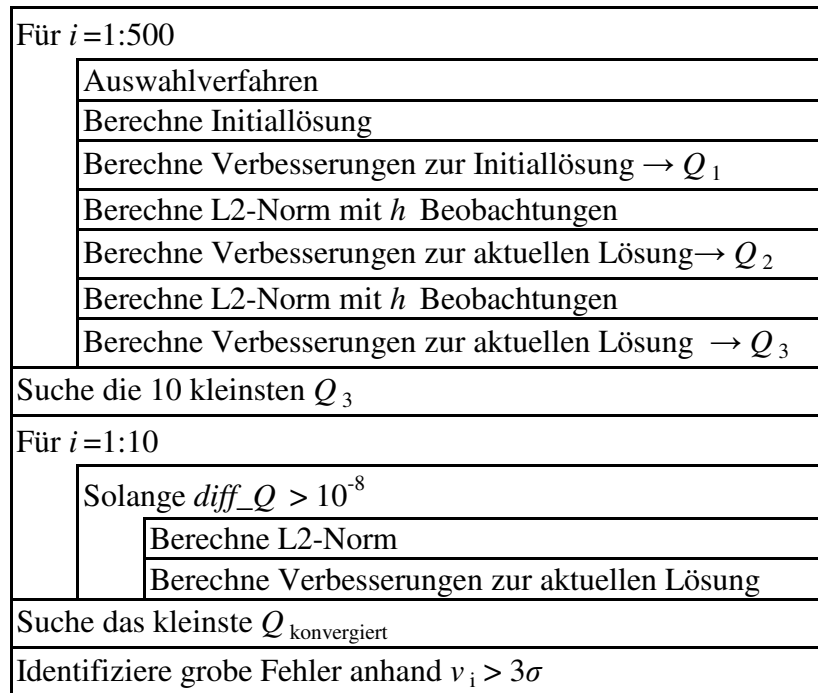


Abbildung 4.6: Struktogramm der Least Trimmed Squares

Es stellt sich die Problematik der Konditionierung der  $\mathbf{A}$ - bzw.  $\mathbf{N}$ -Matrix. Nicht nur bei der Suche eines geeigneten Datensets für die Initiallösung, sondern auch bei der Zusammensetzung der  $h$  Beobachtungen entstehen Konstellationen an Messungen, die eine ausreichende Konditionierung verhindern. Bei Überschreitung einer festgelegten kritischen Konditionszahl von  $cond = 10^{-15}$ , gewonnen aus Erfahrungswerten, wird die aktuelle Iteration abgebrochen und vom Auswahlverfahren an neu gestartet. Diese Vorgehensweise wird auch für RANSAC genutzt.

Nach Ablauf der 500 Iterationen werden aus den archivierten  $Q_3$  die zehn kleinsten Werte ermittelt und über ihre zugehörigen Iterationszahlen auf die  $h$  Beobachtungen, welche in einer Matrix gesammelt wurden, aus dem jeweiligen Durchlauf zugegriffen.

Für jedes der zehn Subsets wird die Berechnung der L2-Norm mit anschließender Bestimmung der Verbesserungen solange wiederholt, bis das Konvergenzkriterium

$$diff\_Q = |Q_i - Q_{i-1}| < 10^{-8} \quad (4.19)$$

eingehalten wird. Die Wahl des Kriteriumwertes geht darauf zurück, dass bei einem größeren Betrag zu früh abgebrochen wird, während ein kleinerer Wert keine zusätzliche Genauigkeitssteigerung bringt.

Als endgültige Lösung ergibt sich jener Parametersatz, dessen  $Q_{\text{konvergiert}}$  minimal ist. Für den Rückschluss auf grobe Fehler wird die Vorgehensweise von RANSAC übernommen.

# Kapitel 5

## Analyse

### 5.1 Ausgangsmodell

Zu Beginn der Arbeit wurde von der Annahme ausgegangen, dass die vorliegenden Messdaten frei von groben Fehlern seien. Die robusten Methoden zeigen bei der Verarbeitung der Ausgangsdaten jedoch das Gegenteil. Welche unterschiedlichen Ergebnisse die einzelnen Verfahren liefern und wie daraus ein bereinigter Datensatz für die weiteren Untersuchungen gewonnen werden kann, zeigen die nachfolgenden Abschnitte.

#### 5.1.1 L1-Norm

Mit der Methode der L1-Norm werden nach 47 Iterationen insgesamt 42 Richtungen und vier Strecken eliminiert, was einem Viertel aller Beobachtungen entspricht. Eine mögliche Erklärung für den hohen Anteil an eliminierten Richtungsbeobachtungen wäre, dass es sich bei den exkludierten Messungen um (grobe) Fehler in verschiedenen Größenordnungen handelt. So ist es denkbar, dass die L1-Norm auf mehrere Fehlerarten anspricht, was durch geeignete Untersuchungen zu bestätigen bleibt.

#### 5.1.2 Biber-Schätzer

Der Biber-Schätzer liefert nach 17 Iterationen eine robuste Lösung. Im *Fehler*-Vektor befinden sich neun unterschiedliche Beobachtungen. Der relative Anteil an groben Fehlern liegt demnach bei knapp 5%. Die Differenz zwischen der Anzahl an Iterationen und der Menge an unterschiedlichen Beobachtungen weist darauf hin, dass einige Beobachtungen mehrmals modifiziert werden.

#### 5.1.3 RANSAC

Bevor RANSAC gestartet werden kann, muss die Anzahl an notwendigen Iterationen festgelegt werden. Geht man von der Richtigkeit der L1-Norm aus, sind 25% der Messungen fehlerbehaftet. Soll mit einer Wahrscheinlichkeit von 99% mindestens ein fehlerfreies Subset gefunden werden, sind gemäß Formel (2.50) 814 089 Iterationen erforderlich.

Bei der ermittelten Durchschnittslaufzeit des Algorithmus von 0,0725s/Iteration, würden über 16 Stunden für die Lösung benötigt. Aufgrund dieser praxisuntauglichen Dauer und der Annahme, dass die L1-Norm auch geringere Abweichungen aufdeckt, wird ein Kompromiss geschlossen

und eine Iterationszahl von  $k = 1000$  festgesetzt. Dies entspricht bei einer Wahrscheinlichkeit von  $z = 0,99$  einem erlaubten Fehleranteil von etwa 12% und einer vertretbaren Rechenzeit von knapp über einer Minute.

Da der Algorithmus auf der zufälligen Auswahl von Beobachtungen basiert und dementsprechend kein eindeutig reproduzierbares Ergebnis vorgesehen ist, weichen die als grobe Fehler identifizierten Messungen von Durchlauf zu Durchlauf voneinander ab. Die Abschätzung der fehlerbehafteten Daten geschieht deshalb aufgrund von mehreren Durchläufen.

So ergeben sich im Schnitt pro Durchlauf 11 grobe Fehler, welche zum Großteil mit einer hohen Häufigkeit auftreten. Das bedeutet, dass dieselben Beobachtungen in 90-100% der Versuche als grobe Fehler identifiziert werden. Dies spricht einerseits für tatsächliche Fehler, die vermutlich auch betragsmäßig am größten sind und andererseits für die Erfolgsrate des Algorithmus. Die endgültige Lösung wird durchschnittlich von 92,18% aller Beobachtungen unterstützt, der relative Anteil an groben Fehlern liegt laut RANSAC demnach bei etwa 8%.

#### 5.1.4 LTS

Bei LTS ist im Unterschied zu RANSAC zwar die Anzahl an Iterationen schon von vornherein festgelegt, allerdings stellt sich auch hier die Problematik der Reproduzierbarkeit. Durch mehrmaliges Starten des Algorithmus wird eine Abschätzung der groben Fehler vorgenommen. Im Schnitt werden pro Durchlauf 27 Beobachtungen als fehlerhaft eingestuft, was umgerechnet 15,08% an Ausreißern bedeutet.

Die am häufigsten gefundenen Fehler treten im Vergleich zur Versuchsanzahl deutlich weniger oft auf. Rund ein Drittel aller Beobachtungen wird zumindest einmal als grober Fehler identifiziert (zum Vergleich: bei RANSAC liegen nur etwa 17% aller Werte mindestens einmal über der  $3\sigma$ -Grenze). Es tritt eine Streuung auf, sodass eine klare Abgrenzung von Kandidaten für grobe Fehler nicht eindeutig möglich ist.

#### 5.1.5 Methode der kleinsten Quadrate

Bei der Auswertung der L2-Norm mit dem ursprünglichen Datensatz besitzt die Varianz der Gewichtseinheit a posteriori (Formel (2.22)) einen markant hohen Wert von  $s_0^2 = 7,8735$  und ist im Vergleich zur a priori festgelegten Varianz  $\sigma_0^2 = 1$  sehr hoch. Niemeier (2008, S.170) gibt für eine signifikante Abweichung der beiden Werte folgende Ursachen an:

- Der Varianzfaktor  $\sigma_0^2$  ist nicht an die tatsächlichen Messbedingungen angepasst.
- Das funktionale Modell ist fehlerhaft.
- Das stochastische Modell trifft nicht zu, was unter anderem durch den Einfluss von groben oder systematischen Fehlern bedingt wird.

Da die Annahme von  $\sigma_0^2 = 1$  aus der einschlägigen Literatur übernommen wurde und auch ein Fehler im funktionalen Modell unwahrscheinlich ist, wird der Fokus auf das stochastische Modell gelegt.

Eine Möglichkeit der Anpassung ist durch die Überprüfung der gewählten  $\sigma_{l_i}$  gegeben. Dem Ausgleichsprotokoll im Technischen Bericht der Feldübung ist zu entnehmen, dass für alle Richtungsbeobachtungen einheitlich eine Standardabweichung von  $\sigma_r = 0,5\text{mgon}$  und für alle Streckenbeobachtungen  $\sigma_s = 5\text{mm} + 2\text{ppm}$  angenommen wurde. Die Genauigkeiten wurden verringert, da die Messungen nicht von einem geübten und erfahrenen Personal, sondern von Studenten durchgeführt wurden und weil das verwendete Messinstrumentarium keine Einrichtung für das automatische Zentrieren auf den Zielpunkt besitzt. Dadurch wirken sich Anzielfehler in vollem Ausmaß auf die Qualität der Beobachtungsdaten aus. Mit dieser Änderung in  $\Sigma_{LL}$  reduziert sich die Gewichtseinheit a posteriori zunächst auf  $s_0^2 = 4,9975$ . Der dennoch zu hohe Wert deutet auf den Einfluss von groben und/oder systematischen Fehlern hin, was durch die Ergebnisse der robusten Schätzer bestätigt werden kann. Führt man den bereinigten Datensatz (siehe Kapitel 5.1.6) in die Ausgleichung ein, verringert sich die Größe auf  $s_0^2 = 0,2506$ .

Um alle teilnehmenden Verfahren unter gleichen Voraussetzungen vergleichen zu können, werden die Genauigkeiten aus Angst et al. (1994) verwendet.

### 5.1.6 Zusammenfassung und Vergleich

Einen guten Überblick über die Ergebnisse verschafft Tabelle 5.1, in der die wichtigsten Kenngrößen pro Verfahren zusammengestellt sind.

Tabelle 5.1: Vergleich der robusten Methoden

Kenngröße	L1	Biber	RANSAC	LTS
Iterationen	47	17	1000	500
Rechenzeit [s]	11	0,5	73	83
Fehler (absolut)	46	9	11	27
Fehler (relativ)	25,70%	5,03%	7,82%	15,08%

Es fällt auf, dass die Rechenzeiten von RANSAC und LTS vergleichsweise hoch sind. Das liegt daran, dass in jedem Iterationsschritt eine ausreichend konditionierte  $\mathbf{A}$ -Matrix gefunden werden muss, um eine eindeutige Lösung berechnen zu können. Zwar werden Einschränkungen bezüglich der Auswahl an Beobachtungen vorgegeben, diese können den Vorgang aber nur bedingt beschleunigen. Aus Erfahrungswerten muss pro Iteration mit knapp 1,5 ungenügend konditionierten Gleichungssystemen gerechnet werden. Hierbei sei erwähnt, dass in der vorliegenden Arbeit der Fokus nicht auf einer optimalen Implementierung der Algorithmen liegt, sondern primär die Erfüllung der Ansprüche im Vordergrund steht. Eine Effizienzsteigerung und damit einhergehende Verkürzung der Rechenzeit ist deshalb noch möglich.

Tabelle 5.2 zeigt jene detektierten Fehler, die von zumindest zwei robusten Verfahren gefunden werden. Die Benennung erfolgt analog zu den Beobachtungsnummern aus dem Inputfile

„richtungen\_a.txt“. Die L1-Norm eliminiert zusätzlich zu den aufgelisteten Beobachtungen noch 32 weitere, der Biber-Schätzer eine und LTS noch 14.

Tabelle 5.2: Übereinstimmende Fehler

L1	Biber	RANSAC	LTS
25	25	25	25
37	37	37	37
92	92	92	92
96	96	96	96
104	104	104	104
108	108	108	108
49		49	49
32			32
33			33
44			44
54			54
59			59
88			88
	95	95	
107		107	
	117	117	

Die L1-Norm gibt den größten Anteil an fehlerbehafteten Messungen an und liefert deshalb auch als einziges Verfahren einen bereinigten Datensatz, dem keines der anderen robusten Methoden entgegensteht. Aus diesem Grund wird der um ein Viertel dezimierte Datensatz („richtungen\_acorr.txt“ und „sGK\_acorr.txt“, siehe Anhang A) der L1-Norm mit 133 statt 179 Beobachtungen als Ausgangsbasis für die weiteren Untersuchungen verwendet. Die Anzahl der Unbekannten bleibt bestehen, die Gesamtredundanz ändert sich auf:  $r = 133 - 42 + 3 = 94$ .

Die ausgeglichenen Unbekannten jedes der Verfahren sind der Vollständigkeit halber in Tabelle 5.3 gegenübergestellt und basieren auf dem bereinigten Datensatz. Sie dienen von nun an als Referenzwerte für die zukünftigen Untersuchungen. Die Ergebnisse von RANSAC, dem Biber-Schätzer und der L2-Norm sind ident, wenn mit fehlerfreien Daten gerechnet wird. Bei der L1-Norm kommen Abweichungen aus dem Grund zustande, da sich die beiden Verfahren in der Gewichtung der Beobachtungen voneinander unterscheiden. Die Abweichung der LTS-Lösung erklärt sich durch die Nutzung von nur  $h$  Beobachtungen zur Berechnung des finalen Parametersatzes.

Die Abweichungen der Parameter von der fehlerfreien L2-Norm bei Verwendung des Ausgangsdansatzes sind in Tabelle 5.4 zusammengefasst. Diese sind gerade deswegen von Interesse, da robuste Methoden unter anderem das Ziel verfolgen, ein Ergebnis möglichst nahe an jenem der Methode der kleinsten Quadrate zu erreichen. Für die Angaben von RANSAC und LTS, die ohne entsprechende Modifikationen im Programmcode kein exakt reproduzierbares Ergebnis liefern, werden die Ergebnisse eines einmal komplett durchlaufenen Algorithmus herangezogen. Im Schnitt (Mittelwert und Median) weichen die einzelnen Ergebnisse in den Koordinaten um nur

Tabelle 5.3: Geschätzte Parameter mit fehlerfreiem Datensatz

Parameter	L2	L1	LTS
Y11	-20629,734	-20629,735	-20629,734
X11	5296245,822	5296245,817	5296245,819
Y100	-18280,869	-18280,873	-18280,872
X100	5298067,330	5298067,331	5298067,331
Y114	-21239,275	-21239,284	-21239,281
X114	5299087,440	5299087,432	5299087,434
Y150	-19197,252	-19197,252	-19197,251
X150	5296895,028	5296895,033	5296895,031
Y151	-22728,037	-22728,035	-22728,036
X151	5297191,752	5297191,755	5297191,752
Y152	-18958,900	-18958,895	-18958,890
X152	5301272,214	5301272,217	5301272,217
Y230	-18056,184	-18056,186	-18056,185
X230	5296909,961	5296909,960	5296909,962
Y233	-18185,359	-18185,350	-18185,361
X233	5299063,884	5299063,884	5299063,885
<hr/>			
O <sub>11,1</sub>	399,9975	399,9974	399,9975
O <sub>11,2</sub>	399,9954	399,9955	399,9956
O <sub>11,3</sub>	399,9959	399,9956	399,9957
O <sub>11,4</sub>	399,9960	399,9959	399,9958
O <sub>100,1</sub>	399,7595	399,7596	399,7594
O <sub>100,2</sub>	399,7593	399,7593	399,7595
O <sub>100,3</sub>	399,7609	399,7606	399,7606
O <sub>100,4</sub>	399,7604	399,7603	399,7604
O <sub>150,1</sub>	83,5599	83,5600	83,5600
O <sub>150,2</sub>	83,5564	83,5563	83,5563
O <sub>150,3</sub>	83,5553	83,5550	83,5550
O <sub>150,4</sub>	83,5548	83,5548	83,5545
O <sub>150,5</sub>	182,4873	182,4873	182,4872
O <sub>150,6</sub>	182,4874	182,4874	182,4873
O <sub>150,7</sub>	182,4873	182,4872	182,4873
O <sub>150,8</sub>	182,4873	182,4874	182,4874
O <sub>230,1</sub>	399,9411	399,9415	399,9411
O <sub>230,2</sub>	399,9523	399,9527	399,9523
O <sub>230,3</sub>	399,9494	399,9498	399,9494
O <sub>230,4</sub>	399,9485	399,9489	399,9485
O <sub>230,5</sub>	399,9548	399,9552	399,9548
O <sub>230,6</sub>	399,9569	399,9573	399,9569
O <sub>230,7</sub>	139,5811	139,5815	139,5811
O <sub>230,8</sub>	139,5809	139,5813	139,5809
O <sub>233,1</sub>	65,0083	65,0082	65,0079
O <sub>233,2</sub>	65,0079	65,0079	65,0079

Tabelle 5.4: Abweichungen der robusten Verfahren von der L2-Norm

	L1	Biber	RANSAC	LTS
<b>Koordinaten [m]</b>				
Mittelwert	0,007	0,009	0,008	0,013
Median	0,005	0,006	0,004	0,011
Minimum	0,001	0,000	0,000	0,000
Maximum	0,028	0,044	0,045	0,011
<b>Orientierung [gon]</b>				
Mittelwert	0,0006	0,0008	0,0007	0,0006
Median	0,0003	0,0005	0,0004	0,0003
Minimum	0,0000	0,0000	0,0000	0,0000
Maximum	0,0028	0,0025	0,0024	0,0027

wenige Millimeter bis knapp über einen Zentimeter von der wahren Lösung ab. Bedenkt man, dass jedes Verfahren verschiedene Beobachtungen als grob fehlerhaft einstuft und daraus dennoch nur Abweichungen im Bereich um einen Zentimeter resultieren, spricht dies für die Anwendbarkeit aller vier robusten Methoden.

Die maximale Differenz in den Koordinaten in der Größe von mehreren Zentimetern tritt bei drei der vier Verfahren am selben Punkt und in derselben Koordinate auf (Y-Koordinate von Punkt 233).

Für die Orientierungsunbekannten bewegen sich die Abweichungen über alle Verfahren und alle Kenngrößen gesehen, in der gleichen Größenordnung. Der Median ergibt Abweichungen, welche zur Gänze im Bereich oder sogar unter der Messgenauigkeit von 0,5mgon liegen.

Durch die recht ähnlichen Abweichungen kann kein Verfahren als klarer Favorit hervorgehoben werden.

## 5.2 Modell mit einem groben Fehler

Die erste Versuchsreihe besteht darin, nacheinander alle Beobachtungen einmal mit einem groben Fehler zu besetzen und die Auswirkungen auf das Netz bzw. die Parameter zu dokumentieren. Zusätzlich wird kontrolliert, ob die Algorithmen in der Lage sind, die fehlerhafte Messung zu identifizieren.

In den folgenden Abschnitten wird primär auf die Auswirkungen der groben Fehler auf die geschätzten Koordinaten eingegangen, da sie in der Praxis von größerer Bedeutung sind als die Orientierungsunbekannten.

Die Beurteilung der Verfahren erfolgt in Bezug zur jeweiligen Referenzlösung, die am Ende des vorigen Abschnitts bereits angesprochen wurde und sich aus Anwendung des fehlerfreien Datensatzes auf jede Methode ergibt.

Die Größe des simulierten groben Fehlers leitet sich vom minimal detektierbaren Ausreißer ab. Dieser Wert ist individuell für jede einzelne Beobachtung mittels Formel (2.27) zu berechnen. Durch die Verzehnfachung des MDB werden grobe Fehler aus der Praxis simuliert, wie zum Beispiel unbemerkte und unabsichtliche Verdrehung des Geräts vor dem Auslösen der Messung, Anzielfehler, Punktverwechslung oder Ähnliches. Mit Ausnahme von LTS (zu hohe Rechenzeit) wird diese Vorgehensweise auch mit groben Fehlern von 5 MDB wiederholt.

Zur Vorbereitung des Datenmaterials werden RANSAC und LTS einmal mit dem bereinigten Datensatz durchlaufen und alle Beobachtungen, die zum Berechnen der 1000 bzw. 500 eindeutigen Lösungen benötigt werden, gespeichert. Dies ermöglicht ein reproduzierbares Ergebnis und einen sinnvollen Vergleich.

### 5.2.1 L1-Norm

In rund 89% der Fälle (entspricht 118 Beobachtungen) kann der grobe Fehler in der Größenordnung von 10 MDB erfolgreich aufgedeckt werden. Die Fehlerquote von 11% kommt durch 15 Fehldetektionen zustande. Elf der 15 Beobachtungen sind Richtungen, die in Sätzen mit zwei gleichartigen Beobachtungen auftreten und bei denen die aus Kapitel 4.4 bekannte Problematik auftritt. Für den Algorithmus per se ist es nicht möglich, diesbezüglich eine richtige Entscheidung zu treffen, da die Redundanz in dem betreffenden Fall zu gering ist. Das zeigt sich auch daran, dass genau in der Hälfte dieser Fälle die fehlerfreie Beobachtung im betreffenden Satz als falsch detektiert wird, während es in der anderen Hälfte zu einer richtigen Auflösung kommt. Die statistische 50:50 Chance einer korrekten Identifikation spiegelt sich darin wider.

Abseits dieser Schwierigkeit treten weitere Fehldetektionen auf, die keiner Gesetzmäßigkeit zu unterliegen scheinen: In vier Fällen wird der Fehler im ersten Schritt gefunden, durch die erneute Iteration kommt es jedoch zu einer Abgewichtung einer weiteren (fehlerfreien) Messung. Bei näherer Betrachtung des Gewichtsverlaufs ist erkennbar, dass ihr Gewicht nur knapp unter dem Grenzwert von 1000 liegt. Im Vergleich dazu nimmt das Gewicht einer tatsächlich verfälschten Beobachtung einen Wert  $< 100$  an. Zudem wird die Beobachtung erst nach einigen Iterationen abgewichtet, während eine grob fehlerhafte Messung zumeist bereits ab der zweiten Iteration reduziert wird. Diese beiden Merkmale lassen auf eine Fehldetektion schließen und sind möglicherweise ein weiterer Hinweis darauf, dass die L1-Norm auch auf nur leicht vom Modell abweichende Daten reagiert oder das Abbruchkriterium für manche Fälle zu sensibel ist. Daraus wird die Erkenntnis gewonnen, dass es ratsam ist, den Gewichtsverlauf zu überprüfen, bevor das Vorliegen eines groben Fehlers bestätigt wird. Auf diese Art können möglicherweise unnötig eliminierte Messungen vermieden und eine ausreichend hohe Redundanz beibehalten werden.

In den 89% aller Beobachtungen, in denen die L1-Norm den eingeführten Fehler richtig detektieren kann, ergeben sich in Bezug zur fehlerfreien L1-Lösung Abweichungen in den Koordinaten bis zu 2,1 Millimeter. Die Ursache dafür liegt zum einen im Ausschluss der betroffenen Beobachtung, wodurch ein Bestimmungsstück im Netz fehlt, was wiederum Einfluss auf die Kontrolliertheit und Bestimmbarkeit der Parameter nimmt. Zum anderen wirkt sich der Aufbau der Gewichtsmatrix,



der für jede Zusammenstellung an noch verwendeten Messungen unterschiedlich ist, auf die Parameterschätzung aus.

Die Abweichungen bei einer Fehldetektion nehmen Werte bis 1,5mm an, sind also kaum von jenen einer richtigen Identifikation zu unterscheiden. Die Abweichungen entstehen, wenn zusätzlich zu einer zu Recht eliminierten Beobachtung eine zweite ausgeschlossen wird. Dadurch geht Information über das Netz verloren. In Kombination mit der individuellen Gewichtsmatrix ergeben sich Abweichungen, die in diesem Fall bis zu 1,5mm groß werden.

Fehldetektionen in Richtungssätzen mit nur zwei Beobachtungen wirken sich nicht auf die geschätzten Koordinaten aus. Wird in einem solchen Fall eine Richtung aufgrund des minimalen Gewichts (fälschlicherweise) eliminiert, besteht keine Diskrepanz mehr zwischen den Orientierungsunbekannten und der grobe Fehler wird zur Gänze durch diesen Parameter absorbiert, die Koordinaten bleiben fehlerfrei. Dies ist auch daran zu beobachten, dass die Abweichung in der betreffenden Orientierungsunbekannte genau dem Wert des eingeführten Fehlers entspricht.

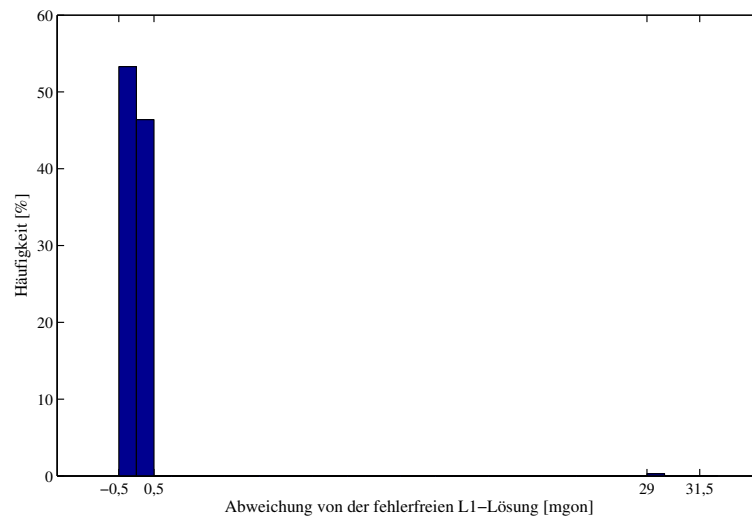


Abbildung 5.1: Abweichungen in den Orientierungsunbekannten, bezogen auf die fehlerfreie L1-Lösung

Werden Fehler mit 5 MDB eingeführt, treten dieselben Effekte auf. Ein Unterschied ergibt sich in der Anzahl der Fehldetektionen, die sich auf 13 beläuft. In sieben Fällen löst ein Redundanzanteil von 0,5 eine falsche Zuordnung des Fehlers aus. In den restlichen sechs Fällen wird nach der richtigen Lokalisierung des Fehlers eine weitere Beobachtung fälschlicherweise als grober Fehler ausgewiesen. Daraus wird der Schluss gezogen, dass die Größe des (eingeführten) groben Fehlers keine signifikante Auswirkung auf die Aufdeckbarkeit hat. Die angeführten Abbildungen beziehen sich im Übrigen auf die Resultate mit einem groben Fehler in der Größe von 10 MDB.

Das Histogramm in Abbildung 5.2 zeigt die Häufigkeit der Abweichungen aller Koordinaten über alle Versuche, die beim Vergleich mit der fehlerfreien L1-Lösung entstehen. Mehr als 90% aller Abweichungen sind vernachlässigbar klein, da sie Werte unter einen halben Millimeter annehmen.

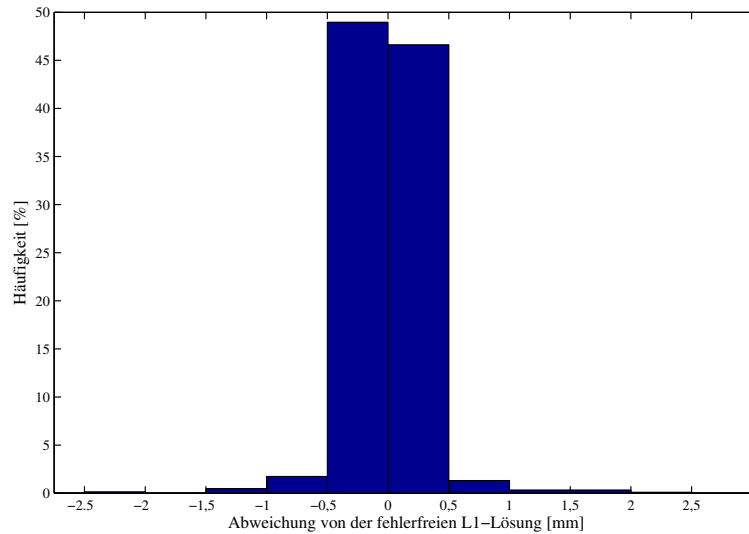


Abbildung 5.2: Abweichungen in den Koordinaten, bezogen auf die fehlerfreie L1-Lösung

Trotz einer detaillierter Datenanalyse, kann kein klarer Zusammenhang zwischen dem Ort der verfälschten Beobachtung und der Änderung in den Koordinaten gefunden werden.

Die Auswirkungen in den Orientierungsunbekannten sind im Histogramm in Abbildung 5.1 dargestellt. 99% aller Abweichungen befinden sich in einem Bereich kleiner als 0,5mgon und liegen unterhalb der Messgenauigkeit. Treten Fehldetektionen auf, ergeben diese Abweichungen in der Größe der eingeführten Fehler. Im Histogramm scheinen sie als Ausreißer auf der rechten Seite auf.

Der Vergleich der 133 Parametersätze mit der Lösung der Methode der kleinsten Quadrate ergibt Abbildung 5.3. Die Abweichungen sind um das Zehnfache größer, mit max. 11mm aber ausreichend klein um beispielsweise die Lagegenauigkeit des vorliegenden Netzes einzuhalten. Auffällig ist die unregelmäßige Verteilung der Abweichungen. Die betragsmäßig größeren Werte sind hauptsächlich darauf zurückzuführen, dass zu den oben besprochenen Effekten noch hinzukommt, dass sich bereits die fehlerfreie L1-Lösung teilweise um maximal neun Millimeter von der L2-Lösung unterscheidet. Durch Überlagerung mit den Abweichungen aus Abbildung 5.2 werden die Unterschiede leicht verstärkt.

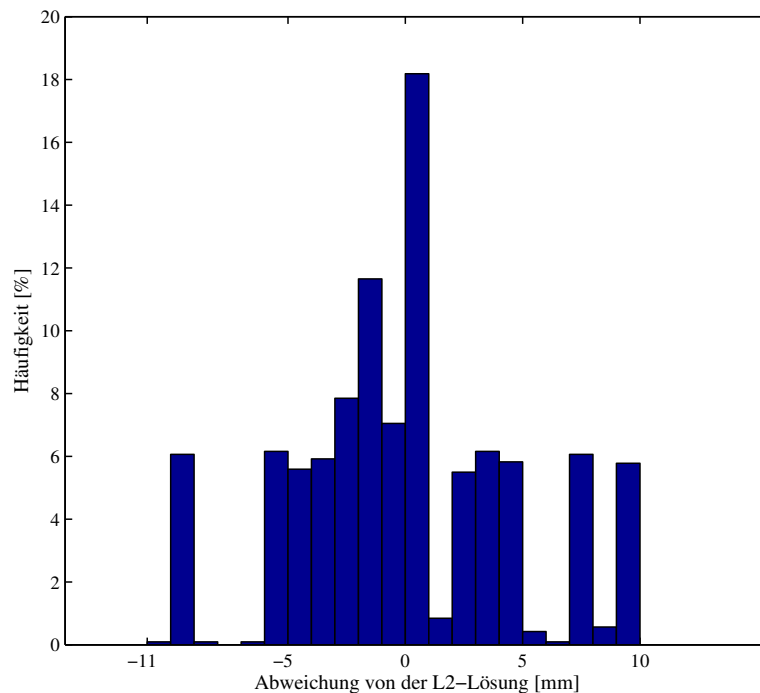


Abbildung 5.3: Abweichungen in den Koordinaten, bezogen auf L2

Die größten Abweichungen werden durch jene Beobachtungen verursacht, bei denen der grobe Fehler nicht richtig zugeordnet werden kann. Der Vergleich der Orientierungsunbekannten führt zu etwa derselben Häufigkeitsverteilung wie in Abbildung 5.1.

Zusammengefasst liefert die L1-Norm bei Vorliegen eines einzigen groben Fehlers im gesamten Netz sehr gute Ergebnisse, die nur minimal von ihrer bestmöglichen Lösung differieren. Dies spricht für die Effizienz des Verfahrens. Die Abweichungen zur L2-Lösung im 1cm-Bereich sind für viele Anwendungen in der Praxis ausreichend gering.

Die Aufdeckbarkeit eines groben Fehlers ist zufriedenstellend. Eine Kontrolle des Gewichtsverlaufs durch den Anwender ist anzuraten, da er Aufschluss über möglicherweise falsche Detektionen gibt.

### 5.2.2 Biber-Schätzer

Die Anzahl der misslungenen Fehlerlokalisierungen beläuft sich beim Biber-Schätzer auf 14. Diese betreffen ausschließlich Richtungsmessungen, die sich in einem Richtungssatz mit zwei gleichen Visuren befinden. Ihre Redundanzanteile betragen 0,5 und verhindern - wie mehrmals besprochen - eine richtige Auflösung. Für diese Fälle ergeben sich in den Koordinaten keine Abweichungen zur fehlerfreien Lösung, da der Fehler alleine von der zugehörigen Orientierungsunbekannte absorbiert wird. Dabei entstehen ähnlich große Werte (28,1mgon) wie bei der L1-Norm.

Wird der grobe Fehler richtig lokalisiert, resultieren aus der angebrachten Modifikation Abweichungen in den Koordinaten bis zu 1,2cm (siehe auch Abbildung 5.4, in der die Häufigkeiten aller Abweichungen dargestellt sind).

Die Maximalwerte für die Orientierungsunbekannten belaufen sich auf 1,1-1,3mgon. Hervorgerufen werden diese durch Richtungsmessungen, die zwar in Sätzen mit nur zwei Beobachtungen zu finden sind, deren Redundanzanteile aber nicht 0,5 betragen. Weiters zählen auch Richtungen mit  $r_i = 0,5$  dazu, bei denen der Fehler korrekt detektiert wird. Wie an den Maximalwerten erkennbar ist, resultieren daraus Abweichungen mit einem geringeren Betrag als bei einer Fehldetektion.

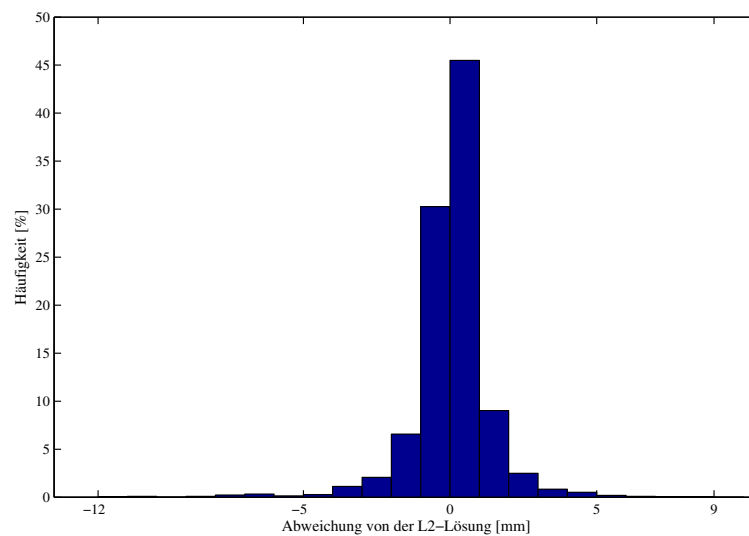


Abbildung 5.4: Abweichungen in den Koordinaten, bezogen auf L2 (Biber-Schätzer)

Eine detaillierte Analyse unabhängig vom Erfolg der Fehlerdetektion ergibt, dass 76% aller Abweichungen kleiner als einen Millimeter sind und 99,9% unter einem Zentimeter bleiben.

Ein systematischer Zusammenhang zwischen modifizierter Beobachtung und Auswirkung auf die Koordinaten ist aus den Daten nicht erkennbar.

Für Fehler mit einer Größe von 5 MDB ergeben sich hinsichtlich der Aufdeckbarkeit und Art der Auswirkung auf die Unbekannten die gleichen Resultate. Lediglich die Größe der Abweichungen ist dementsprechend angepasst und niedriger.

Für das vorliegende Netz kann der Biber-Schätzer sowohl im Bereich der Fehlerdetektion, als auch hinsichtlich der Auswirkungen auf die Parameter Erfolge vorweisen. Die maximalen Abweichungen in den Koordinaten wären bei einem entsprechendem Anwendungsfall für die gesetzlich vorgeschriebene mittlere Punktlagegenauigkeit von 4cm für Messpunkte und 5cm für Grenzpunkte ausreichend (§6 VermV 2010).

Lässt man jene Fehldetektionen außer Acht, welche redundanzbedingt nicht lösbar sind, kann der Biber-Schätzer eine Erfolgsquote von 100% aufweisen.

### 5.2.3 RANSAC

Mit nur neun falschen Lokalisierungen eines groben Fehlers, schneidet RANSAC in dieser Hinsicht am besten ab. Wenig überraschend ist, dass Probleme nur bei Richtungsbeobachtungen mit dem Redundanzanteil von 0,5 auftreten.

In jenen Fällen, in denen RANSAC den Fehler richtig lokalisiert, berechnen sich Abweichungen bis zu 4mm. Der Grund hierfür ist, dass RANSAC die fehlerbehaftete Beobachtung in weiterer Folge nicht mehr für die Berechnung der endgültigen Lösung miteinbezieht. Dadurch verringert sich die Gesamtredundanz des Systems und die erhaltene Lösung kann nicht mehr mit der L2-Lösung übereinstimmen.

Sämtliche Abweichungen, die durch eine Fehldetektion entstehen, sind weitaus kleiner als ein Millimeter. Zurückzuführen ist dieses Verhalten auf dieselben Ursachen wie sie von der L1-Norm und dem Biber-Schätzer bekannt sind.

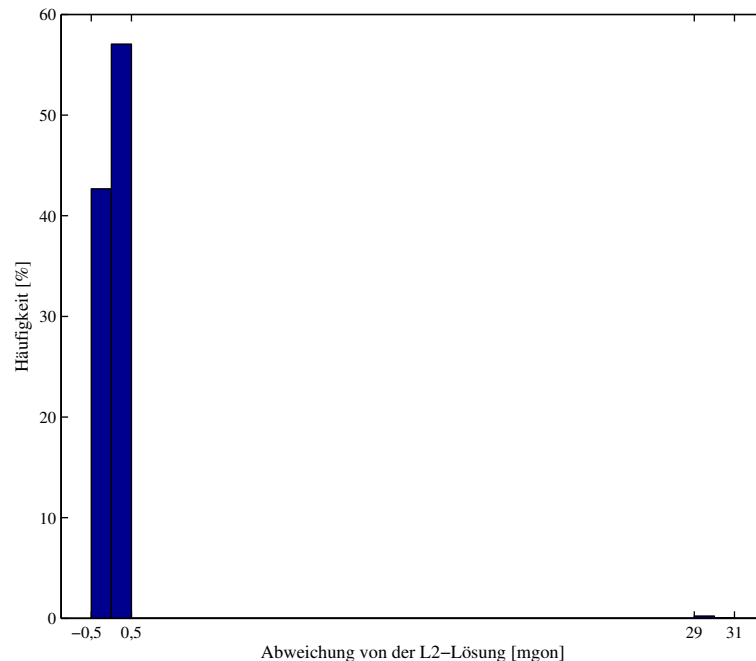


Abbildung 5.5: Abweichungen in den Orientierungsunbekannten, bezogen auf L2 (RANSAC)

Die Einflüsse auf die Orientierungsunbekannten zeigen das gleiche Verhalten wie bei den vorigen Verfahren. Bis auf wenige Ausreißer, die aus demselben Prinzip wie bei der L1-Norm entstehen, liegen die Abweichungen fast gänzlich im Bereich von unter 0,5mgon (siehe Abbildung 5.5).

Fehler von 5 MDB folgen derselben Gesetzmäßigkeit und unterscheiden sich nur in der Größe der Abweichungen von den für 10 MDB gültigen Erkenntnissen.

Mit maximalen Abweichungen im niedrigen Millimeterbereich und einer Fehldetektionsrate von 7% weist RANSAC in beiden zu untersuchenden Kategorien ein erfolgreiches Resultat vor.

#### 5.2.4 LTS

Durch die Verwendung des fehlerfreien Datensatzes kann, abgeleitet von dem für RANSAC festgelegten Fehleranteil von 10%, die Trimmingkonstante auf  $h = 0,9 \cdot n = 120$  erhöht werden. Diese Änderung bewirkt eine geringere Streuung in den geschätzten Parametern und bringt eine Annäherung an die Lösung der L2-Norm, da mehr Beobachtungen in die Berechnung einfließen können.

Sobald ein grober Fehler an eine beliebige Messung angebracht wird, kommt es bei der Bestimmung der  $h$  Beobachtungen zu einer Kombination an Bestimmungsstücken, die eine unzureichende Konditionierung des Systems zur Folge hat und den Algorithmus abbrechen lässt. Hebt man die eingangs definierten Einschränkungen (vorgegebene Auswahl an Beobachtungen für die Initiaillösung) auf, funktioniert die Berechnung wie erwartet. Aufgrund des ständig wechselnden Ausgangspunktes pro Durchlauf ist mit einer Streuung in den Ergebnissen, sowie mit Unterschieden bei der Detektion des groben Fehlers zu rechnen. Dies erschwert, wie auch schon bei der Grobfehlersuche im Ausgangsmodell, eine verlässliche Bewertung des Verfahrens. Für den Vergleich mit der verfahrenseigenen Optimallösung wird das Ergebnis mit der fixierten Beobachtungsauswahl herangezogen.

Aufgrund der hohen durchschnittlichen Laufzeit des Algorithmus von über einer Minute, wird LTS für jeden groben Fehler nur einmal gestartet und die Größenordnung des Fehlers bei 10 MDB belassen.

Insgesamt kann LTS in 20 von 133 Fällen den groben Fehler nicht richtig lokalisieren. Jeder dieser 20 Versuche betrifft eine Richtungsmessung. Erneut sind unter anderem Richtungsbeobachtungen mit  $r_i = 0,5$  betroffen. Wird zu Beginn die fehlerfreie der beiden Messungen für die Bildung der eindeutigen Lösung ausgewählt, besteht die fehlerhafte am Ende des Algorithmus die  $3\sigma$ -Abfrage nicht und die Detektion ist erfolgreich. Im umgekehrten Fall basiert die Schätzung der Parameter auf der falschen Messung, dies bleibt unbemerkt, da keine zusätzliche Richtung im Satz vorhanden ist, die dies aufdecken könnte. Am Ende der Berechnung weicht die eigentlich fehlerfreie Richtung zu stark von der Gesamtlösung ab und eine Fehldetektion ist die Folge. Überdies sind auch Richtungen, die sich in Sätzen mit insgesamt drei Beobachtungen befinden, betroffen. Wird der Algorithmus für einen Fehler in diesen Richtungen mehrmals gestartet, führt dies in manchen Fällen zu einer richtigen Detektion. Dieses Verhalten ist auf die Zusammenstellung der ausgewählten Beobachtungen zurückzuführen und bei LTS unvermeidbar.

Um die Auswirkungen eines groben Fehlers auf die geschätzten Parameter bewerten zu können, werden diese wiederum mit der LTS-Referenzlösung verglichen. Wird der Fehler richtig identifiziert, entstehen Abweichungen bis zu 5mm, bei einer Fehldetektion nehmen sie maximale Beträge von 2mm an. Zurückzuführen ist dies auf die Verwendung  $h$  unterschiedlicher Beobachtungen für die Berechnung der endgültigen Lösung.

Die Differenz zwischen den Maximalabweichungen bei erfolgreicher und nicht erfolgreicher Detektion unterliegt derselben Ursache. Jede fehlerbesetzte Beobachtung beeinflusst individuell die übrigen Messungen, Abweichungen sind daher teilweise größer und teilweise kleiner. Nachdem die

Gruppe der Fehldetektionen einen geringeren Umfang besitzt, wird nicht derselbe Wertebereich an Abweichungen abgedeckt.

Beim Vergleich mit der L2-Lösung tritt der gleiche Überlagerungseffekt wie bei der L1-Norm auf und die Abweichungen nehmen zu. Die LTS-Referenzlösung weicht gegenüber der L2-Lösung um wenige Millimeter bis höchstens einen Zentimeter ab. Der maximale Fehler in den Koordinaten beträgt 12mm (siehe Abb. 5.6), weiters ist ein unregelmäßiger Verlauf der Häufigkeiten zu beobachten.

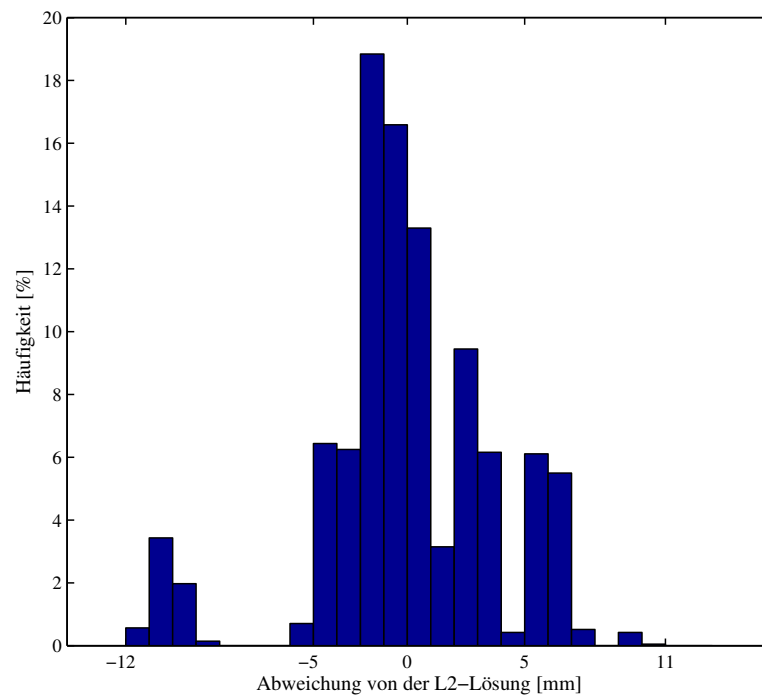


Abbildung 5.6: Abweichungen in den Koordinaten, bezogen auf L2 (LTS)

Für die Orientierungsunbekannten sei erwähnt, dass sie dem Verhalten der übrigen Verfahren folgen. Bis auf einige wenige Ausreißer im Bereich von 25 bis 31mgon, befinden sich 98-99% aller Abweichungen sowohl beim relativen Vergleich zur LTS-Lösung, als auch beim absoluten zur L2-Lösung im Intervall von  $\pm 0,5$ mgon.

Alles in allem liefert LTS bei Anbringen eines groben Fehlers ausreichend gute Ergebnisse für Anwendungen, die Zentimetergenauigkeit erfordern. Die Detektion ist meist verlässlich, in einigen Fällen reicht allerdings ein einmaliges Starten des Algorithmus nicht aus, um den Fehler korrekt zu identifizieren.

### 5.2.5 Conclusio

Durch die Untersuchungen mit einem groben Fehler lassen sich die Arbeitsweisen der einzelnen Verfahren besser verstehen und die Unterschiede und verfahrensspezifischen Eigenheiten werden ersichtlich. Die Entstehung der Abweichungen sollte unbedingt hinterfragt werden, da andernfalls die robusten Methoden falsch beurteilt werden.

Die Vergleiche für die L1-Norm und LTS müssen algorithmusbedingt unter Beachtung mehrerer Aspekte durchgeführt werden und zwar relativ zur jeweiligen Referenzlösung und absolut zur L2-Lösung.

In Tabelle 5.5 sind die wichtigsten Ergebnisse dieses Abschnitts zusammengefasst. Die absoluten Abweichungen beziehen sich auf die Lösung der L2-Norm, die relativen auf die jeweilige Referenzlösung des Verfahrens. Um den Erfolg der robusten Methoden besser einordnen zu können, sind in der letzten Spalte jene maximalen Abweichungen angeführt, die sich ergeben, wenn die groben Fehler nach derselben Vorgehensweise angebracht werden und dabei die Methode der kleinsten Quadrate zur Parameterschätzung eingesetzt wird.

Tabelle 5.5: Eckdaten der robusten Verfahren

	L1	Biber	RANSAC	LTS	L2
Fehldetektionen	15 (4)	14 (0)	9 (0)	20 (5)	-
max. Abweichung in Koordinaten [m]					
absolut	0,009	0,012	0,004	0,012	0,145
relativ	0,002	-	-	0,007	-
max. Abweichung in Orientierungsunbekannten [mgon]					
absolut	0,0309	0,0281	0,0314	0,0308	0,0160
relativ	0,0311	-	-	0,0309	-

Urteilt man rein auf Basis der Erfolgsquote bezüglich der Aufdeckung der groben Fehler, ist RANSAC als Favorit anzusehen. Eine Verbesserung der Statistik ist insofern möglich, da die Detektion eines groben Fehlers in Satzmessungen mit nur zwei Beobachtungen grundsätzlich nicht möglich ist. Zählt man jene Fälle nicht als Misserfolge, kann die Anzahl der Fehldetektionen beim Biber-Schätzer und RANSAC sogar auf null korrigiert werden, für die L1-Norm wird ein Wert von vier erreicht (eingeklammerte Zahl in Tabelle 5.5). Des Weiteren gilt: Wird der Benutzer in den Entscheidungsprozess miteinbezogen, kann er anhand des Gewichtsverlaufs Fehldetektionen erkennen und die Anzahl bei der L1-Norm ebenfalls auf null verringern.

Die absoluten Abweichungen in den Koordinaten betreffend, hebt sich RANSAC wiederum von den übrigen Methoden ab. Unabhängig davon, welche Beobachtung verfälscht wird, kann der Algorithmus überzeugen. Der Biber-Schätzer und LTS erlauben eine Schätzung mit maximalen absoluten Abweichungen von etwa einem Zentimeter. Die gesetzlichen Voraussetzungen für amtliche Vermessungen sind damit erfüllt. Die L1-Norm übertrifft die Leistung dieser beiden



Verfahren und zeigt absolute Maximalabweichungen knapp unter einem Zentimeter. Die relativen Abweichungen von LTS und L1 liegen im Bereich der absoluten Werte von RANSAC, für die L1-Norm sogar darunter.

In Relation zu den Abweichungen, die bei der L2-Norm entstehen, werden die Ansprüche an die robuste Schätzung erfüllt und die Effizienz und der Erfolg aller vier Verfahren kann bestätigt werden.

Die Orientierungsunbekannten werden von allen vier Methoden in etwa gleich genau bestimmt, wobei auffällt, dass die maximale Abweichung aus der L2-Norm wesentlich geringer ausfällt. Dieses Verhalten ist darauf zurückzuführen, dass bei den robusten Schätzern der grobe Fehler oft zur Gänze von den Orientierungsunbekannten absorbiert wird, während bei der L2-Norm ein Verschmierungseffekt auftritt, der sich in einem geringeren Abweichungswert äußert.

### 5.3 Modell mit Fehler und reduzierter Redundanz

In diesem Abschnitt wird sowohl die Anzahl der eingeführten groben Fehler erhöht, als auch die Gesamtredundanz sukzessive reduziert. Bei der Reduktion wird darauf geachtet, dass die verringerten Messdaten in einer Zusammenstellung zurückbleiben, die realen Beobachtungsplänen entspricht.

Auf jedem Standpunkt wird ausgehend vom Ausgangsdatensatz mit 133 Beobachtungen, die Anzahl der Richtungssätze halbiert, sodass jeweils nur mehr zwei oder vier Satzmessungen vorliegen. Bei der Eliminierung wird versucht, möglichst Sätze mit vielen Visuren beizubehalten. Unvermeidbar bleibt, dass Richtungsmessungen mit Redundanzanteilen von 0,5 auftreten. Die Streckenmessungen werden ebenso um die Hälfte gekürzt.

Für die zweite Stufe wird auf eine homogene Anzahl an Messungen geachtet. Auf jedem Standpunkt sollen genau zwei Richtungssätze verwendet werden, ebenso wird jede Strecke zweimal angeführt (das entspricht dem Zustand aus der vorigen Reduktion).

Die Datensätze 3 und 4 sind sich hinsichtlich der Redundanz zwar ähnlich, dies ist aber bewusst festgelegt. In Datensatz 3 wird auf jedem Standpunkt nur mehr ein Richtungssatz belassen und die Strecken bleiben doppelt bestehen. In Datensatz 4 werden die Strecken auf das Minimum gekürzt und treten jeweils nur mehr einfach auf, während die Richtungssätze - wie in Datensatz 2 - in zweifacher Ausführung eingeführt werden. Diese Einteilung wird vorgenommen, um einen etwaigen Unterschied zwischen Richtungs- und Streckendominanz ausmachen zu können.

Im fünften Schritt werden sowohl die Strecken, als auch die Richtungen auf ihr Minimum reduziert. Doppelte Richtungsvisuren innerhalb eines Satzes werden ebenso gestrichen wie doppelte Streckenvisuren, sodass Datensatz 5 der Minimalkonfiguration für das Netz entspricht.

Rückblickend ist ein relativ großer Unterschied zwischen der Gesamtredundanz 4 und 2 bemerkbar, obwohl ein möglichst sinnvolle, der Praxis entsprechende Reduktion vorgenommen wurde. Zur Überbrückung werden die Richtungen soweit gekürzt, dass pro Satz keine doppelten Visuren

mehr auftreten, auf jedem Standpunkt aber dennoch zwei Richtungssätze verbleiben. Die Strecken bleiben in doppelter Ausführung bestehen. Eine andere Möglichkeit der Messungsreduktion, ohne dabei Information in Form von einzigartigen Visuren zu verlieren, ist nicht möglich.

Um den Überblick über die verschiedenen Datensätze und ihren Aufbau zu wahren, sind in Tabelle 5.6 die wichtigsten Kenngrößen zusammengefasst. Die Tabelle umfasst die Anzahl der Beobachtungen  $n$  und der Unbekannten  $u$ , den Datumsdefekt  $d$  und die Gesamtredundanz  $r$ .

Tabelle 5.6: Übersicht über die erstellten (reduzierten) Datensätze

Nr.	$n$	$u$	$d$	$r$
0	133	42	3	94
1	75	30	3	48
2	62	26	3	39
3	41	21	3	23
4	49	26	3	26
5	28	21	3	10
6	54	26	3	31

Die Kürzung der Messungen wirkt sich auf die Redundanzanteile aus, wodurch die minimal detektierbaren Ausreißer neu zu berechnen sind. Tabelle 5.7 gibt die Entwicklung der beiden Größen auszugsweise wieder. Der Einfluss der Reduktion ist enorm: Im stark reduzierten Fall von  $r=10$  muss für eine Streckenbeobachtung der grobe Fehler bereits einen Wert von eineinhalb Meter annehmen, um erkannt zu werden.

Tabelle 5.7: Verlauf der Redundanzanteile und MDB

		$r=94$	$r=39$	$r=10$
Richtungen	$r_i$	0,44 - 0,79	0,47 - 0,73	0,23 - 0,59
	MDB [gon]	0,0023 - 0,0031	0,0024 - 0,0030	0,0027 - 0,0043
Strecken	$r_i$	0,75 - 0,92	0,50 - 0,84	0,001 - 0,61
	MDB [m]	0,034 - 0,056	0,039 - 0,067	0,067 - 1,538

### 5.3.1 Ein grober Fehler

Für die erste Analyse wird nach dem Schema aus Kapitel 5.2 ein einziger grober Fehler an die Messungen angebracht. Die nachfolgenden Untersuchungen beziehen sich auf die sechs reduzierten Datensätze. Der Vollständigkeit halber fließen die Ergebnisse aus dem vorigen Abschnitt in die Analyse mit ein und werden ebenso angeführt.

## L1-Norm

Bei der Berechnung der L1-Referenzlösung mit den Datensätzen 1, 2, 4 und 6 weist das Verfahren jeweils eine Richtungsmessung als fehlerhaft aus, obwohl noch kein Fehler angebracht wurde. Dabei handelt es sich dreimal um dieselbe Messung von Punkt 150 auf 114 und einmal um die Visur von Punkt 150 auf 230. Diese Tatsache deckt sich mit der im vorherigen Kapitel angestellten Vermutung, dass die L1-Norm möglicherweise auch auf andere Fehlerarten anspricht.

Ohne Ausschluss dieser Beobachtungen, wird beim iterativen Anbringen des groben Fehlers an alle Messungen immer die zuvor angezeigte Beobachtung als erste eliminiert, da sie das kleinste Gewicht stellt. Dies führt infolgedessen zu einer Fehlerquote zwischen 94 und 99%, da nur jene Fälle, in denen allein die verfälschte Messung eliminiert wird, als erfolgreiche Detektion zählt. Auf die ausgeglichenen Unbekannten hat dies keinen bedeutenden Einfluss, wie sich im Vergleich mit Tabelle 5.8 zeigt, welche die Erkenntnisse ohne Verwendung der betreffenden Beobachtungen zusammenfasst. Für Datensatz 1 beträgt die maximale Abweichung zur L2-Lösung 1,4cm und zur Referenzlösung 3,7mm. Die prozentuelle Verteilung der Abweichungen entspricht im Großen und Ganzen ebenfalls dem Versuch ohne die Beobachtung. Beim Vergleich von L1 zu L2 treten Differenzen kleiner als 1mm sogar um 10% häufiger auf, als beim Datensatz ohne die betroffene Messung.

Nachdem bei Anwendungen der L1-Norm in der Praxis solch falsche Detektionen nicht bemerkt werden können und auf das Ergebnis vertraut wird, wird jeweils die vorab gefundene Beobachtung für die weiteren Untersuchungen vom Datensatz ausgeschlossen.

Für Tabelle 5.8 und alle zu dieser ähnlichen ist zu beachten, dass die Klassen der Abweichungen unterschiedliche Breiten aufweisen.

Anhand der tabellarischen Darstellung ist eine Abhängigkeit zwischen der Entwicklung der maximalen Abweichungen und der Redundanz zu erkennen. Die Maximalwerte nehmen stetig zu, wobei ein deutlicher Sprung zwischen Datensatz 6 und 3 zu beobachten ist. Innerhalb der ersten vier Untersuchungsfälle tritt kaum eine Verschlechterung auf, weder in Bezug auf die Schätzergebnisse, noch auf die maximalen Abweichungen, obwohl die Redundanz bereits auf ein Drittel der Ausgangssituation reduziert wurde. Vor allem die relativen Abweichungen sind sehr gering, die L1-Norm zählt daher für diese Datensätze neben RANSAC zu den höchst erfolgreichen Verfahren. Sobald eine Beobachtungsgruppe auf ihr Minimum reduziert wird, ist ein starker Anstieg in den Abweichungen sichtbar.

Die größten Abweichungen werden in den ersten vier Fällen (Datensatz 0-2 und 6) jeweils durch Beobachtungen hervorgerufen, bei denen der grobe Fehler richtig lokalisiert wird. Dieses Verhalten ist aus den vorigen Untersuchungen bereits bekannt und auf die geometrische Verteilung der Beobachtung zurückzuführen.

Die robuste L1-Norm würde gegebenenfalls den gesetzlichen Anforderungen an die mittlere Punktlagegenauigkeit von 4 bzw. 5cm in vier von sieben Fällen gerecht.

Tabelle 5.8: Verteilungsverlauf der Abweichungen in den Koordinaten in Abhängigkeit von der Redundanz  $r$  (L1-Norm)

	$r_0=94$		$r_1=48$		$r_2=39$		$r_3=31$		$r_3=23$		$r_4=26$		$r_5=10$	
	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ
< 1mm	25%	99%	40%	85%	37%	91%	35%	92%	30%	91%	18%	71%	10%	53%
< 1cm	99%	100%	99%	100%	94%	100%	95%	100%	89%	97%	66%	75%	51%	54%
< 1dm	100%	100%	100%	100%	100%	100%	100%	100%	98%	98%	79%	80%	62%	62%
Maximalwert [m]	0,011	0,002	0,014	0,004	0,015	0,005	0,015	0,009	1,339	1,347	6,913	6,912	9,599	9,600
Fehldetektionen	15/133		9/74		11/61		3/53		3/41		15/48		14/28	

Tabelle 5.9: Verteilungsverlauf der Abweichungen in den Koordinaten in Abhängigkeit von der Redundanz  $r$  (LTS)

	$r_0=94$		$r_1=48$		$r_2=39$		$r_3=31$		$r_3=23$		$r_4=26$		$r_5=10$	
	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ	absolut	relativ
< 1mm	30%	92%	38%	69%	27%	74%	39%	69%	54%	55%	27%	58%	36%	24%
< 1cm	66%	100%	99%	98%	99%	99%	99,5%	99%	95%	94%	90%	95%	89%	89%
< 1dm	100%	100%	100%	100%	100%	100%	100%	100%	96%	96%	98%	98%	94%	94%
Maximalwert [m]	0,012	0,007	0,017	0,021	0,020	0,025	0,015	0,019	1,362	1,362	6,915	6,911	10,327	10,328
Fehldetektionen	20/133		26/75		7/62		9/54		5/41		46/49		11/28	

Obwohl in Datensatz 3 und 4 jeweils eine Gruppe minimiert ist, unterscheiden sie sich in den Häufigkeitsverteilungen der Abweichungen und den Extremwerten. Sind alle Richtungsbeobachtungen nur einmal vorhanden, die Strecken aber doppelt (Fall 3), sind die Abweichungen kleiner als im umgekehrten Fall. Alle Abweichungen größer als einen Dezimeter werden durch einen einzigen, nicht richtig detektierten groben Fehler in ein und derselben Beobachtung ausgelöst. Der Unterschied zwischen den beiden Fällen lässt vermuten, dass einfach gemessene Strecken das Ergebnis stärker verzerren als Richtungsmessungen, die nur einmal beobachtet werden. Dies ist nicht weiter verwunderlich, ruft man sich die Redundanzanteile aus Tabelle 5.7 in Erinnerung. Eine weitere Bestätigung hierfür ist, dass in Fall 4, bei dem das Verhältnis zwischen Richtungen und Strecken 3:1 beträgt, die maximale Abweichung weitaus größer ist, als in Fall 3.

Für den Extremfall mit Datensatz 5 sind die Ergebnisse in allen Kategorien unzureichend. Die Hälfte der Abweichungen ist bereits größer als einen Zentimeter, zudem liegt ein Drittel sogar im Bereich von über einem Dezimeter. Mit einer maximalen Abweichung von fast zehn Meter sind die Ergebnisse für Vermessungsaufgaben unbrauchbar.

Es kann zudem nur mehr die Hälfte aller groben Fehler richtig detektiert werden, wobei kein einziger Fehler in den Streckenmessungen aufgedeckt werden kann, was auch die Abweichungen über einen Zentimeter zur Folge hat. Bei den Richtungsmessungen treten die altbekannten Schwierigkeiten bei Beobachtungen mit einem Redundanzanteil von 0,5 auf bzw. liegt ein Verschmierungseffekt über Punkt 152 vor, der geometrisch gesehen ungünstig liegt. Messungen, die diesen Punkt beinhalten, schieben sich den Fehler gewissermaßen gegenseitig zu. Nicht weiter verwunderlich ist daher, dass die größte Abweichung auch in dessen Koordinate zu finden ist.

Einen interessanten Verlauf zeigt die Fehldetektionsrate. Zu Beginn liegt sie im Bereich von 11-18% und fällt dann stark auf den einstelligen Prozentbereich ab, obwohl die Redundanz weiter sinkt. Dabei darf nicht vergessen werden, dass die Rate in einzelnen Fällen beschönigt wurde, wie eingangs gezeigt wurde.

## Biber-Schätzer

Bis zu einer Verringerung der Redundanz um zwei Drittel, nehmen die maximalen Abweichungen stetig um wenige Millimeter zu, bis bei Anwendung von Datensatz 3 eine enorme Erhöhung eintritt, siehe Tabelle 5.10.

Tabelle 5.10: Verteilungsverlauf der Abweichungen in den Koordinaten in Abhängigkeit von der Redundanz  $r$  (Biber-Schätzer)

	$r_0=94$	$r_1=48$	$r_2=39$	$r_6=31$	$r_3=23$	$r_4=26$	$r_5=10$
< 1mm	76%	59%	52%	46%	34%	46%	18%
< 1cm	99%	99%	99%	98%	93%	94%	78%
< 1dm	100%	100%	100%	100%	98%	99%	91%
Maximalwert [m]	0,012	0,016	0,020	0,022	1,296	0,384	10,305
Fehldetektionen	14/133	2/75	1/62	7/54	3/41	8/49	8/28

Ein unerwartetes Ergebnis ergibt sich mit Datensatz 4. Wo die übrigen drei robusten Methoden einen Anstieg verzeichnen, verringert sich beim Biber-Schätzer der Maximalwert deutlich. In Kombination mit der Häufigkeitsverteilung deutet der Wert zudem darauf hin, dass dieser nur durch eine oder wenige Messungen verursacht wird, was bei detaillierter Analyse der einzelnen Abweichungswerte bestätigt wird. Eine einzige Streckenmessung ist für die Abweichung über einen Dezimeter verantwortlich, wobei der Fehler in dieser Messung korrekt detektiert wird. Die aufgestellte These des stärkeren Einflusses von Strecken ist für den Biber-Schätzer nicht gültig.

In den ersten drei Versuchen (Datensatz 0-2 und 6) bewirken jeweils richtig detektierte Fehler die größten Abweichungen, für die restlichen Versuche zeigt sich das Gegenteil.

Zwischen Datensatz 2 und 6 ist nur eine minimale Veränderung erkennbar. Die Eliminierung von doppelten Visuren hat beim Biber-Schätzer in Bezug auf die Parameterschätzung keinen signifikanten Einfluss, hinsichtlich der Fehlerdetektion ist jedoch eine Verschlechterung zu beobachten.

Die Häufigkeiten der Abweichungen in den jeweiligen Intervallen verhalten sich gesamt betrachtet direkt proportional zur Redundanz. Die Maximalwerte bewegen sich bei ausreichender Redundanz im Zentimeterbereich. Für Datensatz 4 schafft der Biber-Schätzer die niedrigste Abweichung im Vergleich zu den übrigen Methoden. Speziell für die diesen und den fünften Datensatz ergeben sich so hohe Abweichungshäufigkeiten für den Bereich kleiner als einen Zentimeter, wie sie sonst nur mehr bei LTS auftreten.

Was die Aufdeckbarkeit der groben Fehler betrifft, kann der Biber-Schätzer mit einer teilweise geringen Fehlerrate aufwarten. Waren im Ausgangsdatsatz nur Richtungen mit einem Redundanzanteil von 0,5 für die Fehldetektionen verantwortlich, zählen in allen anderen Versuchen auch Streckenmessungen und andere Richtungsmessungen dazu. Für die Minimalkonfiguration erreicht der Biber-Schätzer die niedrigste Fehlerquote aller robusten Verfahren.

In seltenen Fällen ist ein Zusammenhang zwischen der fehlerhaften Beobachtung und den Koordinaten des zugehörigen Stand- und Zielpunktes erkennbar, meist aber weichen die Koordinaten aller Netzpunkte um mehr oder weniger große Beträge von ihrer Referenzposition ab. Ein Beispiel für eine deutliche Verbindung tritt bei Datensatz 6 und der Streckenmessung von Punkt 11 auf 151.

Tabelle 5.11: Abweichungen der Netzpunkte

Nr.	Y [m]	X [m]
11	-0,006	0,002
114	0,000	0,002
150	0,001	0,000
151	0,004	-0,004

Der grobe Fehler wird richtig detektiert, durch die Modifikation resultieren aber Abweichungen, welche in Abbildung 5.7 als Verschiebungsvektoren überzeichnet dargestellt sind. Es wird ersichtlich, dass hauptsächlich die Koordinaten des Stand- und Zielpunktes betroffen sind, welche auch die größten Abweichungen besitzen (siehe Tabelle 5.11). Die beiden Punkte werden zueinander

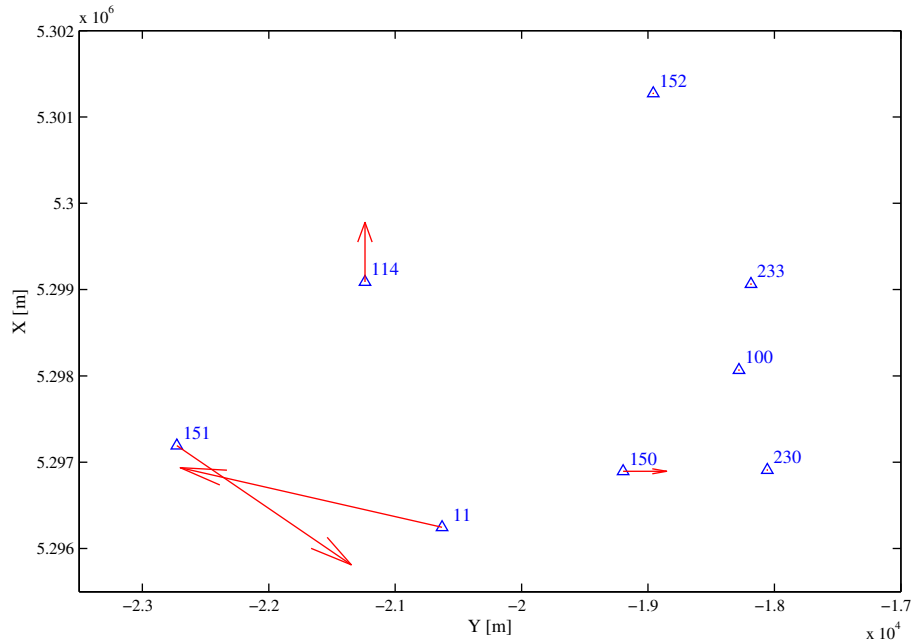


Abbildung 5.7: Verschiebungsvektoren bei grobem Fehler in der Streckenmessung 11-151 (nicht maßstäblich)

verschoben, während sich die Koordinaten der zwei nächstgelegenen Netzpunkte (114, 150) von der betroffenen Stelle entfernen. Die restlichen Netzpunkte weisen Abweichungen unter einem Millimeter auf.

## RANSAC

Bei der Bearbeitung von Datensatz 5 stößt RANSAC erstmals an seine Grenzen: Wird der grobe Fehler an die Streckenbeobachtung 230-114 angebracht, ist der RANSAC-Algorithmus nicht im Stande eine optimale Lösung berechnen. Die Ermittlung der endgültigen Lösung gelingt nicht, da zu wenig unterstützende Beobachtungen vorliegen, um zumindest eine eindeutige Lösung zu berechnen, woraufhin die Berechnung abbrechen muss. Die Auswirkung eines groben Fehlers in dieser Messung kann daher nicht untersucht werden. Die verursachende Streckenbeobachtung ist jene, die bei der L1-Norm ebenfalls zu einer frühzeitigen Elimination führt. Für die restlichen Beobachtungen treten keine Schwierigkeiten auf.

Tabelle 5.12: Verteilungsverlauf der Abweichungen in den Koordinaten in Abhängigkeit von der Redundanz  $r$  (RANSAC)

	$r_0=94$	$r_1=48$	$r_2=39$	$r_6=31$	$r_3=23$	$r_4=26$	$r_5=10$
< 1mm	98%	97%	95%	94%	91%	69%	48%
< 1cm	100%	100%	100%	100%	95%	76%	56%
< 1dm	100%	100%	100%	100%	97%	80%	63%
Maximalwert [m]	0,004	0,003	0,005	0,005	1,353	7,434	10,325
Fehldetektionen	9/133	3/75	1/62	3/54	3/41	13/49	13/27

RANSAC vermag es, trotz einer Redundanzreduktion um zwei Drittel, Ergebnisse zu schätzen, die von einem groben Fehler nahezu unbeeinflusst sind. Während in den Versuchen 0-2 und 6 die korrekt detektierten fehlerbehafteten (Richtungs-) Beobachtungen die größten Abweichungen verschulden, sind es in den anderen drei Fällen die nicht richtig identifizierten Fehler. Es handelt sich dabei entweder um Streckenmessungen oder um Richtungen, die einen Redundanzanteil ungleich 0,5 besitzen.

Der Unterschied in den Maximalwerten zwischen Datensatz 3 und 4 unterstützt die getroffene Annahme bezüglich der Dominanz der Strecken über die Richtungen.

Die Abweichungsverteilungen und Fehldetektionen der letzten beiden Datensätze weichen sichtbar vom vorherigen Verlauf ab. Der Algorithmus kann im Extremfall nur die Hälfte aller fehlerbehafteten Beobachtungen aufdecken, wobei alle Streckenmessungen darunter fallen. Betreffend die Richtungen gilt dieselbe Systematik wie bei der L1-Norm. Die Erfolgsrate von etwa 50% spiegelt sich ebenso in der Verteilung der Abweichungen wider, wo etwa die Hälfte unter einem Zentimeter liegt. Schließt man das Schätzergebnis der Beobachtungen, in denen der grobe Fehler nicht korrekt aufgedeckt wird, von der Statistik aus, betragen die Abweichungen maximal einen halben Zentimeter. Mit der größten Abweichung von über zehn Metern übertrifft RANSAC die Werte der vorangehenden robusten Methoden.

Datensatz 6 weicht bei dieser Methode ebenso nicht vom Ergebnisverlauf der vorigen Datensätze ab und erlaubt, die niedrige Anzahl der Fehldetektionen beizubehalten.

Zusammengefasst liefert RANSAC für die bis zu auf ein Drittel der Ausgangsredundanz reduzierten Datensätze die besten Resultate bei der Schätzung der Unbekannten. Die Aufdeckung der groben Fehler ist wie beim Biber-Schätzer höchst effizient. Beim nur minimal überbestimmten Netz schlägt das Verfahren genau in das Gegenteil um und bleibt in beiden Punkten hinter den Erwartungen zurück.

## LTS

LTS weist einen konstant hohen Anteil an Abweichungen unter einem Dezimeter auf, siehe Tabelle 5.9. In allen sechs Untersuchungen bleibt der Prozentanteil deutlich über 90. Ebenso ist die Häufigkeit im Bereich von unter einem Zentimeter im Vergleich zu den übrigen robusten Verfahren durchwegs am größten.

Die maximalen Abweichungen werden - bis auf Datensatz 0 - jeweils durch die falsch detektierten Fehler verursacht, wodurch sich das Verfahren von den anderen unterscheidet. Anhand einer genauen Analyse gilt für das vorliegende Netz: Es kann davon ausgegangen werden, dass unabhängig von der Redundanzreduktion bei einer korrekten Fehlerzuordnung die größten absoluten Abweichungen im Bereich von 8-15mm liegen.

Die Fehldetektionen betreffen neben den Richtungsmessungen mit einem Redundanzanteil von 0,5 auch Streckenmessungen und Richtungen, bei denen nicht eindeutig festgestellt werden kann, warum die Detektion fehlschlägt.



Gesamt betrachtet befinden sich die maximalen Abweichungen in den ersten vier Fällen in einer Größenordnung, die ohne weiteres für amtliche Vermessungsarbeiten ausreicht. Bei den restlichen Versuchen liegt sie im erwarteten hohen Meterbereich. Trotzdem ist hier der Häufigkeitsanteil von Abweichungen unter einem Zentimeter über alle Verfahren gesehen am höchsten. Derart hohe Werte treten auch beim Biber-Schätzer auf. Diese beiden Verfahren sind in der Lage, die ansonsten sehr gute Leistung von RANSAC in diesem Aspekt zu überbieten. In genau diesen Untersuchungen ist auch der Unterschied zwischen den Maximalwerten im absoluten und relativen Vergleich, vor allem für Abweichungen unter einem Millimeter, gemäßigter. Für Datensatz 6 nehmen die maximalen Abweichungen in Relation zu den vorgehenden Datensätzen etwas ab.

In Bezug auf die Fehldetektionen variiert das Verhalten von LTS. In etwa der Hälfte aller Untersuchungen liegt das Verfahren gesamt betrachtet im Mittelfeld, ansonsten führt es zu relativ großen Fehlraten. Am stärksten ist dies bei Anwendung von Datensatz 4 zu bemerken. Nur drei von 49 Beobachtungen werden eindeutig identifiziert, bei 36 anderen Beobachtungen wird die betroffene Messung detektiert und weitere Fehler in mindestens zwei anderen Messungen angenommen. In den übrigen Fehldetektionen treten weitere Verschmierungseffekte auf.

LTS ist nicht einfach zu beurteilen. Auf der einen Seite liegen die Abweichungen bei ausreichender Redundanz in einem verwertbaren Bereich. Auf der anderen Seite ist die Anzahl der Fehldetektionen relativ hoch. Diese sind meistens nicht nachvollziehbar und folgen keiner Systematik, wodurch das Risiko steigt, problematische Konstellationen nicht erkennen zu können und Abweichungen im Meterbereich zu erhalten.

### 5.3.2 Zwei grobe Fehler

In diesem Abschnitt wird untersucht, wie die robusten Verfahren auf das gleichzeitige Vorliegen von zwei groben Fehlern reagieren. Die beiden Fehler werden an jene Beobachtungen angebracht, die im vorigen Abschnitt die maximalen Abweichungen im jeweiligen Verfahren hervorgerufen haben, unabhängig davon, ob sie aufgedeckt werden konnten oder nicht. Zu beachten ist, dass in jedem Verfahren komplett unterschiedliche Messungen zu den Maximalwerten geführt haben. Das primäre Ziel ist, die ungünstigste Ausgangssituation zu schaffen und die schlechtestenfalls resultierenden Auswirkungen abzuschätzen. Auf diese Art sollen die Grenzen der Algorithmen aufgezeigt werden.

Im Vordergrund steht die Untersuchung des Einflusses auf die Parameterschätzung. Für L1 und LTS wird ab sofort nicht mehr zwischen absoluten und relativen Abweichungen unterschieden. Der Fokus wird nun mehr auf die absoluten Werte gelegt, da die Abweichungen zur L2-Norm von größerer Bedeutung sind und in den relativen Vergleichen jeweils andere Messungen für die maximalen Abweichungen verantwortlich sind, was zur Unübersichtlichkeit führen würde.

Die Versuche werden nur für jene Datensätze und Verfahren durchgeführt, bei denen sich die Abweichungen (bei Vorliegen eines groben Fehlers) in einem für amtliche Vermessungen zulässigen Bereich befinden (4 bzw. 5cm Genauigkeitsgrenze). Aus den Ergebnistabellen der Tests

mit einem einzigen groben Fehler ist abzulesen, dass die Datensätze 0-2, sowie 6 für die weitere Vorgehensweise in Frage kommen.

## Ergebnis

Die resultierenden Abweichungen von L1, RANSAC und LTS von der L2-Norm in Tabelle 5.13 bewegen sich im Bereich von ein bis zwei Zentimetern und sind für eine Verwendung in der Praxis und die weiteren Untersuchungen geeignet.

Beim Biber-Schätzer sind nur die Abweichungen zweier Datensätze gering genug, um einen weiteren groben Fehler anbringen zu können. Mit Datensatz 1 und 2 ergeben sich vielfach größere Maximalwerte, die sich vor allem auch von den übrigen Beträgen in der Tabelle abheben. In Datensatz 6 nimmt die größte Abweichung knapp drei Zentimeter an und liegt noch im Bereich des Brauchbaren. Die Lokalisierung der angebrachten groben Fehler ist dementsprechend auch nur in Datensatz 0 und 6 erfolgreich.

RANSAC liefert die am wenigsten beeinflussten Ergebnisse mit Abweichungen deutlich unter einem Zentimeter. Mit zunehmender Redundanzverringering steigt der Betrag der maximalen Abweichung leicht an. Die Methode kann in den ersten drei Datensätzen alle groben Fehler aufdecken, bei Datensatz 6 weist es eine zusätzliche Beobachtung als fehlerhaft aus.

Die L1-Norm deckt in der Hälfte der Datensätze (2 und 6) alle Fehler richtig auf, in den anderen beiden Fällen wird zusätzlich eine fehlerfreie Beobachtung als grober Fehler identifiziert, die aber mittels einer Kontrolle durch den Anwender oder eine Herabsetzung des Gewichtsgrenzwertes korrigiert werden kann. Eine Erfolgsquote von 100% ist damit möglich. Wie auch bei RANSAC nimmt die Größe der maximalen Abweichung mit abnehmender Redundanz zu.

LTS verhält sich ähnlich und schafft nur in den letzten beiden Versuchen eine erfolgreiche Detektion. Entweder treten Probleme bei Beobachtungen mit dem Redundanzanteil 0,5 auf oder zusätzliche Messungen werden fälschlicherweise detektiert. Hinsichtlich der Entwicklung der maximalen Abweichung ist eine Parallele zum Biber-Schätzer erkennbar. Die Werte steigen von Datensatz 0 auf Datensatz 1 an und fallen in den weiteren beiden Fällen (stark) ab.

Tabelle 5.13: maximale Abweichungen von der L2-Norm in [m] bei Vorliegen von zwei groben Fehlern

Datensatz	L1	Biber	RANSAC	LTS
0	0,011 =	0,009 <	0,002 <	0,012 =
1	0,014 =	1,187 >	0,004 >	0,017 =
2	0,016 >	0,870 >	0,005 =	0,015 <
6	0,016 >	0,027 >	0,006 >	0,004 <

<, >, = : betragsmäßige Veränderung der max. Abweichung im Vergleich zum Vorliegen nur eines groben Fehlers

In Tabelle 5.13 wird anhand der Relationsoperatoren angedeutet, in welche Richtung sich die Maximalwerte im Vergleich zu den Untersuchungen mit nur einem groben Fehler entwickeln. Das Vorliegen eines zweiten Fehlers verursacht Auswirkungen, die insgesamt alle drei Vergleichssymbole hervorrufen. Die Geometrie des Netzes und die räumliche Verteilung der Beobachtungen sind dafür hauptverantwortlich. Jede einzelne Beobachtung trägt seinen Teil zur Bestimmung des Netzes bei, weshalb die Richtung der Auswirkung individuell ist und von der Kombination an betroffenen Messungen abhängt. Durch die Redundanzreduktion ergibt sich eine neue Verteilung und ein neues Zusammenspiel der Beobachtungen. Auch wenn bei zwei verschiedenen Datensätzen dieselben Messungen mit einem Fehler besetzt sind, ergibt sich dadurch ein anderer Effekt. Dies trifft beispielsweise bei RANSAC und Datensatz 0 und 1 zu. Zudem tragen die erwähnten Eigenschaften der Algorithmen (Elimination der Beobachtung, Gewichtsmatrix, Modifikation der Beobachtungen,...) dazu bei, dass sich die Abweichungen unterschiedlich verhalten.

### 5.3.3 Drei grobe Fehler

Mit Ausnahme des Biber-Schätzers, ändert sich die Größenordnung der maximalen Abweichungen kaum, wie Tabelle 5.14 zu entnehmen ist. Die Relationsoperatoren beziehen sich dabei weiterhin auf die Abweichungen aus Kapitel 5.2.

Datensatz 6 überschreitet in Kombination mit dem Biber-Schätzer und drei groben Fehlern die zulässige Abweichung von vier bis fünf Zentimeter deutlich. Die Fehlerdetektion kann nicht erfolgreich durchgeführt werden. Die maximale Abweichung mit Datensatz 0 genügt weiterhin den Ansprüchen, wenn auch eine klare Vergrößerung des Wertes zu beobachten ist.

Tabelle 5.14: maximale Abweichungen von der L2-Norm in [m] bei Vorliegen von drei groben Fehlern

Datensatz	L1	Biber	RANSAC	LTS
0	0,011 =	0,021 >	0,006 >	0,012 =
1	0,014 =	-	0,002 <	0,017 =
2	0,018 >	-	0,004 <	0,009 <
6	0,017 >	0,943 >	0,004 <	0,006 <

<, >, = : betragsmäßige Veränderung der max. Abweichung  
im Vergleich zum Vorliegen nur eines groben Fehlers

Durch RANSAC ergeben sich die kleinsten Abweichungen. Die Fehlerdetektion misslingt nur mit Datensatz 6. Die übrigen Verfahren verhalten sich diesbezüglich gleich wie in der vorigen Untersuchung. In Datensatz 2 und 6 funktioniert das Auffinden der Fehler mit der L1-Norm einwandfrei, für die anderen beiden Datensätze ergeben sich dieselben Probleme wie mit zwei Fehlern.

LTS folgt mit Datensatz 1 demselben Prinzip wie zuvor. Bei Verwendung von Datensatz 0 wird nur einer von drei Fehlern detektiert. Die Ursache hierfür hängt damit zusammen, dass zwei der drei fehlerbehafteten Richtungsmessungen in einem gemeinsamen Satz auftreten und die bereits erörterte Problematik aus Kapitel 4.4 auftritt. Für die übrigen zwei Datensätze ergibt sich eine

problemlose Identifikation der fehlerbehafteten Messungen. Mit maximalen Abweichungen um zwei Zentimeter, sind auch hier die Ergebnisse relativ nahe an der L2-Lösung und für weitere Untersuchungen qualifiziert.

### 5.3.4 Vier grobe Fehler

Bei Vorliegen von vier groben Fehlern bewegen sich die maximalen Abweichungen im selben Bereich wie zuvor und verschlechtern sich, wenn überhaupt, nicht merklich. Die einzige Ausnahme bildet erneut der Biber-Schätzer. Dessen Maximalwert steigert sich bei Einbringung eines weiteren groben Fehlers um einen Zentimeter. Die Lokalisierung der Fehler kann hingegen erfolgreich durchgeführt werden. In Bezug auf die Entwicklung der maximalen Abweichungsbeträge zeigen die L1-Norm und RANSAC ein ähnliches Verhalten. Mit kontinuierlicher Redundanzreduktion steigt der Maximalbetrag.

Bei der Fehlerdetektion der übrigen Verfahren treten Unterschiede zu den vorangegangenen Versuchen auf. RANSAC schafft nur mehr in zwei von vier Fällen eine korrekte Aufdeckung, da jeweils in einem Satz zwei fehlerbesetzte Richtungen auftreten und infolgedessen die dritte Beobachtung fälschlicherweise detektiert wird.

Der L1-Norm gelingt unter anderem durch Überprüfung des Gewichtsverlaufs zweimal eine Fehlerdetektionsrate von Null. In den anderen beiden Fällen tritt dieselbe Problematik bezüglich Satzmessungen auf wie bei RANSAC.

LTS ist in der Lage mit Datensatz 2 und 6 eine korrekte Identifizierung umzusetzen, ansonsten ergeben sich dieselben Schwierigkeiten der anderen Verfahren.

Tabelle 5.15: maximale Abweichungen von der L2-Norm in [m] bei Vorliegen von vier groben Fehlern

Datensatz	L1	Biber	RANSAC	LTS
0	0,011 =	0,031 >	0,005 >	0,010 <
1	0,013 <	-	0,005 >	0,017 =
2	0,017 >	-	0,008 >	0,008 <
6	0,018 >	-	0,009 >	0,006 <

<, >, = : betragsmäßige Veränderung der max. Abweichung  
im Vergleich zum Vorliegen nur eines groben Fehlers

Speziell aus der letzten Untersuchung sollte nicht der Schluss gezogen werden, dass grundsätzlich mit Zunahme der groben Fehler die Fehlerdetektionsrate steigt. Es ist wesentlich, welche Messungen fehlerbehaftet sind. Bei dieser Untersuchung kommt es häufiger zu Fehldetektionen, da vorwiegend problematische Konstellationen an Richtungsmessungen provoziert werden. Mit dem Anstieg des Fehleranteils vergrößert sich zwar die Wahrscheinlichkeit, dass ebensolche Kombinationen auftreten, es ist aber nicht in jedem Fall garantiert.

## 5.4 Conclusio

Die Versuche bestätigen nicht die anfangs vielleicht vermutete Annahme, dass mit zunehmendem Anteil an groben Fehlern, die Abweichungen von der fehlerfreien L2-Lösung ansteigen. In vielen Fällen führt es zu einem gegensätzlichen Verhalten, nämlich dass mehrere grobe Fehler zu einer geringeren Abweichung führen als bei Vorliegen eines einzigen groben Fehlers.

Als Favorit hinsichtlich der Abweichungsbeträge ist der RANSAC-Algorithmus zu nennen. Vom Vorliegen nur eines Fehlers, bis hin zu vier Fehlern, liefert er als einziger durchwegs Abweichungen im Millimeterbereich. Einzig bei Datensätzen mit niedriger Redundanz versagt auch er.

Der Biber-Schätzer kann bei Anbringen eines groben Fehlers mit den anderen Verfahren mithalten, zeigt aber bei Erhöhung des Fehleranteils Schwächen. Die Maximalwerte fallen dann deutlich höher aus als bei den übrigen robusten Schätzern.

Die L1-Norm und LTS sind sich bei Vorhandensein eines groben Fehlers ähnlich (siehe auch Abb. 5.8) und unterscheiden sich bei einem Anstieg des Fehleranteils zunehmend. Dabei kommt es darauf an, welche Gesamtredundanz im Datenmaterial vorliegt. Ab einer Redundanzreduktion von etwa zwei Drittel (Datensatz 2) driften die maximalen Abweichungsbeträge der beiden Methoden relativ klar auseinander. LTS ist in der Lage die Maximalwerte von L1 um das Drei- bis Vierfache zu unterbieten.

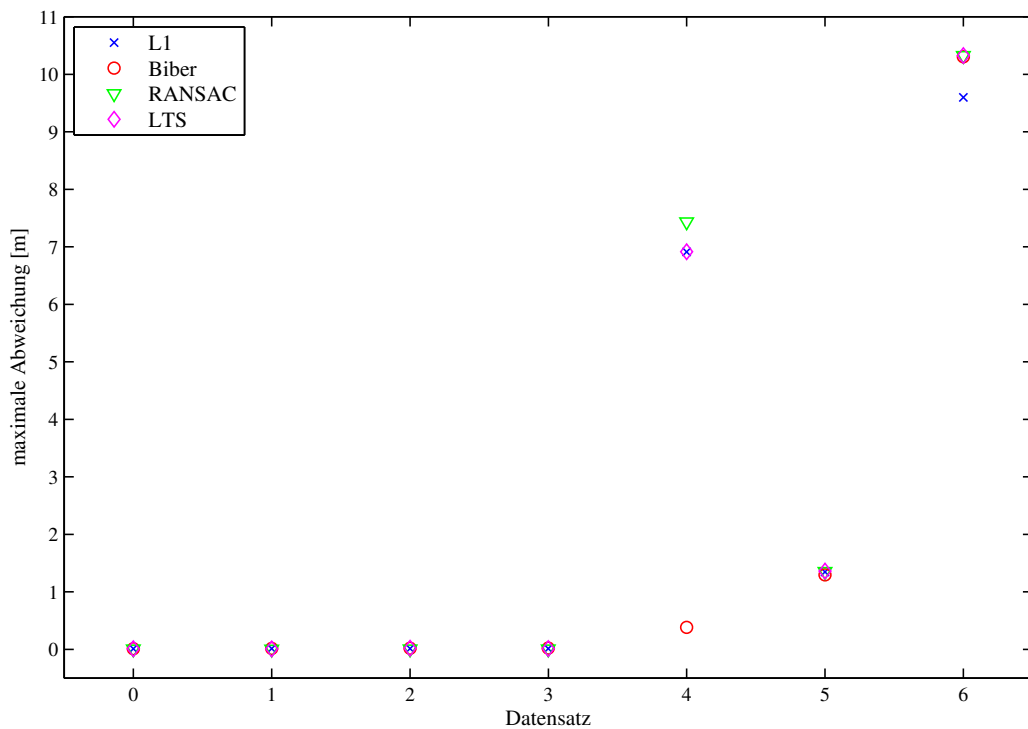


Abbildung 5.8: Übersicht über die maximalen Abweichungen bei Vorliegen eines groben Fehlers

Gesamt betrachtet sind bei Anbringen mehrerer Fehler die Ergebnisse von RANSAC, L1 und LTS teils höchst zufriedenstellend, da selbst mit vier Fehlern die maximalen Abweichungen nur knapp zwei Zentimeter betragen.

In der Kategorie der Aufdeckbarkeit von groben Fehlern ist eine Beurteilung schwierig. Keine Methode zeigt über alle Untersuchungen ein gleiches oder annähernd gleich bleibendes Verhalten. Ein verfahrensübergreifender Zusammenhang zwischen Redundanz und Anzahl der Fehldetektionen kann nicht eindeutig festgestellt werden.

Tabelle 5.16: Fehlerdetektion bei Anbringen von ein bis vier Fehlern (v.l.n.r): L1, Biber, RANSAC, LTS

	2	3	4		2	3	4		2	3	4		2	3	4
0	×	×	×	0	✓	✓	✓	0	✓	✓	✓	0	×	×	×
1	×	×	×	1	×	×	×	1	✓	✓	×	1	×	×	×
2	✓	✓	×	2	×	×	×	2	✓	✓	×	2	✓	✓	✓
6	✓	✓	✓	6	✓	×	×	6	×	×	×	6	✓	✓	✓

Für die Untersuchungen mit wechselndem Fehleranteil sind eher Strukturen und Zusammenhänge zu erkennen, wie ein Blick auf Tabelle 5.16 beweist. Die Spalten repräsentieren die Anzahl an Fehlern, die Zeilen die verwendeten Datensätze. Ein Kreuz symbolisiert eine Fehldetektion bzw. dass keine komplett richtige Fehleridentifikation stattgefunden hat, sondern die Fehler entweder nur teilweise oder zusammen mit fehlerfreien Beobachtungen detektiert wurden. RANSAC hat mit vier Fehlern und Datensatz 6 die meisten Probleme, die L1-Norm schneidet hauptsächlich mit diesem Datensatz am besten ab. LTS grenzt die Erfolge deutlich von den Misserfolgen ab, welche sich rein nach den Datensätzen orientieren. Der Biber-Schätzer kann nur mit Datensatz 0 überzeugen. Festzuhalten ist, dass die Erfolgsquote über alle Versuche und Verfahren im besten Fall nur knapp über 50% liegt.

Bei Vorliegen nur eines groben Fehlers besitzt LTS tendenziell die höchste Fehldetektionsrate. Nachdem die Auswahl der Beobachtungen zu Beginn auf dem Zufall basiert und für die Untersuchungen kein fixer Ausgangsdatsatz möglich ist, variieren die Ergebnisse. Dies ist in der Praxis problematisch, da der Benutzer keine Orientierungsmöglichkeit hat, um zu entscheiden, welche angedeuteten Beobachtungen tatsächlich auszuschließen sind.

RANSAC kann am öftesten eine Fehldetektionsrate von unter 10% vorweisen, besitzt aber im Gegenzug für beide Datensätze, in denen eine Beobachtungsgruppe minimiert wurde, deutlich höhere Werte.

Die L1-Norm und der Biber-Schätzer sind sich in diesem Aspekt ähnlich und wechseln zwischen Fehlraten im Bereich von 2-18%, mit Ausreißern in Datensatz 5.

In Abbildung 5.9 ist ein Überblick über die Fehldetektionsraten aller Verfahren gegeben, welche die Unregelmäßigkeiten widerspiegelt. Die Angaben beziehen sich auf die unkorrigierten Raten, also inklusive der eigentlich nicht auflösbaren Fehler bei Beobachtungen mit  $r_i = 0,5$ .

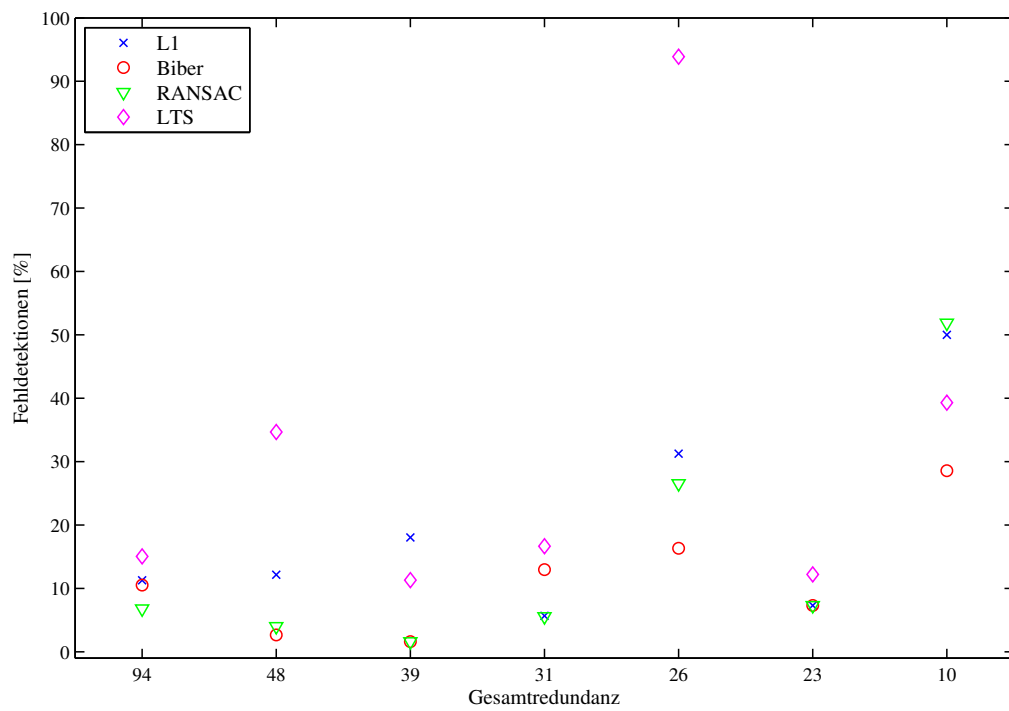


Abbildung 5.9: Übersicht über den Anteil an Fehldetektionen bei Vorliegen eines groben Fehlers

## Kapitel 6

# Zusammenfassung und Ausblick

Die robusten Algorithmen setzen unterschiedliche Strategien ein, um groben Fehlern entgegenzuwirken. L1, RANSAC und LTS unterbinden den Einfluss völlig, indem sie die betroffene Messung von den Berechnungen ausschließen. Der Biber-Schätzer bezieht alle Daten in die Berechnung der Lösung mit ein, modifiziert aber die betroffene Messung, sodass der Einfluss des Fehlers verringert wird. Durch die verschiedenen Ansätze müssen die Methoden unter mehreren Aspekten verglichen werden.

Während der Biber-Schätzer und RANSAC im Stande sind auch bei fehlerfreien Daten (nahezu) die Lösung der Methode der kleinsten Quadrate zu berechnen, ist dies bei L1 und LTS nicht möglich. Es muss daher eine äquivalente Vergleichsbasis geschaffen werden, um Abweichungen zur jeweils bestmöglichen Lösung ermitteln zu können. Daraus ergeben sich absolute und relative Abweichungen, die einen einfachen Vergleich unübersichtlich wirken lassen und bei dem jede Bewertung mit Einschränkungen verknüpft ist.

Des Weiteren beinhalten RANSAC und LTS Elemente, die per Zufall bestimmt werden und Variationen in den Ergebnissen hervorrufen. Die Untersuchungen haben gezeigt, dass ein mehrmaliges Durchlaufen des Algorithmus - speziell bei LTS - unbedingt notwendig ist.

Die Testdaten, bei denen ursprünglich davon ausgegangen wurde, dass sie keine groben Fehler aufweisen, liefern die erste Bewährungsprobe für die robusten Schätzer. Die voneinander abweichenden Resultate, vor allem hinsichtlich der Kandidaten für grobe Fehler, spiegeln die unterschiedlichen Vorgehensweisen und Charakteristiken der Verfahren wider.

Nach der Bereinigung des Datensatzes bleiben Messungen in teilweise ungünstiger Verteilung zurück, wie beispielsweise Sätze, die nur aus Visuren zum selben Zielpunkt bestehen. Im weiteren Verlauf führt dies zu nicht lösbaren Konstellationen und einer Schmälerung der Erfolgsstatistik im Bereich der Fehlerdetektion. Dadurch wird zwar auf die Problematik hingewiesen, in der Praxis wird solch eine Satzmessung allerdings selten auftreten.

Treten Fehldetektionen auf, können die Ursachen dafür zum Großteil gefunden und schlüssig begründet werden. Teils resultieren sie aus Konstellationen, die speziell das vorliegende Testnetz betreffen und unvermeidbar sind, in jedem Fall geben sie aber wichtige Hinweise für zukünftige Anwendungen.



Die Abweichungen der robusten Parameterschätzung von der L2-Schätzung nehmen bei gleichzeitigem Vorliegen eines groben Fehlers und einer Redundanzreduktion grundsätzlich zu. Entgegen den Erwartungen werden die maximalen Abweichungen in vielen Fällen aber nicht durch Fehlerdetektionen hervorgerufen. Mittels einer Analyse des Datenmaterials können hierfür Begründungen gefunden werden.

RANSAC erzeugt bei der Mehrzahl der Datensätze mit Abstand die geringsten absoluten Abweichungen. Liegt eine sehr niedrige Redundanz vor, kann keines der vier Verfahren seine vergleichbar niedrigen Abweichungen beibehalten.

Werden mehrere grobe Fehler simuliert, ist für die Mehrheit der robusten Schätzer kein signifikanter Anstieg in den Maximalwerten der Abweichungen zu beobachten. Alleine der Biber-Schätzer hat bei zunehmendem Fehleranteil Probleme.

In Fachkreisen hat sich der Biber-Schätzer vor allem durchgesetzt, weil er sich unter anderem durch seine geringe Rechenzeit auszeichnet. Seit seiner Veröffentlichung in den späten 90er Jahren und der seither vorangeschrittenen Technologie, können rechenintensive Verfahren wie RANSAC oder LTS diesen Vorteil relativieren. Neben dem Zeitaspekt zeigt RANSAC von allen vier robusten Methoden über viele Untersuchungen hinweg das günstigste Verhalten. Der Algorithmus liefert bei Vorliegen eines oder mehrerer grober Fehler fast immer die geringsten Abweichungen zur Methode der kleinsten Quadrate und kann im Schnitt bei der Fehlerdetektion am meisten überzeugen. Als weiteren Vorteil ist zu nennen, dass RANSAC mit fehlerfreien Daten exakt die Lösung der Methode der kleinsten Quadrate berechnet. Dieses Verfahren kann daher in jedem Fall bedenkenlos angewendet werden ohne dabei Genauigkeitseinbußen in Kauf nehmen zu müssen. Ist die Gesamtredundanz jedoch gering (und liegt nur ein grober Fehler vor), gerät RANSAC ins Hintertreffen und der Biber-Schätzer kann größere Erfolge vorweisen. Zudem erlaubt der Biber-Schätzer bei Daten ohne grobe Fehler ebenso eine Schätzung, welche mit jener der L2-Norm ident ist. Bei Vorliegen eines groben Fehlers lokalisiert er diesen am häufigsten richtig, erhöht sich der Fehleranteil, sind aber Schwächen erkennbar. Gemeinsam mit der L1-Norm und LTS produziert der Biber-Schätzer Abweichungen vom L2-Ausgleichsergebnis in etwa derselben Größenordnung. LTS besitzt den Nachteil, dass die Fehlerdetektion zumeist unzuverlässig ist, während L1 durchaus gute Erfolge vorweist. Speziell bei der L1-Norm ist eine Kontrolle durch den Anwender empfehlenswert, um ein bestmögliches Resultat hinsichtlich der Fehlerlokalisierung zu erreichen.

Das Themengebiet „Robuste Schätzung“ ist weitreichend und bietet Stoff für weitere Untersuchungen. Vor allem im Bereich der Geodäsie ist sie ein ebenso herausforderndes wie unerlässliches Werkzeug, nicht zuletzt aufgrund des hohen Stellenwerts und ausschlaggebenden Faktors der Geometrie. Interessante Überlegungen für zukünftige Analysen wären etwa die Untersuchung, inwiefern das Verhältnis von Richtungen zu Streckenmessungen ausschlaggebend für die Aufdeckbarkeit von groben Fehlern und die Abweichungen von der L2-Norm ist. Erste Ansätze hierzu werden in dieser Arbeit bereits gezeigt. Ebenfalls angesprochen wird der wesentliche Einfluss des Vorzeichens mehrerer grober Fehler, vor allem wenn die Fehler in derselben Satzmessung auftreten. Ein Variieren der Vorzeichen kann neue Erkenntnisse, besonders hinsichtlich der Aufdeckbarkeit von groben Fehlern, bringen.

Für das vorliegende Netz wird angeführt, dass eine Zunahme des Fehleranteils nicht notwendigerweise mit einem Anstieg der Abweichungen von der L2-Norm einhergeht. Ob dieses Verhalten auch bei anderen (Netz-)Messungen auftritt, kann durch weitere Beispiele aus der Praxis analysiert werden und ist empfehlenswert.

Eine umfassende und deshalb in der vorliegenden Arbeit aus platz- und zeittechnischen Gründen nicht mehr behandelte Thematik befasst sich mit den Auswirkungen, die durch die Konzentration der Fehler auf bestimmte Teile des Netzes entstehen. Dazu zählt, mit welchen Folgen zu rechnen ist, wenn sich mehrere Fehler nur auf eine Seite des Netzes konzentrieren oder gleichverteilt über alle Himmelsrichtungen auftreten.

Durch Bildung aller möglichen Kombinationen von zwei oder mehr groben Fehlern kann der Bruchpunkt eines robusten Verfahrens bestimmt werden. Diese rechenzeitintensive Aufgabe kann mit der Optimierung des im Zuge dieser Arbeit entworfenen Programmcodes verbunden werden. In diesem Zusammenhang kann zudem erörtert werden, inwiefern die Trimmingkonstante  $h$  der LTS-Methode auf die Ergebnisse Einfluss nimmt und eventuell das Potential des robusten Verfahrens weiter ausschöpft.

Ein weiterer Aspekt, der in dieser Arbeit nicht mehr behandelt werden konnte, aber ebenso wichtig ist, betrifft die Rolle der Näherungskordinaten. Welche Konsequenzen zu erwarten sind, wenn grobe Fehler in diesem Teil der Daten vorliegen, bleibt zu erforschen.

# Literaturverzeichnis

- Amiri-Simkooei, AliReza (2003): Formulation of L1 Norm Minimization in Gauss-Markov Models. *Journal of Surveying Engineering*, Vol. 129(1).
- Angst, Michaela; Varija Bunjevac; Wolfgang Fischer et al. (1994): *Landesvermessung Feldübung*. Technischer Bericht, Technische Universität Wien.
- Benáček, Vladimír; M. Jarolím und Jan Ámos Víšek (1998): Supply-side characteristics and the industrial structure of Czech foreign trade. *Proceedings of the conference Business and economic development in central and eastern Europe: Implications for economic integration into wider Europe*, S. 51–68. Technical University in Brno together with University of Wisconsin, Whitewater and the Nottingham Trent University.
- Caspary, Wilhelm (1988): Fehlerverteilungen, Methode der kleinsten Quadrate und Alternativen. *Zeitschrift für Vermessungswesen (ZfV)*, 113. Jahrgang(3):123–133.
- Čížek, Pavel und Jan Ámos Víšek (2000): Least trimmed squares. *Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes*, No. 2000:53. Discussion Papers.
- Collins, Robert (2007): Lecture 15 - Robust Estimation: RANSAC. <http://www.cse.psu.edu/~rcollins/CSE486/lecture15.pdf>. Vorlesungsfolien. CSE Department, Penn State University. Abgerufen am 12.03.2014.
- Fahrmeir, Ludwig; Thomas Kneib und Stefan Lang (2009): *Regression - Modelle, Methoden und Anwendungen*. Springer Verlag, 2. Aufl.
- Fischler, Martin A. und Robert C. Bolles (1981): Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.
- Hossainali, Masoud Mashhadi (2006): *A Comprehensive Approach to the Analysis of the 3D-Kinematics of Deformation*. Dissertation, Fachbereich Vermessungswesen, Technische Universität Darmstadt.
- Joeckel, Rainer und Manfred Stober (1991): *Elektronische Entfernung- und Richtungsmessung*. Verlag Konrad Wittwer, 2. Aufl.
- Jäger, Reiner; Tilman Müller; Heinz Saler und Rainer Schwäble (2005): *Klassische und robuste Ausgleichungsverfahren*. Herbert Wichmann Verlag, 1. Aufl.

- Kahmen, Heribert (2006): *Vermessungskunde*. Walter de Gruyter, 20. Aufl. S.170-180.
- Krarup, Torben; J. Juhl und K. Kubik (1980): Götterdämmerung over Least Squares Adjustment. *Proceedings XIV. Congress ISPRS, Hamburg, International Archives of Photogrammetry*, B3, Commission III:369–378.
- Kraus, Karl (1996): *Photogrammetrie*. Ferd. Dümmler Verlag, Bonn, 3. Aufl. Band 2.
- Leica (2014): LEICA Instrument EDM Specifications. [https://www.ucviden.dk/student-portal/files/11409964/BILAG\\_20\\_Leica\\_Instrument\\_parameters.pdf](https://www.ucviden.dk/student-portal/files/11409964/BILAG_20_Leica_Instrument_parameters.pdf). Abgerufen am 01.03.2014.
- Navratil, Gerhard (2008): Ausgleichsrechnung I oder Die Anwendung statistischer Methoden in Vermessungswesen und GIS. *Vorlesungsskriptum*. Department für Geodäsie und Geoinformation, Technische Universität Wien.
- Navratil, Gerhard (2011): Ausgleichsrechnung II. *Vorlesungsskriptum*. Department für Geodäsie und Geoinformation, Technische Universität Wien.
- Neitzel, Frank (2004): *Identifizierung konsistenter Datengruppen am Beispiel der Kongruenzuntersuchung geodätischer Netze*. Dissertation, Fakultät Bauingenieurwesen und Angewandte Geowissenschaften, Technische Universität Berlin.
- Niemeier, Wolfgang (2008): *Ausgleichsrechnung*. Walter de Gruyter, Berlin, 2. Aufl.
- Rousseeuw, Peter J. (1985): Multivariate Estimation With High Breakdown Point. *Mathematical Statistics and Applications*, vol. B:283–297. Edited by W. Grossmann, G. Pflug, I. Vincze and W. Werty, Reidel Publishing Company, Dordrecht.
- Rousseeuw, Peter J. und Katrien van Driessen (2006): Computing LTS Regression for Large Data Sets. *Data Mining and Knowledge Discovery*, 12:29–45.
- Rousseeuw, Peter J. und Annick M. Leroy (1987): *Robust Regression and Outlier Detection*. John Wiley and Sons, New York.
- Rüeger, Jean M. (1996): *Electronic Distance Measurement*. Springer, 4. Aufl.
- Schindler, Konrad (2011): RANSAC - highly robust estimation. [http://foto.hut.fi/opetus/295/pg\\_course2011/lectures/Schindler\\_helsinki\\_optional\\_ransac.pdf](http://foto.hut.fi/opetus/295/pg_course2011/lectures/Schindler_helsinki_optional_ransac.pdf). Vorlesungsfolien. Institut für Geodäsie und Photogrammetrie, ETH Zürich. Abgerufen am 13.03.2014.
- Schlossmacher, E. J. (1973): An Iterative Technique for Absolute Deviations Curve Fitting. *Journal of the American Statistical Association*, 68(344):857–859.
- Österreich Werbung Wien (2014): Interaktive Karte Österreich. <http://www.austria.info/at/interaktive-karte-oesterreich>. Abgerufen am 19.08.2014.
- Wicki, Fridolin (1998): *Robuste Schätzverfahren für die Parameterschätzung in geodätischen Netzen*. Dissertation, Institut für Geodäsie und Photogrammetrie, ETH Zürich.

Wieser, Andreas (2011): *Statistische Grundlagen*. Vorlesungsfolien, Department für Geodäsie und Geoinformation, Technische Universität Wien. S.5-9.

Wild-Heerbrugg (2014): Virtuelles Archiv von Wild Heerbrugg. [http://www.wild-heerbrugg.com/shop/index.php?cPath=1\\_3\\_5\\_29\\_48](http://www.wild-heerbrugg.com/shop/index.php?cPath=1_3_5_29_48). Abgerufen am 08.02.2014.

Witte, Berthold und Peter Sparla (2011): *Vermessungskunde und Grundlagen der Statistik für das Bauwesen*. Wichmann, 7. Aufl.

# Anhang

## A Datenfiles

In diesem Abschnitt sind die Messdaten in Form der Inputfiles zu finden.

### A.1 naeherung\_a.txt

1	11	-20629.91	5296245.76
	100	-18280.00	5298068.00
	114	-21239.35	5299087.38
	150	-19197.40	5296894.93
	151	-22728.15	5297191.77
6	152	-18958.94	5301272.04
	230	-18056.34	5296909.84
	233	-18185.52	5299063.71

### A.2 richtungen\_acorr.txt

	11	100	58.0093	258.0107	0.15	58.0102	1
2	11	100	58.0097	258.0107	0.15	58.0102	1
	11	114	386.5511	186.5501	0.15	386.5507	1
	11	151	326.9655	126.9640	0.15	326.9648	1
	11	100	58.0116	258.0116	0.15	58.0121	2
	11	100	58.0118	258.0124	0.15	58.0121	2
7	11	114	386.5535	186.5519	0.15	386.5531	2
	11	100	58.0124	258.0118	0.15	58.0120	3
	11	100	58.0117	258.0123	0.15	58.0120	3
	11	114	386.5524	186.5516	0.15	386.5520	3
	11	100	58.0124	258.0128	0.15	58.0120	4
12	11	100	58.0119	258.0121	0.15	58.0120	4
	11	114	386.5527	186.5518	0.15	386.5519	4
	11	151	326.9666	126.9657	0.15	326.9660	4
	100	230	188.0326	388.0337	0.5	188.0332	1
	100	230	188.0333	388.0330	0.5	188.0332	1
17	100	114	321.3796	121.3797	0.5	321.3797	1
	100	230	188.0336	388.0334	0.5	188.0332	2
	100	230	188.0329	388.0334	0.5	188.0332	2
	100	114	321.3794	121.3814	0.5	321.3802	2
	100	152	386.9681	186.9688	0.5	386.9683	2
22	100	230	188.0319	388.0324	0.5	188.0322	3
	100	230	188.0317	388.0330	0.5	188.0322	3
	100	151	287.8627	87.8631	0.5	287.8629	3
	100	230	188.0321	388.0329	0.5	188.0325	4
	100	230	188.0331	388.0329	0.5	188.0325	4
27	100	114	321.3790	121.3779	0.5	321.3784	4
	100	152	386.9664	186.9682	0.5	386.9671	4
	150	152	319.9066	119.9001	0.5	319.9032	1
	150	230	15.6101	215.6054	0.5	15.6072	1
	150	151	221.7835	21.7788	0.5	221.7812	2
32	150	151	221.7844	21.7788	0.5	221.7812	2
	150	151	221.7841	21.7808	0.5	221.7825	3
	150	151	221.7854	21.7794	0.5	221.7825	3
	150	152	319.9105	119.9045	0.5	319.9075	3
	150	151	221.7832	21.7793	0.5	221.7830	4
37	150	151	221.7838	21.7822	0.5	221.7830	4
	150	152	319.9101	119.9031	0.5	319.9078	4
	150	230	15.6165	215.6075	0.5	15.6124	4
	150	151	122.8509	322.8494	0.15	122.8502	5

42	150	151	122.8511	322.8492	0.15	122.8502	5
	150	114	169.7738	369.7720	0.15	169.7729	5
	150	152	220.9765	20.9752	0.15	220.9759	5
	150	151	122.8507	322.8495	0.15	122.8501	6
47	150	151	122.8509	322.8491	0.15	122.8501	6
	150	152	220.9767	20.9753	0.15	220.9760	6
	150	233	245.3037	45.3032	0.15	245.3035	6
	150	230	316.6797	116.6785	0.15	316.6792	6
52	150	151	122.8511	322.8494	0.15	122.8503	7
	150	151	122.8506	322.8492	0.15	122.8503	7
	150	114	169.7742	369.7724	0.15	169.7734	7
	150	152	220.9764	20.9750	0.15	220.9759	7
57	150	230	316.6786	116.6790	0.15	316.6791	7
	150	151	122.8510	322.8491	0.15	122.8501	8
	150	151	122.8500	322.8491	0.15	122.8501	8
	150	114	169.7736	369.7723	0.15	169.7731	8
62	150	152	220.9764	20.9754	0.15	220.9761	8
	150	233	245.3026	45.3028	0.15	245.3030	8
	150	230	316.6800	116.6786	0.15	316.6797	8
	230	233	396.2488	196.2436	0.5	396.2455	1
67	230	233	396.2480	196.2430	0.5	396.2455	1
	230	233	396.2474	196.2273	0.5	396.2343	2
	230	233	396.2413	196.2273	0.5	396.2343	2
	230	233	396.2429	196.2332	0.5	396.2372	3
72	230	233	396.2414	196.2330	0.5	396.2372	3
	230	100	387.8489	187.8390	0.5	387.8434	3
	230	233	396.2416	196.2342	0.5	396.2381	4
	230	233	396.2414	196.2347	0.5	396.2381	4
77	230	233	396.2408	196.2295	0.5	396.2318	5
	230	233	396.2395	196.2240	0.5	396.2318	5
	230	233	396.2386	196.2292	0.5	396.2297	6
	230	233	396.2382	196.2212	0.5	396.2297	6
82	230	233	256.6058	56.6057	0.15	256.6055	7
	230	233	256.6053	56.6056	0.15	256.6055	7
	230	233	256.6052	56.6055	0.15	256.6057	8
	230	233	256.6059	56.6054	0.15	256.6057	8
3	233	230	131.1784	331.1790	0.15	131.1788	1
	233	151	210.1052	10.1064	0.15	210.1057	1
	233	114	235.4814	35.4824	0.15	235.4821	1
	233	230	131.1793	331.1779	0.15	131.1785	2
8	233	151	210.1059	10.1066	0.15	210.1062	2
	233	114	235.4828	35.4834	0.15	235.4830	2
	11	151	2301.6652				
	11	151	2301.6652				
13	11	151	2301.6652				
	11	151	2301.6652				
	11	114	2906.2565				
	11	114	2906.2565				
18	11	114	2906.2575				
	11	114	2906.2565				
	11	100	2972.3845				
	11	100	2972.3835				
23	11	100	2972.3845				
	11	100	2972.3825				
	100	152	3275.8201				
	100	152	3275.8221				
28	100	152	3275.8211				
	100	152	3275.8211				
	100	114	3129.3464				
	100	114	3129.3433				
33	100	114	3129.3464				
	100	114	3129.3444				
	150	151	3543.2291				
	150	151	3543.2291				
38	150	151	3543.2291				
	150	151	3543.2291				
	150	151	3543.2301				
	150	114	2996.0813				
43	150	114	2996.0813				
	150	114	2996.0823				
	150	114	2996.0823				
	150	233	2393.2947				
48	150	233	2393.2907				
	150	233	2393.2947				
	150	233	2393.2977				
	150	230	1141.1632				
53	150	230	1141.1632				
	150	230	1141.1632				

### A.3 sGK\_acorr.txt

3	11	151	2301.6652				
	11	151	2301.6652				
	11	151	2301.6652				
	11	151	2301.6652				
8	11	114	2906.2565				
	11	114	2906.2565				
	11	114	2906.2575				
	11	114	2906.2565				
13	11	100	2972.3845				
	11	100	2972.3835				
	11	100	2972.3845				
	11	100	2972.3825				
18	100	152	3275.8201				
	100	152	3275.8221				
	100	152	3275.8211				
	100	152	3275.8211				
23	100	114	3129.3464				
	100	114	3129.3433				
	100	114	3129.3464				
	100	114	3129.3444				
28	150	151	3543.2291				
	150	151	3543.2291				
	150	151	3543.2291				
	150	151	3543.2291				
33	150	151	3543.2301				
	150	114	2996.0813				
	150	114	2996.0813				
	150	114	2996.0823				
38	150	114	2996.0823				
	150	233	2393.2947				
	150	233	2393.2907				
	150	233	2393.2947				
43	150	233	2393.2977				
	150	230	1141.1632				
	150	230	1141.1632				
	150	230	1141.1632				

	150	230	1141.1632
	150	230	1141.1632
38	230	233	2157.7934
	230	233	2157.7934
	230	233	2157.7934
	230	233	2157.7924
	230	114	3856.6179
	230	114	3856.6169
43	230	114	3856.6229
	230	114	3856.6158
	230	114	3856.6169
	230	114	3856.6148
	230	100	1178.9765
48	230	100	1178.9755
	230	100	1178.9785
	230	100	1178.9785
	230	100	1178.9785

#### A.4 strecken\_acorr.txt

	11	151	2319.059	17	16	931	907	1.554	642.70	0.235	925.94	-20629.91	-22728.15	1	1
	11	151	2319.059	17	16	931	907	1.554	642.70	0.235	925.94	-20629.91	-22728.15	1	1
	11	151	2319.059	17	16	931	907	1.554	642.70	0.235	925.94	-20629.91	-22728.15	1	1
4	11	151	2319.059	17	16	931	907	1.554	642.70	0.235	925.94	-20629.91	-22728.15	1	1
	11	114	2915.210	17	15	931	912	1.554	642.70	1.695	867.82	-20629.91	-21239.35	1	1
	11	114	2915.210	17	15	931	912	1.554	642.70	1.695	867.82	-20629.91	-21239.35	1	1
	11	114	2915.211	17	15	931	912	1.554	642.70	1.695	867.82	-20629.91	-21239.35	1	1
	11	114	2915.210	17	15	931	912	1.554	642.70	1.695	867.82	-20629.91	-21239.35	1	1
	11	100	2985.974	17	18.5	931	961	1.554	642.70	1.655	359.538	-20629.91	-18280.00	1	1
9	11	100	2985.973	17	18.5	931	961	1.554	642.70	1.655	359.538	-20629.91	-18280.00	1	1
	11	100	2985.974	17	18.5	931	961	1.554	642.70	1.655	359.538	-20629.91	-18280.00	1	1
	11	100	2985.972	17	18.5	931	961	1.554	642.70	1.655	359.538	-20629.91	-18280.00	1	1
	100	152	3310.878	9	8	976	922	1.724	359.538	0.235	839.93	-18280.00	-18958.94	3	2
14	100	152	3310.880	9	8	976	922	1.724	359.538	0.235	839.93	-18280.00	-18958.94	3	2
	100	152	3310.879	9	8	976	922	1.724	359.538	0.235	839.93	-18280.00	-18958.94	3	2
	100	152	3310.879	9	8	976	922	1.724	359.538	0.235	839.93	-18280.00	-18958.94	3	2
	100	114	3170.569	9	8.5	976	918	1.724	359.538	1.630	867.82	-18280.00	-21239.35	3	2
	100	114	3170.566	9	8.5	976	918	1.724	359.538	1.630	867.82	-18280.00	-21239.35	3	2
19	100	114	3170.569	9	8.5	976	918	1.724	359.538	1.630	867.82	-18280.00	-21239.35	3	2
	100	114	3170.567	9	8.5	976	918	1.724	359.538	1.630	867.82	-18280.00	-21239.35	3	2
	150	151	3585.464	16	11	968.5	908	0.238	378.88	0.24	925.94	-19197.40	-22728.15	3	2
	150	151	3585.464	16	11	968.5	908	0.238	378.88	0.24	925.94	-19197.40	-22728.15	3	2
	150	151	3585.464	16	11	968.5	908	0.238	378.88	0.24	925.94	-19197.40	-22728.15	3	2
24	150	151	3585.465	16	11	968.5	908	0.238	378.88	0.24	925.94	-19197.40	-22728.15	3	2
	150	114	3036.141	16	14	968	913.9	0.238	378.88	1.648	867.82	-19197.50	-21239.35	3	2
	150	114	3036.141	16	14	968	913.9	0.238	378.88	1.648	867.82	-19197.50	-21239.35	3	2
	150	114	3036.142	16	14	968	913.9	0.238	378.88	1.648	867.82	-19197.50	-21239.35	3	2
	150	114	3036.142	16	14	968	913.9	0.238	378.88	1.648	867.82	-19197.50	-21239.35	3	2
29	150	233	2393.679	16.5	16.5	968.5	956.5	0.238	378.88	1.545	415.07	-19197.50	-18185.52	3	2
	150	233	2393.675	16.5	16.5	968.5	956.5	0.238	378.88	1.545	415.07	-19197.50	-18185.52	3	2
	150	233	2393.679	16.5	16.5	968.5	956.5	0.238	378.88	1.545	415.07	-19197.50	-18185.52	3	2
	150	233	2393.682	16.5	16.5	968.5	956.5	0.238	378.88	1.545	415.07	-19197.50	-18185.52	3	2
	150	230	1143.998	16.5	16.5	968	959.7	0.238	378.88	1.630	457.299	-19197.50	-18056.34	3	2
34	150	230	1143.998	16.5	16.5	968	959.7	0.238	378.88	1.630	457.299	-19197.50	-18056.34	3	2
	150	230	1143.998	16.5	16.5	968	959.7	0.238	378.88	1.630	457.299	-19197.50	-18056.34	3	2
	150	230	1143.998	16.5	16.5	968	959.7	0.238	378.88	1.630	457.299	-19197.50	-18056.34	3	2
	230	233	2158.300	18.5	18.5	953	961	1.609	457.299	1.370	415.07	-18056.34	-18185.52	1	1
	230	233	2158.300	18.5	18.5	953	961	1.609	457.299	1.370	415.07	-18056.34	-18185.52	1	1
39	230	233	2158.300	18.5	18.5	953	961	1.609	457.299	1.370	415.07	-18056.34	-18185.52	1	1
	230	233	2158.299	18.5	18.5	953	961	1.609	457.299	1.370	415.07	-18056.34	-18185.52	1	1
	230	114	3878.680	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
	230	114	3878.679	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
	230	114	3878.685	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
44	230	114	3878.678	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
	230	114	3878.679	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
	230	114	3878.677	18.5	17	953	911	1.609	457.299	1.695	867.82	-18056.34	-21239.35	1	1
	230	100	1183.062	18.5	18.5	953	961	1.609	457.299	1.655	359.538	-18056.34	-18280.00	1	1
	230	100	1183.061	18.5	18.5	953	961	1.609	457.299	1.655	359.538	-18056.34	-18280.00	1	1
49	230	100	1183.064	18.5	18.5	953	961	1.609	457.299	1.655	359.538	-18056.34	-18280.00	1	1
	230	100	1183.064	18.5	18.5	953	961	1.609	457.299	1.655	359.538	-18056.34	-18280.00	1	1
	230	100	1183.064	18.5	18.5	953	961	1.609	457.299	1.655	359.538	-18056.34	-18280.00	1	1



## B Streckenkorrektur

```
%Einlesen der Inputdatei
strecken=textread('strecken.txt');

4  %Mittelung, Einheitswechsel von mbar auf Torr
druck_gemittelt=(strecken(:,6)+strecken(:,7))/2;
temp_gemittelt=(strecken(:,4)+strecken(:,5))/2;
druck_gemittelt_torr=druck_gemittelt/1.33322;

9  %%Meteorologische Reduktion
%Barrel und Sears
lambda=0.85;
nGr=(287.604 + 3*(1.6288/lambda^2)+5*(0.0136/lambda^4)+10^6)/10^6;

14 %Kohlrausch
alpha=0.003661;
E=9.426;
D=0.000662;
e=E;

19 for i=1:length(temp_gemittelt)
    nL(i,1)=1+(nGr-1)/(1+alpha*temp_gemittelt(i,1))*(druck_gemittelt_torr(i,1)/760)-((5.5*10^-8)/(1+alpha*temp_gemittelt(i,1)))*e;
end

24 %Normbezugswerte
tb=12;
pb=760;

nL_s=1+(nGr-1)/(1+alpha*tb)*(pb/760);

29 for i=1:length(temp_gemittelt)
    mmet(i,1)=nL_s/nL(i,1);
end

34 for i=1:length(mmet)
    D(i,1)=strecken(i,3)*mmet(i,1);
end

%%Geometrische Reduktion
39 k=0.13;
Breite=48.83887;
a=6377397.155;
b=6356078.963;
c=a^2/b;
44 e2=(a^2-b^2)/b^2;
V=sqrt(1+e2*(cos(Breite*(pi/180)))^2);
M=c/(V^3);
N=c/V;
R=sqrt(M*N);

49 for i=1:length(D)
    rK(i,1)=-k^2*(D(i,1)^3/(24*R^2));
end

54 SR=D+rK;

deltaH=abs((strecken(:,11)+strecken(:,10))-(strecken(:,9)+strecken(:,8)));

for i=1:length(D)
59 rH(i,1)=(sqrt((1-(deltaH(i,1)/SR(i,1))^2)/((1+(strecken(i,9)+strecken(i,8))/R)*(1+(strecken(i,10)+strecken(i,11))/R))))-1)*SR(i,1);
end

for i=1:length(D)
    rE(i,1)=(D(i,1)^3/(24*R^2));
64 end

S=SR+rH+rE;

for i=1:length(D)
69 sGK(i,1)=S(i,1)*(1+(1/(6*R^2))*(strecken(i,12)^2+2*strecken(i,12)*strecken(i,13)+strecken(i,13)^2));
end

%Ergebnis in Textdatei auslesen
exp=[strecken(:,1)';strecken(:,2)';sGK'];
74 fid=fopen('sGK.txt','w');
fprintf(fid,'%d %d %6.4f\n',exp);
fclose(fid);
```

## C Methode der kleinsten Quadrate

```
clc
clear all
close(gcf)

4
sGK_imp=textread('sGK_acorr.txt');
richtungen=textread('richtungen_acorr.txt');
str=textread('strecken_acorr.txt');
koord=textread('naeherung_a.txt'); %Standpunkte müssen absteigend sortiert sein

9
alle=[richtungen(:,1:2);sGK_imp(:,1:2)];
standpunkte_r =unique(richtungen(:,1));
gr=size(richtungen);

14
%Definitionen für die Zuordnung in der A-Matrix
verwendete_pkte=unique(alle);
index=(1:2:2*length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Koordinaten verwendet
ind=(1:1:length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Orientierungsunbekannten verwendet
Index=[verwendete_pkte index ind];

19
%Matrix mit Stand- u. Zielpunktsnr.
Such_y=alle(:,1:2);
Such_x=alle(:,1:2);

24
%Matrix erstellen, die Standpunkte und ihre Anzahl von Sätzen enthält
for i=1:length(standpunkte_r)
    [row]=find(richtungen(:,1)==standpunkte_r(i,1));
    bet=richtungen(row(1,1):row(length(row),1),7);
    anzahl=unique(bet);
29
    anzahl2=anzahl(length(anzahl),1);
    saetze(i,1)=standpunkte_r(i,1);
    saetze(i,2)=anzahl2;
end

34
%% Beobachtungsvektor L aufstellen
L_r=richtungen(:,6);
L=[L_r;sGK_imp(:,3)];

%% Genauigkeit (P-Matrix)
39
%verwendete Geräte mit unterschiedlicher Genauigkeit beachten
% s_genauigkeit=[sGK_imp(:,3) str(:,14:15)];
% for i=1:length(s_genauigkeit)
%     s_genauigkeit(i,4)=(s_genauigkeit(i,2))^2+((s_genauigkeit(i,1)/1000)*s_genauigkeit(i,3))^2*10^-6;
% end

44
% r_genauigkeit=richtungen(:,5);
% for i=1:length(r_genauigkeit)
%     r_genauigkeit(i,2)=(r_genauigkeit(i,1)/1000)^2;
% end

49
for i=1:length(sGK_imp(:,1))
    s_genauigkeit(i,4)=(5)^2+((sGK_imp(i,3)/1000)*2)^2*10^-6;
end

54
for i=1:length(richtungen)
    r_genauigkeit(i,2)=(0.5/1000)^2;
end

sigma2=[r_genauigkeit(:,2);s_genauigkeit(:,4)];
59
S11=eye(length(alle),length(alle));
for i=1:length(alle)
    S11(i,i)=sigma2(i);
end

64
sigma02=1;
Q11=(1/sigma02)*S11;
P=Q11^-1;

%% Unbekanntenvektor X0 erstellen
69
koord_t=koord(:,2:3)';
Koord=reshape(koord_t, 2*length(koord),1);

iteration=0;
max_diff_k=1;
74
max_diff_o=1;

while max_diff_k>0.0001 && max_diff_o>0.00001
    iteration=iteration+1;
79
```

```

%% L0 aufstellen
%Näherungen für Orientierungen berechnen
Such_y_k=Such_y;
Such_x_k=Such_x;
84

%y-Koordinaten er- bzw. einsetzen
for j=1:length(Index)
    ii=Such_y==Index(j,1);
    Such_y_k(ii)=koord(j,2);
89
end

%x-Koordinaten er- bzw. einsetzen
for j=1:length(Index)
    ii=Such_x==Index(j,1);
94
    Such_x_k(ii)=koord(j,3);
end

naeherung_r=[Such_y_k(1:gr(1,1),1) Such_x_k(1:gr(1,1),1) Such_y_k(1:gr(1,1),2) Such_x_k(1:gr(1,1),2)];

99
for i=1:gr(1,1)
    t(i,1)=naeherung_r(i,3)-naeherung_r(i,1);    %Zähler für Richtungswinkel
    t(i,2)=naeherung_r(i,4)-naeherung_r(i,2);    %Nenner für Richtungswinkel
end

104
    %Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:length(t)
    if t(i,1)> 0 && t(i,2)< 0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
    end
109
    if t(i,1)<0 && t(i,2)<0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
    end
    if t(i,1)<0 && t(i,2)>0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+400;
114
    end
    if t(i,1)>0 && t(i,2)>0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi);
    end
end

119
    if iteration==1    %1.Ausgleich: Näherungen für Orientierungsunbekannte berechnen
        for i=1:length(t)    %Berechnung der Einzelorientierungen
            o(i,1)=t(i,3)-L_r(i);
        end

124
        for i=1:length(o)
            if o(i,1)<0
                o(i,1)=o(i,1)+400;
            end
        end

129
        for i=1:length(o)    %für korrekte Berechnung des Mittelwerts
            if o(i,1)<0.9995
                o(i,1)=o(i,1)+400;
            end
        end

134
        end

m=1;
for i=1:length(saetze)
139
    for j=1:saetze(i,2)
        [row]=find(richtungen(:,1)==saetze(i,1));
        [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
144
        mittelo=mean(o(s(1,1):s(length(s)),1));
        X0_o(m,1)=mittelo;
        m=m+1;

        for k=s(1,1):s(length(s),1)
149
            o(k,2)=mittelo;
        end
    end
end

154
X0=[Koord;X0_o];

%Berechnung von L0 für Richtungen
L0_r=t(:,3)-o(:,2);    %gemittelte Orientierungen
else
159
    L0_r=t(:,3)-o_probe;    %alle weiteren Iterationen: Xd als Näherungswert
    X0=Xd;
end

```

```

164     for i=1:length(L0_r)
         if L0_r(i,1)<0
             L0_r(i,1)=L0_r(i,1)+400;
         else
             L0_r(i,1)=L0_r(i,1);
         end
     end
169
     % STRECKEN
     naeherung_s=[Such_y_k(gr(1,1)+1:length(alle),1) Such_x_k(gr(1,1)+1:length(alle),1) Such_y_k(gr(1,1)+1:length(alle),2)
                 Such_x_k(gr(1,1)+1:length(alle),2)];

174     for i=1:length(naeherung_s)
         L0_s(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
     end

     L0=[L0_r;L0_s];
     l=L-L0;
179     for i=1:length(L0)
         if l(i,1)==0 && L0(i,1)>=400
             l(i,1)=mod(L0(i,1),400);
         end
     end
184

     %%Aufstellen der A-Matrix
     %y-Koordinaten er- bzw. einsetzen für Ri & Str.
     Such_y_a=Such_y;
     Such_x_a=Such_x;
189

     for j=1:length(Index)
         ii=Such_y==Index(j,1);
         Such_y_a(ii)=koord(j,2);
     end
194

     %x-Koordinaten er- bzw. einsetzen für Ri & Str.
     for j=1:length(Index)
         ii=Such_x==Index(j,1);
         Such_x_a(ii)=koord(j,3);
     end
199

     naeherung_r=[Such_y_a(1:gr(1,1),1) Such_x_a(1:gr(1,1),1) Such_y_a(1:gr(1,1),2) Such_x_a(1:gr(1,1),2)]; %Struktur: [y-
     Koord.Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

     %Ableitungen nach Richtungen berechnen
204     for i=1:gr(1,1)
         ableitungen_r(i,1)=(-(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
             naeherung_r(i,1))^2))* (200/pi); %dR/dyS
         ableitungen_r(i,2)=(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
             naeherung_r(i,1))^2))* (200/pi); %dR/dxS
         ableitungen_r(i,3)=(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
             naeherung_r(i,1))^2))* (200/pi); %dR/dyZ
         ableitungen_r(i,4)=(-(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
             naeherung_r(i,1))^2))* (200/pi); %dR/dxZ
209     end

     naeherung_s=[Such_y_a(gr(1,1)+1:length(alle),1) Such_x_a(gr(1,1)+1:length(alle),1) Such_y_a(gr(1,1)+1:length(alle),2)
                 Such_x_a(gr(1,1)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

     %Ableitungen nach Strecken berechnen
214     for i=1:length(sGK_imp(:,1))
         ableitungen_s(i,1)=(naeherung_s(i,1)-naeherung_s(i,3))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
             naeherung_s(i,1))^2)); %dS/dyS
         ableitungen_s(i,2)=(naeherung_s(i,2)-naeherung_s(i,4))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
             naeherung_s(i,1))^2)); %dS/dxS
         ableitungen_s(i,3)=(naeherung_s(i,3)-naeherung_s(i,1))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
             naeherung_s(i,1))^2)); %dS/dyZ
         ableitungen_s(i,4)=(naeherung_s(i,4)-naeherung_s(i,2))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
             naeherung_s(i,1))^2)); %dS/dxZ
219     end

     %A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
     A=zeros(length(L),length(Koord));

224     % Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Standpunkt gehören
     for i=1:gr(1,1)
         for j=1:length(Index)
             if richtungen(i,1)==Index(j,1)
                 A(i,Index(j,2))=ableitungen_r(i,1); %Ableitung nach der y-Koordinate des Standpunktes einsetzen
229                 A(i,(Index(j,2)+1))=ableitungen_r(i,2); %Ableitung nach der x-Koordinate des Standpunktes einsetzen
             end
         end
     end
end

```

```

234 %Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:gr(1,1)
    for j=1:length(Index)
        if richtungen(i,2)==Index(j,1)
            A(i,Index(j,2))=ableitungen_r(i,3); %Ableitung nach der y-Koordinate des Zielpunktes einsetzen
            A(i,(Index(j,2)+1))=ableitungen_r(i,4); %Ableitung nach der x-Koordinate des Zielpunktes einsetzen
239        end
    end
end

% Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
244 for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,1)==Index(j,1)
            A(gr(1,1)+i,Index(j,2))=ableitungen_s(i,1);
            A(gr(1,1)+i,(Index(j,2)+1))=ableitungen_s(i,2);
249        end
    end
end

%Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
254 for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,2)==Index(j,1)
            A(gr(1,1)+i,Index(j,2))=ableitungen_s(i,3);
            A(gr(1,1)+i,(Index(j,2)+1))=ableitungen_s(i,4);
259        end
    end
end

%eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
264 saetze(1,3)=0;
for i=1:length(saetze)-1
    saetze(i+1,3)=saetze(i,3)+saetze(i,2); %Spaltenposition für die Ableitung nach der O.unbek. berechnen
end

269 A0=zeros(gr(1,1),length(X0_o));
for i=1:length(saetze)
    n=saetze(i,3);
    for j=1:saetze(i,2)
        [row]=find(richtungen(:,1)==saetze(i,1));
274 [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
        n=n+j;
        for k=s(1,1):s(length(s),1)
279 A0(k,n)=-1;
        end
        n=n-j;
    end
end

284 %A0 um Nullbereich für die Streckenbeobachtungen erweitern
A0=[A0;zeros(length(sGK_imp),length(X0_o))];

A=[A A0];

289 %Ränderungsmatrix G aufstellen
G1= repmat([0 1],1,length(Koord)/2);
G2= repmat([1 0],1,length(Koord)/2);
null=zeros(1,length(X0_o));
294 G=[G1 null;G2 null];

for i=1:2:length(Koord)
    G(3,i+1)=Koord(i,1);
end

299 for i=2:2:length(Koord)
    G(3,i-1)=-Koord(i,1);
end

304 null_N=zeros(3,3);
null_xk=zeros(3,1);
N=[A'*P*A G';G null_N];
xk=N^-1*[A'*P*1>null_xk];
x=xk(1:length(X0),1);

309 Xd=X0+x;
v=A*x-1;
Ld=L+v;

314

```

```

%Bilanz
n=length(alle);
u=length(X0);
d=3;
319
%Varianz der Gewichtseinheit a posteriori
s0=(v'*P*v)/(n-u+d);

%% Hauptprobe
324 Xd_koord=Xd(1:length(G1),1);
Xd_hilfe_r=reshape(Xd_koord,2,length(G1)/2);
Xd_hilfe_r=Xd_hilfe_r';
Xd_hilfe_r=[Index(1:length(G1)/2,1) Xd_hilfe_r];

329 %ausgeglichene Orientierungen den Standpunkten zuweisen
Xd_o=Xd(length(G1)+1:length(Xd),1);

m=1;
for i=1:length(saetze)
334   for j=1:saetze(i,2)
       [row]=find(richtungen(:,1)==saetze(i,1));
       [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
       offset=row(1,1)-1;
       s=s+offset;
339   for k=s(1,1):s(length(s),1)
       o_probe(k,1)=Xd_o(m,1);
       end
       m=m+1;
       end
344 end

Xd_probe_r_y=richtungen(:,1:2);
Xd_probe_r_x=richtungen(:,1:2);

349 %y-Koordinaten er- bzw. einsetzen
for i=1:length(richtungen(:,1))
   for j=1:length(Xd_hilfe_r)
       jj=Xd_probe_r_y==Xd_hilfe_r(j,1);
       Xd_probe_r_y(jj)=Xd_hilfe_r(j,2);
354   end
end

%x-Koordinaten er- bzw. einsetzen
for i=1:length(richtungen(:,1))
359   for j=1:length(Xd_hilfe_r)
       jj=Xd_probe_r_x==Xd_hilfe_r(j,1);
       Xd_probe_r_x(jj)=Xd_hilfe_r(j,3);
       end
end
364

%Zuordnung der ausgeglichenen Koordinaten zu den ausgeglichenen Orientierungen
Ld_probe_r=[Xd_probe_r_y(:,1) Xd_probe_r_x(:,1) Xd_probe_r_y(:,2) Xd_probe_r_x(:,2) o_probe(:,1)];

%Berechnung der Richtungen aus den ausgeglichenen Koord & Orientierungen
369 %Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:gr(1,1)
   Ld_HP_r(i,1)=Ld_probe_r(i,3)-Ld_probe_r(i,1);   %Zähler der Richtungswinkel
   Ld_HP_r(i,2)=Ld_probe_r(i,4)-Ld_probe_r(i,2);   %Nenner der Richtungswinkel
end
374

for i=1:length(Ld_HP_r)
   if Ld_HP_r(i,1)> 0 && Ld_HP_r(i,2)< 0
       Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;
   end
379   if Ld_HP_r(i,1)< 0 && Ld_HP_r(i,2)< 0
       Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;
   end
   if Ld_HP_r(i,1)< 0 && Ld_HP_r(i,2)> 0
       Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+400;
384   end
   if Ld_HP_r(i,1)> 0 && Ld_HP_r(i,2)> 0
       Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi);
   end
end
389

for i=1:gr(1,1)
   Ld_HP_r(i,4)=Ld_HP_r(i,3)-o_probe(i,1);
end

394 for i=1:length(Ld_HP_r)
   if Ld_HP_r(i,4)< 0
       Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;

```

```

    end
end
399
for i=1:length(Ld_HP_r)
    if Ld_HP_r(i,4)<0.9995
        Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;
    end
404
end

%%Strecken für HP vorbereiten
Xd_strecken=Ld(gr(1,1)+1:length(Ld),1);

409
Xd_probe_s_y=sGK_imp(:,1:2);
Xd_probe_s_x=sGK_imp(:,1:2);

%ausgegliche y-Koordinaten einsetzen
for i=1:length(sGK_imp(:,1))
414
    for j=1:length(Index)
        ll=Xd_probe_s_y==Xd_hilfe_r(j,1);
        Xd_probe_s_y(ll)=Xd_hilfe_r(j,2);
    end
end
419

%ausgegliche x-Koordinaten einsetzen
for i=1:length(sGK_imp(:,1))
    for j=1:length(Index)
424
        mm=Xd_probe_s_x==Xd_hilfe_r(j,1);
        Xd_probe_s_x(mm)=Xd_hilfe_r(j,3);
    end
end

Ld_probe_s=[Xd_probe_s_y(:,1) Xd_probe_s_x(:,1) Xd_probe_s_y(:,2) Xd_probe_s_x(:,2) Xd_strecken];
429

for i=1:length(Xd_strecken)
    Ld_HP_s(i,1)=sqrt((Ld_probe_s(i,3)-Ld_probe_s(i,1))^2 + (Ld_probe_s(i,4)-Ld_probe_s(i,2))^2);
end

434
for i=1:gr(1,1)
    if Ld(i,1)<0
        Ld(i,1)=Ld(i,1)+400;
    else
        Ld(i,1)=Ld(i,1);
439
    end
end

for i=1:gr(1,1)
    %Anpassungen für dieses Netz
444
    if Ld(i,1)<0.9995 && Ld_HP_r(i,4)>390.9928
        Ld_p(i,1)=Ld(i,1)+400;
    elseif Ld(i,1)>390.9928 && Ld_HP_r(i,4)<0.9995
        Ld_p(i,1)=abs(Ld(i,1)-400);
    else
        Ld_p(i,1)=Ld(i,1);
449
    end
end

HP=[Ld_p;Ld(gr(1,1)+1:length(Ld),1)]-[Ld_HP_r(:,4);Ld_HP_s];
454
HPabs=abs(HP);
maximum=max(HPabs);
maximum_r=max(HPabs(1:gr(1,1),1));
maximum_s=max(HPabs(gr(1,1)+1:length(HP),1));

%Widerspruchsvektor
459
for i=1:length(G1)/2
    dxi(i,1)=x(2*i);
end

dyi=zeros(length(G1),1);
464
for i=2:2:length(G1)
    dyi(i-1,1)=x(i-1);
end
dyi=dyi(dyi~=0);

469
summexi=sum(dxi); %Bedingung 1: Summe aller Zuschläge zu den x-Näherungskoordinaten = 0
summeyi=sum(dyi); %Bedingung 2: Summe aller Zuschläge zu den y-Näherungskoordinaten = 0

for i=1:length(dxi)
474
    bed3(i,1)=Xd_hilfe_r(i,2)*dxi(i,1) - Xd_hilfe_r(i,3)*dyi(i,1);
end

summebed3=sum(bed3); %Bedingung 3
w=[summexi;summeyi;summebed3];

```

```

479     %Probe 2
k=xk (length(X0)+1:length(xk),1);
links=A'*P*v;
rechts=-G'*k;
erg=links-rechts;      %soll 0 ergeben
484
    %Probe 3
links2=v'*P*v;
rechts2=w'*k-l'*P*A*x + l'*P*l;
erg2=links2-rechts2;  %soll 0 ergeben
489
    if iteration==1      %Ausgeglichene Unbekannte archivieren
        Xd_sammel(:,1)=Xd;
    else
494        Xd_sammel(:,iteration)=Xd;
        diff_Xd=abs(Xd_sammel(:,iteration-1)-Xd);
        max_diff_k=max(diff_Xd(1:length(G1),1));      %Werte für Abbruchkriterium aktualisieren
        max_diff_o=max(diff_Xd(length(G1):length(Xd),1));
    end
499    koord=Xd_hilfe_r;      %Startwerte für neue Iteration
        Koord=Xd(1:length(G1));
end

```



## D L1-Norm

```
clc
clear all

3
tic          %Start für Zeitmessung der Rechendauer
sGK_imp=textread('sGK_acorr.txt');
richtungen=textread('richtungen_acorr.txt. ');
richtungen(:,8)=1:length(richtungen(:,1));
8
sGK_imp(:,4)=length(richtungen(:,1))+1:length(richtungen(:,1))+length(sGK_imp(:,1));

p_min=2;
iteration=0;
hilfe=1;     %Hilfsvariable für Auflistung der eliminierten Beobachtungen

13
while p_min < 1000
    iteration=iteration+1;

    if iteration > 1
18        sGK_imp=textread('sGK_acorr.txt');
        richtungen=textread('richtungen_acorr.txt. ');
        end

    %Definitionen für die Zuordnung in der A-Matrix
23    alle=[richtungen(:,1:2);sGK_imp(:,1:2)];
    standpunkte_r =unique(richtungen(:,1));
    allebeob=[richtungen;sGK_imp zeros(length(sGK_imp(:,1)),4)];
    verwendete_pkte=unique(alle);
    index=(1:2:2*length(verwendete_pkte)); %für Positionierung der Ableitung nach den Koordinaten verwendet
28    ind=(1:1:length(verwendete_pkte)); %für Positionierung der Ableitung nach den Orientierungsunbekannten verwendet
    Index=[verwendete_pkte index ind];
    koord=textread('naeherung_a.txt'); %Standpunkte müssen absteigend sortiert sein

    %Matrix mit Stand- u. Zielpunktsnr.
33    Such_y=alle(:,1:2);
    Such_x=alle(:,1:2);

    %Vektorgrößen definieren
38    gr=size(richtungen);
    gs=size(sGK_imp);

    %Matrix erstellen, die die Standpunkte und ihre Anzahl von Sätzen zeigt
    for i=1:length(standpunkte_r)
43        [row]=find(richtungen(:,1)==standpunkte_r(i,1));
        bet=richtungen(row(1,1):row(length(row),1),7);
        anzahl=unique(bet);
        anzahl2=anzahl(length(anzahl),1);
        saetze(i,1)=standpunkte_r(i,1);
        saetze(i,2)=anzahl2;
48    end

    %% Beobachtungsvektor L aufstellen
    L_r=richtungen(:,6);
    L=[L_r;sGK_imp(:,3)];
53

    %% L0 aufstellen
    %Näherungen für Orientierungen berechnen
    %Richtungswinkel berechnen
58    Such_y_k=Such_y;
    Such_x_k=Such_x;

    %y-Koordinaten er- bzw. einsetzen
    for j=1:length(Index)
63        ii=Such_y==Index(j,1);
        Such_y_k(ii)=koord(j,2);
    end

    %x-Koordinaten er- bzw. einsetzen
68    for j=1:length(Index)
        ii=Such_x==Index(j,1);
        Such_x_k(ii)=koord(j,3);
    end

    naeherung_r=[Such_y_k(1:gr(1,1),1) Such_x_k(1:gr(1,1),1) Such_y_k(1:gr(1,1),2) Such_x_k(1:gr(1,1),2)];
73

    for i=1:gr(1,1)
        t(i,1)=naeherung_r(i,3)-naeherung_r(i,1); %Zähler für Richtungswinkel tSZ
        t(i,2)=naeherung_r(i,4)-naeherung_r(i,2); %Nenner für Richtungswinkel tSZ
78    end
end
```

```

for i=1:length(t)                                     %Berechnung der Richtungswinkel mit Quadrantenzuordnung
    if t(i,1) > 0 && t(i,2) < 0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
83    end
    if t(i,1) < 0 && t(i,2) < 0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
88    end
    if t(i,1) < 0 && t(i,2) > 0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+400;
93    end
    if t(i,1) > 0 && t(i,2) > 0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi);
    end
end

for i=1:length(t)                                     %Berechnung der Einzelorientierungen
    o(i,1)=t(i,3)-L_r(i);
98    end

for i=1:length(o)
    if o(i,1) < 0
        o(i,1)=o(i,1)+400;
103    end
end

for i=1:length(o)                                     %für korrekte Berechnung des Mittelwerts
    if o(i,1) < 0.9995
        o(i,1)=o(i,1)+400;
108    end
end

m=1;
for i=1:length(saetze)
113    for j=1:saetze(i,2)
        [row]=find(richtungen(:,1)==saetze(i,1));
        [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
118

        mittelo=mean(o(s(1,1):s(length(s),1)));
        X0_o(m,1)=mittelo;
        m=m+1;

123        for k=s(1,1):s(length(s),1)
            o(k,2)=mittelo;
        end

    end
128    end

%Berechnung von L0 für Richtungen
L0_r=t(:,3)-o(:,2);                                %gemittelte Orientierungen

133

for i=1:length(L0_r)
    if L0_r(i,1) < 0
        L0_r(i,1)=L0_r(i,1)+400;
138    else
        L0_r(i,1)=L0_r(i,1);
    end
end

% STRECKEN
naeherung_s=[Such_y_k(gr(1,1)+1:length(alle),1) Such_x_k(gr(1,1)+1:length(alle),1) Such_y_k(gr(1,1)+1:length(alle),2)
143    Such_x_k(gr(1,1)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

for i=1:length(naeherung_s)
    L0_s(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
148    end

L0=[L0_r;L0_s];

%% Unbekanntenvektor X0 erstellen
koord_t=koord(:,2:3)';
153    Koord=reshape(koord_t, 2*length(koord),1);
    X0=[Koord;X0_o];

w=zeros(3,1);
c=1*10^-9;
158    maximum=1;
    diff_v=1;
    iteration_s=1;

```

```

while diff_v>0.00001
163   l=L-L0;
   for i=1:length(L0)
       if L(i,1)==0 && L0(i,1)>=400
           l(i,1)=mod(L0(i,1),400);
       end
   end
168
   %% P-Matrix aufstellen
   if iteration_s==1
       P=eye(length(alle),length(alle));
   else
173   P=zeros(length(alle),length(alle));
       for i=1:gr(1,1)
           P(i,i)=1/(abs(v(i,1))+c);
       end
       for i=gr(1,1)+1:length(alle)
178   P(i,i)=1/(abs(v(i,1)/10)+c);
       end
   end

   for i=1:length(alle)           %listet alle Gewichte für jeden Iterationsschritt auf
183   p(i,iteration_s)=P(i,i);
   end

   %%Aufstellen der A-Matrix
   %y-Koordinaten er- bzw. einsetzen für Ri & Str.
188   Such_y_a=Such_y;
   Such_x_a=Such_x;

   for j=1:length(Index)
       ii=Such_y==Index(j,1);
       Such_y_a(ii)=koord(j,2);
193   end

   %x-Koordinaten er- bzw. einsetzen für Ri & Str.
   for j=1:length(Index)
198   ii=Such_x==Index(j,1);
       Such_x_a(ii)=koord(j,3);
   end

   naeherung_r=[Such_y_a(1:gr(1,1),1) Such_x_a(1:gr(1,1),1) Such_y_a(1:gr(1,1),2) Such_x_a(1:gr(1,1),2)]; %Struktur: [y-
203   Koord.Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

   %Ableitungen nach Richtungen berechnen
   for i=1:gr(1,1)
       ableitungen_r(i,1)=(-(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i
       ,3)-naeherung_r(i,1))^2))*(200/pi); %dR/dyS
       ableitungen_r(i,2)=(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)
       -naeherung_r(i,1))^2))*(200/pi); %dR/dxS
208   ableitungen_r(i,3)=(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)
       -naeherung_r(i,1))^2))*(200/pi); %dR/dyZ
       ableitungen_r(i,4)=(-(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i
       ,3)-naeherung_r(i,1))^2))*(200/pi); %dR/dxZ
   end

   naeherung_s=[Such_y_a(length(richtungen)+1:length(alle),1) Such_x_a(length(richtungen)+1:length(alle),1) Such_y_a(
       length(richtungen)+1:length(alle),2) Such_x_a(length(richtungen)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt
213   . x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

   %Ableitungen nach Strecken berechnen
   for i=1:length(sGK_imp)
       ableitungen_s(i,1)=(naeherung_s(i,1)-naeherung_s(i,3))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(
       i,3)-naeherung_s(i,1))^2)); %dS/dyS
       ableitungen_s(i,2)=(naeherung_s(i,2)-naeherung_s(i,4))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(
       i,3)-naeherung_s(i,1))^2)); %dS/dxS
218   ableitungen_s(i,3)=(naeherung_s(i,3)-naeherung_s(i,1))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(
       i,3)-naeherung_s(i,1))^2)); %dS/dyZ
       ableitungen_s(i,4)=(naeherung_s(i,4)-naeherung_s(i,2))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(
       i,3)-naeherung_s(i,1))^2)); %dS/dxZ
   end

   %A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
223   A=zeros(length(L),length(Koord));

   % Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Standpunkt gehören
   for i=1:gr(1,1)
       for j=1:length(Index)
228   if richtungen(i,1)==Index(j,1)
           A(i,Index(j,2))=ableitungen_r(i,1); %Ableitung nach der y-Koordinate des Standpunktes einfüllen
           A(i,(Index(j,2)+1))=ableitungen_r(i,2); %Ableitung nach der x-Koordinate des Standpunktes einfüllen
       end
   end

```

```

233     end
end

%Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:gr(1,1)
    for j=1:length(Index)
238         if richtungen(i,2)==Index(j,1)
            A(i,Index(j,2))=ableitungen_r(i,3);           %Ableitung nach der y-Koordinate des Zielpunktes einsetzen
            A(i,(Index(j,2)+1))=ableitungen_r(i,4);       %Ableitung nach der x-Koordinate des Zielpunktes einsetzen
        end
    end
243 end

% Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
248         if sGK_imp(i,1)==Index(j,1)
            A(gr(1,1)+i,Index(j,2))=ableitungen_s(i,1);
            A(gr(1,1)+i,(Index(j,2)+1))=ableitungen_s(i,2);
        end
    end
253 end

%Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
258         if sGK_imp(i,2)==Index(j,1)
            A(gr(1,1)+i,Index(j,2))=ableitungen_s(i,3);
            A(gr(1,1)+i,(Index(j,2)+1))=ableitungen_s(i,4);
        end
    end
263 end

%eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
saetze(1,3)=0;
for i=1:length(saetze)-1
268     saetze(i+1,3)=saetze(i,3)+saetze(i,2);           %Spaltenposition für die Ableitung nach der Orientierungsunbekannten
                                                berechnen
end

A0=zeros(gr(1,1),length(X0_o));
for i=1:length(saetze)
273     n=saetze(i,3);
        for j=1:saetze(i,2)
            [row]=find(richtungen(:,1)==saetze(i,1));
            [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
278             offset=row(1,1)-1;
                s=s+offset;
                    n=n+j;
                        for k=s(1,1):s(length(s),1)
                            A0(k,n)=-1;
                        end
283                 n=n-j;
        end
end

%A0 um Nullbereich für die Streckenbeobachtungen erweitern
288 A0=[A0;zeros(length(sGK_imp),length(X0_o))];
A=[A A0];

%Ränderungsmatrix G aufstellen
G1=repmat([0 1],1,length(Koord)/2);
293 G2=repmat([1 0],1,length(Koord)/2);
null=zeros(1,length(X0_o));
G=[G1 null;G2 null];

for i=1:2:length(Koord)
298     G(3,i+1)=Koord(i,1);
end

for i=2:2:length(Koord)
303     G(3,i-1)=-Koord(i,1);
end

null_N=zeros(3,3);
null_xk=-w;
N=[A'*P*A G';G null_N];
308 xk=N^-1*[A'*P*1>null_xk];
x=xk(1:length(X0),1);
Xd=X0+x;
v=A*x-1;
Ld=L+v;

```

```

313      %Bilanz
      n=length(alle);
      u=length(X0);
      d=3;

318      %Varianz der Gewichtseinheit a posteriori
      s0=sqrt((v'*P*v)/(n-u+d));

      %% Hauptprobe
      Xd_koord=Xd(1:length(G1),1);
323      Xd_hilfe_r=reshape(Xd_koord,2,length(G1)/2);
      Xd_hilfe_r=Xd_hilfe_r';
      Xd_hilfe_r=[Index(1:length(G1)/2,1) Xd_hilfe_r];

      %ausgeglichene Orientierungen den Standpunkten zuweisen
328      Xd_o=Xd(length(G1)+1:length(Xd),1);

      m=1;
      for i=1:length(saetze)
          for j=1:saetze(i,2)
333              [row]=find(richtungen(:,1)==saetze(i,1));
              [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
              offset=row(1,1)-1;
              s=s+offset;
              for k=s(1,1):s(length(s),1)
338                  o_probe(k,1)=Xd_o(m,1);
              end
              m=m+1;
              end
          end

343      Xd_probe_r_y=richtungen(:,1:2);
      Xd_probe_r_x=richtungen(:,1:2);

      %y-Koordinaten er- bzw. einsetzen
348      for i=1:length(richtungen(:,1))
          for j=1:length(Xd_hilfe_r)
              jj=Xd_probe_r_y==Xd_hilfe_r(j,1);
              Xd_probe_r_y(jj)=Xd_hilfe_r(j,2);
          end
353      end

      %x-Koordinaten er- bzw. einsetzen
      for i=1:length(richtungen(:,1))
          for j=1:length(Xd_hilfe_r)
358              jj=Xd_probe_r_x==Xd_hilfe_r(j,1);
              Xd_probe_r_x(jj)=Xd_hilfe_r(j,3);
          end
          end

363      %Zuordnung der ausgeglichenen Koordinaten zu den ausgeglichenen Orientierungen
      Ld_probe_r=[Xd_probe_r_y(:,1) Xd_probe_r_x(:,1) Xd_probe_r_y(:,2) Xd_probe_r_x(:,2) o_probe(:,1)];

      %Berechnung der Richtungen aus den ausgeglichenen Koord & Orientierungen
      %Berechnung der Richtungswinkel mit Quadrantenzuordnung
368      for i=1:gr(1,1)
          Ld_HP_r(i,1)=Ld_probe_r(i,3)-Ld_probe_r(i,1);          %Zähler der Richtungswinkel
          Ld_HP_r(i,2)=Ld_probe_r(i,4)-Ld_probe_r(i,2);          %Nenner der Richtungswinkel
          end

373      for i=1:length(Ld_HP_r)
          if Ld_HP_r(i,1) > 0 && Ld_HP_r(i,2) < 0
              Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;
          end
          if Ld_HP_r(i,1) < 0 && Ld_HP_r(i,2) < 0
378              Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;
          end
          if Ld_HP_r(i,1) < 0 && Ld_HP_r(i,2) > 0
              Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+400;
          end
383      if Ld_HP_r(i,1) > 0 && Ld_HP_r(i,2) > 0
          Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi);
          end
          end

388      for i=1:gr(1,1)
          Ld_HP_r(i,4)=Ld_HP_r(i,3)-o_probe(i,1);
          end

      for i=1:length(Ld_HP_r)
393          if Ld_HP_r(i,4) < 0
              Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;
          end
      end

```

```

end
end
398 for i=1:length(Ld_HP_r)
    if Ld_HP_r(i,4)<0.9995
        Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;
    end
end
403 %%Strecken für HP vorbereiten
Xd_strecken=Ld(gr(1,1)+1:length(Ld),1);

Xd_probe_s_y=sGK_imp(:,1:2);
408 Xd_probe_s_x=sGK_imp(:,1:2);

%ausgeglichene y-Koordinaten einsetzen
for i=1:length(sGK_imp(:,1))
    for j=1:length(Index)
413         ll=Xd_probe_s_y==Xd_hilfe_r(j,1);
            Xd_probe_s_y(ll)=Xd_hilfe_r(j,2);
        end
    end

418 %ausgeglichene x-Koordinaten einsetzen
for i=1:length(sGK_imp(:,1))
    for j=1:length(Index)
        mm=Xd_probe_s_x==Xd_hilfe_r(j,1);
423         Xd_probe_s_x(mm)=Xd_hilfe_r(j,3);
    end
end

Ld_probe_s=[Xd_probe_s_y(:,1) Xd_probe_s_x(:,1) Xd_probe_s_y(:,2) Xd_probe_s_x(:,2) Xd_strecken];

428 for i=1:length(Xd_strecken)
    Ld_HP_s(i,1)=sqrt((Ld_probe_s(i,3)-Ld_probe_s(i,1))^2 + (Ld_probe_s(i,4)-Ld_probe_s(i,2))^2);
end

for i=1:gr(1,1)
433     if Ld(i,1)<0
        Ld(i,1)=Ld(i,1)+400;    %Richtigstellung der verbesserten Richtungen
    else
        Ld(i,1)=Ld(i,1);
    end
438 end

for i=1:gr(1,1)
    if Ld(i,1)<0.9995 && Ld_HP_r(i,4)>390.9928    %Anpassungen für dieses Netz
        Ld_p(i,1)=Ld(i,1)+400;
443     elseif Ld(i,1)>390.9928 && Ld_HP_r(i,4)<0.9995
        Ld_p(i,1)=abs(Ld(i,1)-400);
    else
        Ld_p(i,1)=Ld(i,1);
    end
448 end

HP=[Ld_p;Ld(gr(1,1)+1:length(Ld),1)]-[Ld_HP_r(:,4);Ld_HP_s];
HPabs=abs(HP);
maximum=max(HPabs);
453 maximum_r=max(HPabs(1:gr(1,1),1));
maximum_s=max(HPabs(gr(1,1)+1:length(HP),1));

%Widerspruchsvektor
for i=1:length(G1)/2
458     dxi(i,1)=x(2*i);
end

dyi=zeros(length(G1),1);
for i=2:2:length(G1)
463     dyi(i-1,1)=x(i-1);
end
dyi=dyi+(dyi~=0);

summexi=sum(dxi);    %Bedingung 1: Summe aller Zuschläge zu den x-Näherungskoordinaten = 0
468 summeysi=sum(dyi);    %Bedingung 2: Summe aller Zuschläge zu den y-Näherungskoordinaten = 0

for i=1:length(dxi)
    bed3(i,1)=Xd_hilfe_r(i,2)*dxi(i,1) - Xd_hilfe_r(i,3)*dyi(i,1);
473 end

summebed3=sum(bed3);    %Bedingung 3
w=[summexi;summeysi;summebed3];

```

```

478     %Probe 2
k=xk (length(X0)+1:length(xk),1);
links=A'*P*v;
rechts=-G'*k;
erg=links-rechts;    %soll 0 ergeben

483     %Probe 3
links2=v'*P*v;
rechts2=w'*k-l'*P*A*x + l'*P*1;
erg2=links2-rechts2;    %soll 0 ergeben

488     v_abs=abs(v);

if iteration_s==1
    diff_v=max(v_abs);
    v_alt(1:length(v),1)=v;
493     Xd_alt(:,1)=Xd;
else
    v_alt(1:length(v),iteration_s)=v;
    diff_v=max(abs(v_alt(:,iteration_s)-v_alt(:,iteration_s-1)));
    Xd_alt(:,iteration_s)=Xd;
498     end
iteration_s=iteration_s+1;
end

gp=size(p);
503     p_min=min(p(:,gp(1,2)));
if p_min < 1000
    [zeile]=find(p(:,gp(1,2))==p_min);
    allebeob(zeile(1,1),:)=[];

508     if zeile(1,1)<= gr(1,1)
        gr(1,1)=gr(1,1)-1;
        eliminiert(hilfe,1)=richtungen(zeile(1,1),8);
    else
        eliminiert(hilfe,1)=sGK_imp(zeile(1,1)-length(richtungen(:,1)),4);
513     end

richtungen=allebeob(1:gr(1,1),:);
sGK_imp=allebeob(gr(1,1)+1:length(allebeob(:,1)),1:4);

518     %Übrige Messungen in Textdatei auslesen
fid=fopen('sGK_acorr.txt','w');
fprintf(fid,'%d %d %6.4f %d\n',sGK_imp);
fclose(fid);

523     fid=fopen('richtungen_acorr.txt','w');
fprintf(fid,'%d %d %6.4f %6.4f %6.4f %6.4f %d %d\n',richtungen);
fclose(fid);

hilfe=hilfe+1;

528     clearvars t o_probe p L0_s Ld_p Ld Ld_HP_r Ld_HP_s    %Variablen für neue Iteration löschen bzw. reseten
end

533     eliminiert_sort=sortrows(eliminiert,1) %Zeilennr. der eliminierten Beobachtungen am Bildschirm ausgeben
toc    %Ende der Zeitmessung

```

## E Biber-Schätzer

```
1  clc
   clear all

   tic           %Start für Zeitmessung der Rechendauer
   sGK_imp=textread('sGK_acorr.txt');
6  richtungen=textread('richtungen_acorr.txt');
   %str=textread('strecken_acorr.txt');
   koord=textread('naeherung_a.txt'); %Standpunkte müssen absteigend sortiert sein

   alle=[richtungen(:,1:2);sGK_imp(:,1:2)];
11  gr=size(richtungen);

   %Definitionen für die Zuordnung in der A-Matrix
   verwendete_pkte=unique(alle);
   index=(1:2:2*length(verwendete_pkte)); %wird später für die Positionierung der Ableitung nach den Koordinaten benötigt
16  ind=(1:1:length(verwendete_pkte)); %wird später für die Positionierung der Ableitung nach der
      Orientierungsunbekannten verwendet
   Index=[verwendete_pkte index ind];

   %Matrix mit Stand- u. Zielpunktsnr.
   Such_y=alle(:,1:2);
21  Such_x=alle(:,1:2);

   %% Ausgleich nach der Methode der kleinsten Quadrate
   %% L aufstellen
   standpunkte_r =unique(richtungen(:,1));
26

   %Matrix erstellen, die die Standpunkte und ihre Anzahl von Sätzen zeigt
   for i=1:length(standpunkte_r)
       [row]=find(richtungen(:,1)==standpunkte_r(i,1));
       bet=richtungen(row(1,1):row(length(row),1),7);
31       anzahl=unique(bet);
           anzahl2=anzahl(length(anzahl),1);
           saetze(i,1)=standpunkte_r(i,1);
           saetze(i,2)=anzahl2;
   end
36

   L_r=richtungen(:,6);
   L=[L_r;sGK_imp(:,3)];

   %% L0 aufstellen
41  %Näherungen für Orientierungen berechnen
   %Richtungswinkel berechnen
   Such_y_k=Such_y;
   Such_x_k=Such_x;

46  %y-Koordinaten er- bzw. einsetzen
   for j=1:length(Index)
       ii=Such_y==Index(j,1);
       Such_y_k(ii)=koord(j,2);
   end
51

   %x-Koordinaten er- bzw. einsetzen
   for j=1:length(Index)
       ii=Such_x==Index(j,1);
       Such_x_k(ii)=koord(j,3);
56  end

   naeherung_r=[Such_y_k(1:gr(1,1),1) Such_x_k(1:gr(1,1),1) Such_y_k(1:gr(1,1),2) Such_x_k(1:gr(1,1),2)];
   for i=1:gr(1,1)
       t(i,1)=naeherung_r(i,3)-naeherung_r(i,1); %Zähler für Richtungswinkel tSZ
61  t(i,2)=naeherung_r(i,4)-naeherung_r(i,2); %Nenner für Richtungswinkel tSZ
   end

   %Berechnung der Richtungswinkel mit Quadrantenzuordnung
   for i=1:length(t)
66       if t(i,1) > 0 && t(i,2) < 0
           t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
       end
       if t(i,1) < 0 && t(i,2) < 0
           t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
71  end
       if t(i,1) < 0 && t(i,2) > 0
           t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+400;
       end
       if t(i,1) > 0 && t(i,2) > 0
76  t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi);
       end
   end
end
```



```

for i=1:length(t)           %Berechnung der Einzelorientierungen
81   o(i,1)=t(i,3)-L_r(i);
end

for i=1:length(o)
    if o(i,1)<0
86       o(i,1)=o(i,1)+400;
    end
end

for i=1:length(o)           %für korrekte Berechnung des Mittelwerts
91   if o(i,1)<0.9995
        o(i,1)=o(i,1)+400;
    end
end

m=1;
96   for i=1:length(saetze)
        for j=1:saetze(i,2)
            [row]=find(richtungen(:,1)==saetze(i,1));
            [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
            offset=row(1,1)-1;
101          s=s+offset;
            mittelo=mean(o(s(1,1):s(length(s)),1));
            X0_o(m,1)=mittelo;
            m=m+1;
            for k=s(1,1):s(length(s),1)
106              o(k,2)=mittelo;
            end
        end
    end

111 %Berechnung von L0
    % RICHTUNGEN
    L0_r=t(:,3)-o(:,2);

for i=1:length(L0_r)
116   if L0_r(i,1)<0
        L0_r(i,1)=L0_r(i,1)+400;
    else
        L0_r(i,1)=L0_r(i,1);
    end
121 end

% STRECKEN
naeherung_s=[Such_y_k(gr(1,1)+1:length(alle),1) Such_x_k(gr(1,1)+1:length(alle),1) Such_y_k(gr(1,1)+1:length(alle),2) Such_x_k
    (gr(1,1)+1:length(alle),2)];

126 for i=1:length(naeherung_s)
        L0_s(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
    end

L0=[L0_r;L0_s];
131

%% Genauigkeit (P-Matrix)
%verwendete Geräte mit unterschiedlicher Genauigkeit beachten
% s_genauigkeit=[sgk_imp(:,3) str(:,14:15)];
% for i=1:length(s_genauigkeit)
136 %     s_genauigkeit(i,4)=(s_genauigkeit(i,2))^2+((s_genauigkeit(i,1)/1000)*s_genauigkeit(i,3))^2*10^-6;
% end
% r_genauigkeit=richtungen(:,5);
% for i=1:length(r_genauigkeit)
141 %     r_genauigkeit(i,2)=(r_genauigkeit(i,1)/1000)*(pi/200))^2;
% end

for i=1:length(sgk_imp)
    s_genauigkeit(i,4)=(5)^2+((sgk_imp(i,3)/1000)*2)^2*10^-6;
end
146 for i=1:length(richtungen)
    r_genauigkeit(i,2)=(0.5/1000)^2;
end

sigma2=[r_genauigkeit(:,2);s_genauigkeit(:,4)];
151 S11=eye(length(alle),length(alle));

for i=1:length(alle)
    S11(i,i)=sigma2(i);
end

156 sigma02=1;
    Q11=(1/sigma02)*S11;
    P=Q11^-1;

```

```

161 %% Unbekanntenvektor X0 erstellen
koord_t=koord(:,2:3)';
Koord=reshape(koord_t, 2*length(koord),1);
X0=[Koord;X0_o];
166
%% gekürzter Beobachtungsvektor
l=L-L0;

%% Aufstellen der A-Matrix
171 for i=1:length(L0)
    if L(i,1)==0 && L0(i,1)>=400
        l(i,1)=mod(L0(i,1),400);
    end
end
176
%y-Koordinaten er- bzw. einsetzen für Ri & Str.
Such_y_a=Such_y;
Such_x_a=Such_x;

181 for j=1:length(Index)
    ii=Such_y==Index(j,1);
    Such_y_a(ii)=koord(j,2);
end

186 %x-Koordinaten er- bzw. einsetzen für Ri & Str.
for j=1:length(Index)
    ii=Such_x==Index(j,1);
    Such_x_a(ii)=koord(j,3);
end
191
naeherung_r=[Such_y_a(1:gr(1,1),1) Such_x_a(1:gr(1,1),1) Such_y_a(1:gr(1,1),2) Such_x_a(1:gr(1,1),2)];

%Ableitungen nach Richtungen berechnen
for i=1:gr(1,1)
196 ableitungen_r(i,1)=(-(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dyS
    ableitungen_r(i,2)=(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dxS
    ableitungen_r(i,3)=(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dyZ
    ableitungen_r(i,4)=(-(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dxZ
end
201
naeherung_s=[Such_y_a(length(richtungen)+1:length(alle),1) Such_x_a(length(richtungen)+1:length(alle),1) Such_y_a(length(
    richtungen)+1:length(alle),2) Such_x_a(length(richtungen)+1:length(alle),2)];

%Ableitungen nach Strecken berechnen
for i=1:length(sGK_imp)
206 ableitungen_s(i,1)=(naeherung_s(i,1)-naeherung_s(i,3))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dyS
    ableitungen_s(i,2)=(naeherung_s(i,2)-naeherung_s(i,4))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dxS
    ableitungen_s(i,3)=(naeherung_s(i,3)-naeherung_s(i,1))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dyZ
    ableitungen_s(i,4)=(naeherung_s(i,4)-naeherung_s(i,2))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dxZ
end
211
%A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
A=zeros(length(L),length(Koord));

% Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen
216 % Standpunkt gehören
for i=1:gr(1,1)
    for j=1:length(Index)
        if richtungen(i,1)==Index(j,1)
            A(i,Index(j,2))=ableitungen_r(i,1); %Ableitung nach der y-Koordinate des Standpunktes einfüllen
            A(i,(Index(j,2)+1))=ableitungen_r(i,2); %Ableitung nach der x-Koordinate des Standpunktes einfüllen
221        end
    end
end
226
%Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt
%gehören
for i=1:gr(1,1)
    for j=1:length(Index)
        if richtungen(i,2)==Index(j,1)
231            A(i,Index(j,2))=ableitungen_r(i,3); %Ableitung nach der y-Koordinate des Zielpunktes einsetzen
            A(i,(Index(j,2)+1))=ableitungen_r(i,4); %Ableitung nach der x-Koordinate des Zielpunktes einsetzen
        end
    end
end

```

```

        end
    end
end
236 % Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,1)==Index(j,1)
241             A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,1);
                A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,2);
        end
    end
end
246 %Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,2)==Index(j,1)
251             A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,3);
                A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,4);
        end
    end
end
256 %eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
saetze(1,3)=0;
for i=1:length(saetze)-1
    saetze(i+1,3)=saetze(i,3)+saetze(i,2);    %Spaltenposition für die Ableitung nach der 0.unbek. berechnen
261 end

A0=zeros(gr(1,1), length(X0_o));
for i=1:length(saetze)
    n=saetze(i,3);
266     for j=1:saetze(i,2)
        [row]=find(richtungen(:,1)==saetze(i,1));
        [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
271         n=n+j;
        for k=s(1,1):s(length(s),1)
            A0(k,n)=-1;
        end
        n=n-j;
276     end
end

%A0 um Nullbereich für die Streckenbeobachtungen erweitern
A0=[A0; zeros(length(sGK_imp), length(X0_o))];
281 A=[A A0];

%Ränderungsmatrix G aufstellen
G1= repmat([0 1], 1, length(Koord)/2);
G2= repmat([1 0], 1, length(Koord)/2);
286 null=zeros(1, length(X0_o));
G=[G1 null; G2 null];

for i=1:2:length(Koord)
    G(3, i+1)=Koord(i,1);
291 end
for i=2:2:length(Koord)
    G(3, i-1)=-Koord(i,1);
end

296 null_N=zeros(3,3);
null_xk=zeros(3,1);
N=[A'*P*A G'; G null_N];
xk=N^-1*[A'*P*1; null_xk];
x=xk(1:length(X0),1);
301

Xd_a=X0+x;
v=A*x-l;
Ld_a=L+v;

306 %% Beginn Biber-Schätzer
N_inv=N^-1;
Qxx=N_inv(1:length(X0),1:length(X0));
Ql1dd=A*Qxx*A';
Qvv=Q11-Ql1dd;
311 R=Qvv*Q11^-1;    %Redundanzanteile

sigma0=sqrt(sigma02);
v_rob=v;    %Ergebnisse von MdkQ als Startwerte

```

```

316 dd=zeros(length(v),1);      %Differenzvektor deltad erstellen
    c=3;

    for i=1:length(v)
        w(i,1)=v(i,1)/(sqrt(R(i,i))*sqrt(Q11(i,i))*sigma0);    %standardisierte Verb.
    end
321 for i=1:length(v)
        k(i,1)=c*sqrt(R(i,i))*sqrt(Q11(i,i))*sigma0;
    end

    w_abs=abs(w);
326 for i=1:length(v)
        if w_abs(i,1)>c
            test(i,1)=w_abs(i,1);
        else
            test(i,1)=0;
331     end
    end

    w_max=max(test);          %max. absolute standardisierte Verb.
    [nr]=find(w_abs(:,1)==w_max);    %Zeile, in der maximale standardisierte Verbesserung steht
336 snr=size(nr);
    if snr(1,1)>1
        nr=nr(1,1);    %wenn der maximale Wert mehr als 1x auftritt, die erste vorkommende Beobachtung mit diesem Wert verwenden
    end

341 iteration=0;

    while sum(test)~=0
        iteration=iteration+1;
        fehler(iteration,1)=nr;
346
        d=v_rob(nr,1)-sign(v_rob(nr,1))*k(nr,1);
        z=R(nr,nr);
        ddnr=zeros(length(v),1);
        ddnr(nr,1)=d/-z;
351     dd=dd+ddnr;    %alle angebrachten Modifikationen beachten

        N_rob=[A'*P*A G';G null_N];
        xk=N_rob^-1*[A'*P*dd>null_xk];
        x_rob=x-xk(1:length(X0),1);
356     v_rob=A*x_rob-(1-dd);
        v_tot=A*x_rob-1;

        for i=1:length(v)
            w_rob(i,1)=v_rob(i,1)/(sqrt(R(i,i))*sqrt(Q11(i,i))*sigma0);
361     end
        for i=1:length(v)
            w_tot(i,1)=v_tot(i,1)/(sqrt(R(i,i))*sqrt(Q11(i,i))*sigma0);
        end

366     w_abs=abs(w_rob);
        for i=1:length(v)
            if w_abs(i,1)>c
                test(i,1)=w_abs(i,1);
            else
371         test(i,1)=0;
            end
        end

        w_max=max(test);          %max. absolute standardisierte Verb.
        [nr]=find(w_abs(:,1)==w_max);    %Zeile, in der max. std.V. steht
376 snr=size(nr);
        if snr(1,1)>1
            nr=nr(1,1);    %wenn der maximale Wert mehr als 1x auftritt, die erste vorkommende Beobachtung mit diesem Wert verwenden
        end
        if abs(w_max-c)<0.0000001
381     test=0;
        end
    end

    if iteration==0
386     Xd=Xd_a;
        Ld=Ld_a;
    else
        Xd=X0+x_rob;
        Ld=L+v_tot;
391     fehleruni=unique(fehler);
    end

    anzR_alle=length(richtungen(:,1));
    fehleruni    %modifizierte Beobachtungen unabhängig von der Anzahl der Modifikation am Bildschirm ausgeben
396 toc        %Ende der Zeitmessung

```

## F RANSAC

Die Implementierung beinhaltet sowohl das Hauptprogramm, als auch die zwei, zum Zwecke der Übersichtlichkeit ausgelagerten, Funktionen „verbesserung5“ und „verbesserungen\_final“, welche sich nur in der Benennung der Variablen unterscheiden.

### F.1 Hauptprogramm

```
clc           %Hauptfenster leeren
clear all    %alle Variablen löschen
tic         %Start für Zeitmessung der Rechendauer
4
global wahr_sample p v_alle schwelle r_genauigkeit %globale Variablen (für ausgelagerte Funktionen)

%alle Beobachtungen einlesen
sgk_imp=textread('sgk_acorr.txt');
9 richtungen=textread('richtungen_acorr.txt. ');
koord=textread('naeherung_a.txt');
%str=textread('strecken_acorr.txt');

richtungen_1=richtungen; %Hilfsvariable für Auswahlverfahren
14 richtungen_1(:,8)=1:length(richtungen);
richtungen_0=richtungen;
richtungen_0(:,8)=1:length(richtungen);
alle=[richtungen(:,1:2) richtungen(:,7);sgk_imp(:,1:2) sgk_imp(:,4)];
alle(:,3)=1:length(alle);
19 alleR=richtungen; %alle Richtungen im Original speichern, für Berechnung der Verbesserungen
alleR2=richtungen; %alle Richtungen im Original speichern, für Berechnung der endgültigen Lösung
alleB=alle; %alle Beobachtungen im Original speichern, zum Reseten für neue Iteration

%Definitionen für die Zuordnung in der A-Matrix
24 verwendete_pkte=unique(alle(:,1:2));
index=(1:2:2*length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Koordinaten verwendet
ind=(1:1:length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Orientierungsunbekannten verwendet
Index=[verwendete_pkte index ind];

29 %Matrix mit Stand- u. Zielpunktsnr.
Such_y=alle(:,1:2);
Such_x=alle(:,1:2);

%Größen definieren
34 gr=size(richtungen);
anzr_alle=gr(1,1); %Anzahl aller Richtungsmessungen, für endgültige Lösung benötigt
beob=length(alle(:,1));
d=3;

39 %Matrix erstellen, die die Standpunkte und ihre Anzahl von Sätzen zeigt
for i=1:gstdpk_r(1,1)
    [row]=find(richtungen(:,1)==standpunkte_r(i,1));
    bet=richtungen(row(1,1):row(length(row),1),7);
    anzahl=unique(bet);
44    anzahl2=anzahl(length(anzahl),1);
    saetze(i,1)=standpunkte_r(i,1);
    saetze(i,2)=anzahl2;
end

49 %% L aufstellen
standpunkte_r =unique(richtungen(:,1));
gstdpk_r=size(standpunkte_r);

%Matrix erstellen, die die Standpunkte der RICHTUNGEN und ihre Anzahl von Sätzen zeigt
54 for i=1:gstdpk_r(1,1)
    [row]=find(richtungen(:,1)==standpunkte_r(i,1));
    bet_r=richtungen(row(1,1):row(length(row),1),7);
    anzahl_r=unique(bet_r);
    anzahl2_r=anzahl_r(length(anzahl_r),1);
59    saetze_r(i,1)=standpunkte_r(i,1);
    saetze_r(i,2)=anzahl2_r;
end

unbek=length(Index(:,1))*2+sum(saetze_r(:,2));
64 standpunkte_s =unique(sgk_imp(:,1));
gstdpk_s=size(standpunkte_s);

%Matrix erstellen, die die Standpunkte der STRECKEN und ihre Anzahl von Sätzen zeigt
69 for i=1:gstdpk_s(1,1)
    [row]=find(sgk_imp(:,1)==standpunkte_s(i,1));
```

```

bet_s=sGK_imp(row(1,1):row(length(row),1),4);
anzahl_s=unique(bet_s);
anzahl2_s=anzahl_s(length(anzahl_s),1);
saetze_s(i,1)=standpunkte_s(i,1);
74 saetze_s(i,2)=anzahl2_s;
end

wahl=unbek-sum(saetze_s(:,2))-sum(saetze_r(:,2)); %Anzahl der benötigten Beobachtungen für eine eindeutig bestimmte Lösung

79 L_r=richtungen(:,6);
L=[L_r;sGK_imp(:,3)];
L_alle=L; %für Berechnung der endgültigen Lösung
L_1=L; %Hilfsvariable für Auswahlverfahren
L_1(:,2)=1:length(L_1); %Nummerierung der Messungen in 2.Spalte
84 L_0=L;
L_0(:,2)=1:length(L_0); %Nummerierung der Messungen in 2.Spalte

%% L0 aufstellen
%Näherungen für Orientierungen berechnen
89 %Richtungswinkel berechnen
Such_y_k=Such_y;
Such_x_k=Such_x;

%y-Koordinaten er- bzw. einsetzen
94 for j=1:length(Index)
ii=Such_y==Index(j,1);
Such_y_k(ii)=koord(j,2);
end

99 %x-Koordinaten er- bzw. einsetzen
for j=1:length(Index)
ii=Such_x==Index(j,1);
Such_x_k(ii)=koord(j,3);
end
104 naeherung_r=[Such_y_k(1:gr(1,1),1) Such_x_k(1:gr(1,1),1) Such_y_k(1:gr(1,1),2) Such_x_k(1:gr(1,1),2)];

for i=1:gr(1,1)
t(i,1)=naeherung_r(i,3)-naeherung_r(i,1); %Zähler für Richtungswinkel tSZ
109 t(i,2)=naeherung_r(i,4)-naeherung_r(i,2); %Nenner für Richtungswinkel tSZ
end

%Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:length(t)
114 if t(i,1) > 0 && t(i,2) < 0
t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
end
if t(i,1) < 0 && t(i,2) < 0
119 t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
end
if t(i,1) < 0 && t(i,2) > 0
t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+400;
end
124 if t(i,1) > 0 && t(i,2) > 0
t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi);
end
end

t_alle=t; %für Berechnung der endgültigen Lösung
129 t_1=t; %Hilfsvariable für Eliminieren
t_1(:,4)=1:length(t); %Nummerierung der Messungen in 2.Spalte
t_0=t;
t_0(:,4)=1:length(t); %Nummerierung der Messungen in 2.Spalte

134 for i=1:length(t) %Berechnung der Einzelorientierungen
o(i,1)=t(i,3)-L_r(i);
end

for i=1:length(o)
139 if o(i,1) < 0
o(i,1)=o(i,1)+400;
end
end

144 for i=1:length(o) %für korrekte Berechnung des Mittelwerts
if o(i,1) < 0.9995
o(i,1)=o(i,1)+400;
end
end

149 o_alle=o; %für Berechnung der endgültigen Lösung
o_1=o; %Hilfsvariable für Auswahlverfahren

```

```

o_1(:,3)=1:length(o); %Numerierung der Messungen in 2.Spalte
o_0=0;
154 o_0(:,3)=1:length(o); %Numerierung der Messungen in 2.Spalte

m=1;
for i=1:length(saetze_r)
    for j=1:saetze_r(i,2)
159     [row]=find(richtungen(:,1)==saetze_r(i,1));
        [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
        mittelo=mean(o(s(1,1):s(length(s)),1));
164     X0_o(m,1)=mittelo;
        m=m+1;
        for k=s(1,1):s(length(s),1)
            o(k,2)=mittelo;
        end
169     end
end

%Berechnung von L0
% RICHTUNGEN
174 L0_r=t(:,3)-o(:,2);
for i=1:length(L0_r)
    if L0_r(i,1)<0
        L0_r(i,1)=L0_r(i,1)+400;
    else
179     L0_r(i,1)=L0_r(i,1);
    end
end

% STRECKEN
184 naeherung_s=[Such_y_k(gr(1,1)+1:length(alle),1) Such_x_k(gr(1,1)+1:length(alle),1) Such_y_k(gr(1,1)+1:length(alle),2) Such_x_k
    (gr(1,1)+1:length(alle),2)];

for i=1:length(naeherung_s)
    L0_s(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
end
189

L0=[L0_r;L0_s];
L0_alle=L0; %für Berechnung der endgültigen Lösung
L0_1=L0; %Hilfsvariable für Auswahlverfahren
L0_1(:,2)=1:length(L0); %Numerierung der Messungen in 2.Spalte
L0_0=L0;
194 L0_0(:,2)=1:length(L0); %Numerierung der Messungen in 2.Spalte

%% Genauigkeit (P-Matrix)
% verwendete Geräte mit unterschiedlicher Genauigkeit beachten
199 % s_genauigkeit=[sGK_imp(:,3) str(:,14:15)];
% for i=1:length(s_genauigkeit)
%     s_genauigkeit(i,4)=(s_genauigkeit(i,2))^2+((s_genauigkeit(i,1)/1000)*s_genauigkeit(i,3))^2)*10^-6;
% end
% r_genauigkeit=richtungen(:,5);
204 % for i=1:length(r_genauigkeit)
%     r_genauigkeit(i,2)=(r_genauigkeit(i,1)/1000)*(pi/200))^2;
% end

for i=1:length(sGK_imp)
209     s_genauigkeit(i,1)=(5)^2+((sGK_imp(i,3)/1000)*2)^2)*10^-6;
end
for i=1:length(richtungen)
    r_genauigkeit(i,1)=(0.5/1000)^2;
end
214

sigma2=[r_genauigkeit(:,1);s_genauigkeit(:,1)];
sigma2_alle=sigma2; %für Berechnung der endgültigen Lösung
sigma2_1=sigma2; %Hilfsvariable für Auswahlverfahren
sigma2_1(:,2)=1:length(sigma2); %Numerierung der Messungen in 2.Spalte
219 sigma2_0=sigma2;
sigma2_0(:,2)=1:length(sigma2); %Numerierung der Messungen in 2.Spalte

%% Unbekanntenvektor X0 erstellen
koord_t=koord(:,2:3)';
224 Koord=reshape(koord_t, 2*length(koord),1);

%% Ableitungen für die A-Matrix berechnen
%y-Koordinaten er- bzw. einsetzen für Ri & Str.
Such_y_a=Such_y;
229 Such_x_a=Such_x;

for j=1:length(Index)
    ii=Such_y==Index(j,1);

```

```

    Such_y_a(ii)=koord(j,2);
234 end

%x-Koordinaten er- bzw. einsetzen für Ri & Str.
for j=1:length(Index)
    ii=Such_x==Index(j,1);
239     Such_x_a(ii)=koord(j,3);
end
naeherung_r=[Such_y_a(1:gr(1,1),1) Such_x_a(1:gr(1,1),1) Such_y_a(1:gr(1,1),2) Such_x_a(1:gr(1,1),2)]; %Struktur: [y-Koord.
    Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

%Ableitungen nach Richtungen berechnen
244 for i=1:gr(1,1)
    ableitungen_r(i,1)=(-(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
        naeherung_r(i,1))^2))*(200/pi); %dR/dyS
    ableitungen_r(i,2)=(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
        naeherung_r(i,1))^2))*(200/pi); %dR/dxS
    ableitungen_r(i,3)=(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
        naeherung_r(i,1))^2))*(200/pi); %dR/dyZ
    ableitungen_r(i,4)=(-(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
        naeherung_r(i,1))^2))*(200/pi); %dR/dxZ
249 end

naeherung_s=[Such_y_a(length(richtungen)+1:length(alle),1) Such_x_a(length(richtungen)+1:length(alle),1) Such_y_a(length(
    richtungen)+1:length(alle),2) Such_x_a(length(richtungen)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt. x-Koord.
    Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

%Ableitungen nach Strecken berechnen
254 for i=1:length(sGK_imp)
    ableitungen_s(i,1)=(naeherung_s(i,1)-naeherung_s(i,3))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
        naeherung_s(i,1))^2)); %dS/dyS
    ableitungen_s(i,2)=(naeherung_s(i,2)-naeherung_s(i,4))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
        naeherung_s(i,1))^2)); %dS/dxS
    ableitungen_s(i,3)=(naeherung_s(i,3)-naeherung_s(i,1))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
        naeherung_s(i,1))^2)); %dS/dyZ
    ableitungen_s(i,4)=(naeherung_s(i,4)-naeherung_s(i,2))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
        naeherung_s(i,1))^2)); %dS/dxZ
259 end

%% Ränderungsmatrix G aufstellen
G1= repmat([0 1],1,length(Koord)/2);
G2= repmat([1 0],1,length(Koord)/2);
264 null=zeros(1,length(X0_o));
G=[G1 null;G2 null];

for i=1:2:length(Koord)
    G(3,i+1)=Koord(i,1);
269 end
for i=2:2:length(Koord)
    G(3,i-1)=-Koord(i,1);
end

274 null_N=zeros(3,3);
null_xk=zeros(3,1);

%% A-Matrix aufstellen
%A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
279 A=zeros(length(L),length(Koord));

% Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Standpunkt gehören
for i=1:gr(1,1)
    for j=1:length(Index)
284         if richtungen(i,1)==Index(j,1)
            A(i,Index(j,2))=ableitungen_r(i,1); %Ableitung nach der y-Koordinate des Standpunktes einfüllen
            A(i,(Index(j,2)+1))=ableitungen_r(i,2); %Ableitung nach der x-Koordinate des Standpunktes einfüllen
        end
    end
289 end

%Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:gr(1,1)
    for j=1:length(Index)
294         if richtungen(i,2)==Index(j,1)
            A(i,Index(j,2))=ableitungen_r(i,3); %Ableitung nach der y-Koordinate des Zielpunktes einsetzen
            A(i,(Index(j,2)+1))=ableitungen_r(i,4); %Ableitung nach der x-Koordinate des Zielpunktes einsetzen
        end
    end
299 end

% Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)

```



```

304     if sGK_imp(i,1)==Index(j,1)
        A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,1);
        A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,2);
    end
    end
309 end

%Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
314         if sGK_imp(i,2)==Index(j,1)
            A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,3);
            A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,4);
        end
    end
319 end

%eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
saetze(1,3)=0;
for i=1:length(saetze)-1
324     saetze(i+1,3)=saetze(i,3)+saetze(i,2);    %Spaltenposition für die Ableitung nach der 0.unbek. berechnen
end

A0=zeros(gr(1,1), length(X0_o));
for i=1:length(saetze)
329     n=saetze(i,3);
        for j=1:saetze(i,2)
            [row]=find(richtungen(:,1)==saetze(i,1));
            [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
            offset=row(1,1)-1;
334             s=s+offset;
                n=n+j;
                for k=s(1,1):s(length(s),1)
                    A0(k,n)=-1;
                end
339             n=n-j;
        end
    end

%A0 um Nullbereich für die Streckenbeobachtungen erweitern
344 A0=[A0;zeros(length(sGK_imp), length(X0_o))];
A=[A A0];
gA=size(A);
A_alle=A;    %für Berechnung der endgültigen Lösung
A_1=A;    %Hilfsvariable für Eliminieren
349 A_1(:, unbek+1)=1:length(A);    %Nummerierung der Messungen in 2.Spalte
A_0=A;
A_0(:, unbek+1)=1:length(A);    %Nummerierung der Messungen in 2.Spalte

iteration=1;
354 abbruch=0;    %Zähler für den Abbruch wegen nicht ausreichender Konditionierung

while iteration ~ =1001
    %1. Auswahlverfahren
    for i=1:length(saetze_r)
359         for j=1:saetze_r(i,2)
            [row]=find(richtungen(:,1)==saetze_r(i,1));    %Zeilen, die einen gemeinsamen Standpunkt haben
            [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);    %Zeilen, die einen gemeinsamen Satz haben
            offset=row(1,1)-1;
            s=s+offset;
364             abstand=s(1,1)-1;
                auswahl=randsample(length(s),1);    %1 RICHTUNG auswählen, die verwendet werden soll
                alle(auswahl+abstand,4)=1;    %ausgewählte RICHTUNG mit "1" markieren
            end
        end
369     end

    for i=1:length(saetze_s)
        for j=1:saetze_s(i,2)
            [row]=find(sGK_imp(:,1)==saetze_s(i,1));    %Zeilen, die einen gemeinsamen Standpunkt haben
            [s]=find(sGK_imp(row(1,1):row(length(row),1),4)==j);    %Zeilen, die einen gemeinsamen Satz haben
374             offset=anzR_alle + row(1,1)-1;
                s=s+offset;
                abstand=s(1,1)-1;
                auswahl=randsample(length(s),1);    %1 STRECKE auswählen, die verwendet werden soll
                alle(auswahl+abstand,4)=1;    %ausgewählte STRECKE mit "1" markieren
379             end
        end

    for i=0:length(alle)-1    %von der letzten zur ersten Messung, da so die falsche Zeilenzuordnung umgangen wird
        if alle(length(alle)-i,4)==0
384             A_1(length(alle)-i,:)=[];    %nicht ausgewählte "0" Beob. werden gelöscht
                sigma2_1(length(alle)-i,:)=[];
        end
    end
end

```

```

L_1(length(alle)-i,:)=[];
L0_1(length(alle)-i,:)=[];
else
389 A_0(length(alle)-i,:)=[]; %ausgewählte Beob. werden in gegenteiliger Matrix gelöscht
sigma2_0(length(alle)-i,:)=[];
L_0(length(alle)-i,:)=[];
L0_0(length(alle)-i,:)=[];
end
394 end

alle_r=alle(1:anzR_alle,:); %alleB aufteilen in Ri & Str
alle_s=alle(length(alle_r)+1:length(alle),:);

399 for i=0:anzR_alle-1
if alle_r(anzR_alle-i,4)==0
richtungen_1(anzR_alle-i,:)=[]; %nicht ausgewählte Ri löschen
o_1(anzR_alle-i,:)=[]; %dementsprechende Orientierungen
t_1(anzR_alle-i,:)=[]; %dementsprechende Richtungswinkel
404 else
richtungen_0(anzR_alle-i,:)=[]; %ausgewählte Beob. werden in gegenteiliger Matrix gelöscht
o_0(anzR_alle-i,:)=[];
t_0(anzR_alle-i,:)=[];
end
409 end

alle_rs_h=[alle_r;alle_s];
alle_rs=[alle_r;alle_s];
alle_rs_rest=[alle_r;alle_s]; %Matrix duplizieren, damit für weitere Auswahl verwendet werden kann
414

for i=0:length(alleB)-1
if alle_rs_h(length(alleB)-i,4)==0 %alle nicht ausgewählte Zeilen löschen
alle_rs(length(alleB)-i,:)=[];
else
419 alle_rs_rest(length(alleB)-i,:)=[]; %Matrix mit den verbleibenden Beob.
end
end

%% 2.Auswahlverfahren (3 Beobachtungen werden ausgewählt)
424 alle_rs_rest=alle_rs_rest(:,1:3); %letzte Spalte löschen, damit Beob. neu markiert werden können
auswahl=randsample(length(alle_rs_rest),wahl,false); %3 Beobachtungen auswählen damit insgesamt 42 Beob.; auswahl=
Zeilennr NICHT Beob.nr.

for i=1:length(auswahl)
alle_rs_rest(auswahl(i),4)=1; %neu ausgewählte Beob. markieren
429 end

for i=0:length(alle_rs_rest)-1 %von der letzten zur ersten Messung, weil so die falsche Zeilenzuordnung umgangen wird
if alle_rs_rest(length(alle_rs_rest)-i,4)==0
A_0(length(alle_rs_rest)-i,:)=[]; %nicht ausgewählte "0" Beob. werden gelöscht
434 sigma2_0(length(alle_rs_rest)-i,:)=[];
L_0(length(alle_rs_rest)-i,:)=[];
L0_0(length(alle_rs_rest)-i,:)=[];
end
end
439

aR=0; %zählen, wieviele von den nach der 1.Auswahl übrigen Beob. Ri sind
for i=1:length(alle_rs_rest)
if alle_rs_rest(i,3)<=anzR_alle %alle mit Zeilennr. <anzR_alle sind Richtungen
aR=aR+1;
444 end
end

rest_r=alle_rs_rest(1:aR,:);
rest_s=alle_rs_rest(aR+1:length(sigma2_0),:);
449

for i=0:length(rest_r)-1
if rest_r(length(rest_r)-i,4)==0
richtungen_0(length(rest_r)-i,:)=[]; %nicht ausgewählte Ri ausschneiden
o_0(length(rest_r)-i,:)=[]; %dementsprechende Orientierungen
454 t_0(length(rest_r)-i,:)=[]; %dementsprechende Richtungswinkel
end
end

%Wahl des 1. & 2.Auswahlverfahrens zusammenfügen und entsprechend Beob.nr. sortieren
459 A_sample=[A_1;A_0];
A_sample=sortrows(A_sample,unbek+1);
sigma2_sample=[sigma2_1;sigma2_0];
sigma2_sample=sortrows(sigma2_sample,2);
L0_sample=[L0_1;L0_0];
464 L0_sample=sortrows(L0_sample,2);
L_sample=[L_1;L_0];
L_sample=sortrows(L_sample,2);

```

```

o_sample=[o_1;o_0];
o_sample=sortrows(o_sample,3);
469 t_sample=[t_1;t_0];
t_sample=sortrows(t_sample,4);
richtungen_sample=[richtungen_1;richtungen_0];
richtungen_sample=sortrows(richtungen_sample,8);

474 %Zählen, wieviele von den ausgewählten Beob. Ri sind
aR=0;
for i=1:length(L0_sample)
    if L0_sample(i,2)<=anzR_alle %alle mit Zeilennr. <anzR_alle sind Richtungen
479         aR=aR+1;
    end
end

%L0_r mit den ausgewählten Ri neu berechnen
standpunkte_r_sample =unique(richtungen_sample(:,1));
484

%neue Matrix erstellen, die die verwendeten Standpunkte und ihre Anzahl von Sätzen zeigt
for i=1:length(standpunkte_r_sample)
    [row]=find(richtungen_sample(:,1)==standpunkte_r_sample(i,1));
    bet_sample=richtungen_sample(row(1,1):row(length(row),1),7);
489    anzahl_sample=unique(bet_sample);
    anzahl2_sample=anzahl_sample(length(anzahl_sample),1);
    saetze_sample(i,1)=standpunkte_r_sample(i,1);
    saetze_sample(i,2)=anzahl2_sample;
end

494 m=1;
for i=1:length(saetze_sample)
    for j=1:saetze_sample(i,2)
        [row]=find(richtungen_sample(:,1)==saetze_sample(i,1));
499        [s]=find(richtungen_sample(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
        mittelo_sample=mean(o_sample(s(1,1):s(length(s)),1));
504        X0_o_sample(m,1)=mittelo_sample;
        m=m+1;
        for k=s(1,1):s(length(s),1)
            o_sample(k,2)=mittelo_sample;
        end

509    end
end

%Berechnung von L0 für Richtungen
L0_r_sample=t_sample(:,3)-o_sample(:,2);
514 for i=1:length(L0_r_sample)
    if L0_r_sample(i,1)<0
        L0_r_sample(i,1)=L0_r_sample(i,1)+400;
    else
519        L0_r_sample(i,1)=L0_r_sample(i,1);
    end
end
L0_sample=[L0_r_sample;L0_sample(aR+1:length(L0_sample),1)];
l_sample=L_sample(:,1)-L0_sample(:,1);

524 %P-Matrix aufstellen
S11=eye(length(L_sample),length(L_sample));
for i=1:length(L_sample)
    S11(i,i)=sigma2_sample(i,1);
end
529 sigma02=1;
Q11=(1/sigma02)*S11;
P_sample=Q11^-1;

X0_sample=[Koord;X0_o_sample];
534 N_sample=[A_sample(:,1:unbek)'+P_sample*A_sample(:,1:unbek) G';G null_N];
condi=rcond(N_sample);

if condi<=10^-15
    disp('System nicht ausreichend konditioniert. Ergebnis ist nicht zuverlässig und wird verworfen.')
539    abbruch=abbruch+1;
else
    %Bestimmung der eindeutigen Lösung
    xk_sample=N_sample^-1*[A_sample(:,1:unbek)'+P_sample*l_sample>null_xk];
544    x_sample=xk_sample(1:length(X0_sample),1);

    Xd_sample=X0_sample+x_sample;
    v_sample=A_sample(:,1:unbek)*x_sample-l_sample;
    Ld_sample=L_sample(:,1)+v_sample;

```

```

549     samples(1:length(sigma2_sample),iteration)=sigma2_sample(:,2); %Beob.nr der ausgewählten Beob.

    verbesserungen5 (Xd_sample, sGK_imp, G1, anzR_alle, saetze, alleR, Index, alleB, alle, iteration); %Funktion zur
    Berechnung der Verbesserungen

    iteration=iteration+1;
554 end

    %%Matrizen, Vektoren für neuen Durchlauf reseten
    A_1=A;
    A_1(:,unbek+1)=1:length(A);
559 A_0=A;
    A_0(:,unbek+1)=1:length(A);
    sigma2_1=sigma2;
    sigma2_1(:,2)=1:length(sigma2);
    sigma2_0=sigma2;
564 sigma2_0(:,2)=1:length(sigma2);
    L0_1=L0;
    L0_1(:,2)=1:length(L0);
    L0_0=L0;
    L0_0(:,2)=1:length(L0);
569 o_1=o;
    o_1(:,3)=1:length(o);
    o_0=o;
    o_0(:,3)=1:length(o);
    t_1=t;
574 t_1(:,4)=1:length(t);
    t_0=t;
    t_0(:,4)=1:length(t);
    L_1=L;
    L_1(:,2)=1:length(L_1);
579 L_0=L;
    L_0(:,2)=1:length(L_0);
    richtungen_1=richtungen;
    richtungen_1(:,8)=1:length(richtungen);
    richtungen_0=richtungen;
584 richtungen_0(:,8)=1:length(richtungen);
    alle=alleB;
end

%% Endgültige Lösung mit unterstützenden Beobachtungen (_support) berechnen
589 maxp=max(p); %Iteration(en) mit der größten Wahrscheinlichkeit suchen
[zeilen]=find(p(:,1)==maxp); %in welcher Iteration tritt die größte Wahrscheinlichkeit auf?
gz=size(zeilen);
if gz(1,1)>1
    beste=zeilen(1,1); %wenn mehrere Iterationen mit größter Wahrscheinlichkeit, dann die erste auswählen
594 else
    beste=zeilen;
end

[support]=find(wahr_sample(:,beste)==1); %unterstützende Beobachtungen finden und deren Zeilen in "alle" anzeigen
599 iteration_l=0;
max_diff_k=1;
max_diff_o=1;

604 %Iterativer Ausgleich
while max_diff_k>0.0001 && max_diff_o>0.00001

    iteration_l=iteration_l+1;
    if iteration_l==1 %1.Ausgleich: Näherungen für Orientierungsunbekannte berechnen

609 %unterstützende Beobachtungen extrahieren
    alleB(:,4)=1:length(alleB); %4.Spalte in "alleB" mit Zeilenindex füllen, damit anhand diesem die unterstützenden
    extrahiert werden können
    for i=1:length(support)
        [row]=find(alleB(:,4)==support(i,1));
614 alleB(row,5)=1; %unterstützenden Beob. wird in der 4.Zeile ein "1" zugeordnet, den anderen ein "0"
    end

    for i=0:length(alleB)-1 %von der letzten zur ersten Messung, weil so die falsche Zeilenzuordnung umgangen wird
        if alleB(length(alleB)-i,5)==0
619 A_alle(length(alleB)-i,:)=[]; %nicht unterstützende "0" Beob. werden gelöscht
            sigma2_alle(length(alleB)-i,:)=[];
            L_alle(length(alleB)-i,:)=[];
            L0_alle(length(alleB)-i,:)=[];
        end
624 end

    A_support=A_alle;
    sigma2_support=sigma2_alle;
    L_support=L_alle;

```

```

629 L0_support=L0_alle;
A_support(:,~any(A_support))=[]; %Nullspalten, die durch Wegfallen von ganzen Richtungssätzen entstehen, löschen

%Anzahl der unterstützenden Richtungen berechnen. Dafür muss die letzte unterstützende Richtungsbeob. gefunden werden,
um anhand ihrer Zeilennr. alle unterstützenden Ri. zählen zu können
gefunden=0;
634 i=0;

while gefunden==0
    if alleB(anzR_alle-i,5)==0 %ausgehend von der letzten ursprünglichen Richtungsmessung, prüfen ob diese verwendet
        wird
            gefunden=0; %wenn nicht verwendet, dann "0"
639 i=i+1;
        else
            gefunden=1; %wenn verwendet, dann "1" und Abbruch der Schleife
            hilfe=anzR_alle-i; %Zeile anzeigen, in der die letzte verwendete Ri steht
        end
644 end
[anzR_support]=find(support==hilfe); %suchen, wo letzte Ri steht. zugehörige Zeile = Anzahl der unterstützenden Ri.
anzS_support=length(L_support)-anzR_support; %Anzahl der unterstützenden Strecken

for i=0:anzR_alle-1
649 if alleB(anzR_alle-i,5)==0 %nicht unterstützende Ri löschen
        alleR2(anzR_alle-i,:)=[];
        o_alle(anzR_alle-i,:)=[]; %dementsprechende Orientierungen
        t_alle(anzR_alle-i,:)=[]; %dementsprechende Richtungswinkel
    end
654 end

o_support=o_alle;
t_support=t_alle;

659 for i=1:length(support)
    if support(i,1)>anzR_alle %wenn Beob. = Strecke, also Zeilennummer größer als die Anzahl der verw. Ri
        sGK_imp_sup(i,1:3)=sGK_imp(support(i,1)-anzR_alle,1:3); %verwendete Strecken suchen und entsprechende
        Spalten in neuer Matrix speichern
    end
664 end

sGK_imp_sup(~any(sGK_imp_sup,2),:)=[];

%% Testen ob kompletter Richtungssatz fehlt bzw. Satznummern fortlaufend sind
standpunkte_support =unique(alleR2(:,1));
669 gstdpk_support=size(standpunkte_support);

%Matrix erstellen, die die noch vorhandenen Standpunkte und ihre Anzahl von Sätzen zeigt
for i=1:gstdpk_support(1,1)
674 [row_sup]=find(alleR2(:,1)==standpunkte_support(i,1));
        bet_sup=alleR2(row_sup(1,1):row_sup(length(row_sup),1),7);
        anzahl_sup=unique(bet_sup);
        anzahl2_sup=anzahl_sup(length(anzahl_sup),1);
        saetze_sup(i,1)=standpunkte_support(i,1);
679 saetze_sup(i,2)=anzahl2_sup;

        zahlen=unique(alleR2(row_sup(1,1):row_sup(length(row_sup),1),7)); % zugehörige Satznummern
        gza=size(zahlen);

        if gza(1,1)<2 %wenn nur mehr eine Beobachtung in einem Satz übrig ist
684 li=zahlen(1,1)-zahlen(1,1);
            re=1;
        elseif gza(1,1)==2 && zahlen(1,1)*2 == zahlen(2,1) %wenn Satznr 1 = 2*Satznr 2
            li=0;
            re=1;
689 elseif gza(1,1)==3 && zahlen(1,1)*2 == zahlen(2,1) && zahlen(2,1)==zahlen(3,1)*2 %wenn Satznr 1 = 2*Satznr 2 und
            Satznr 2 = 2*Satznr 3
            li=0;
            re=1;
        else
            for j=0:(gza(1,1)-2) %wenn noch mind. 2 Sätze übrig sind
694 li(j+1,1)=zahlen(gza(1,1)-j,1)-zahlen(1,1);
                re(j+1,1)=zahlen(gza(1,1)-(j+1),1);
            end
        end
699 test=isequal(li,re); %testet, ob die beiden Vektoren gleich sind, also noch alle Sätze vorhanden sind

if test==0
    saetze_sup(i,2)=gza(1,1); %Anzahl der Sätze in Matrix saetze richtigstellen
    for k=1:gza(1,1)
704 [such]=find(alleR2(row_sup(1,1):row_sup(length(row_sup),1),7)==zahlen(k,1)); %Zeilen finden, in denen
        die falschen Satzbezeichnungen stehen
    end
end

```

```

        offset_sup=row_sup(1,1)-1;
        such=such+offset_sup;
        alleR2(such(1,1):such(length(such),1),7)=k;      %falsche durch richtige fortlaufende Bezeichnung
                ersetzen
709         end
        end
end

714
m=1;          %Näherung für Orientierung berechnen
for i=1:length(saetze_sup)
    for j=1:saetze_sup(i,2)
        [row]=find(alleR2(:,1)==saetze_sup(i,1));
719        [s]=find(alleR2(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
        mittelo_support=mean(o_support(s(1,1):s(length(s)),1));
        X0_o_support(m,1)=mittelo_support;
724        m=m+1;
        for k=s(1,1):s(length(s),1)
            o_support(k,2)=mittelo_support;
        end
        end
729    end

X0_support=[Koord;X0_o_support];
L0_s_support=L0_support(anzR_support+1:length(L0_alle),1);
L0_r_support=t_support(:,3)-o_support(:,2);
734

else %alle weiteren Iterationen: Xd als Näherungswert
    %% L0 aufstellen
    %Richtungen

739    %Näherungen für Orientierungen berechnen
    %Richtungswinkel berechnen
    Such_y_k=[alleR2(:,1:2);sGK_imp_sup(:,1:2)];
    Such_x_k=[alleR2(:,1:2);sGK_imp_sup(:,1:2)];

744    %y-Koordinaten er- bzw. einsetzen
    for j=1:length(Index)
        ii=Such_y_k==Index(j,1);
        Such_y_k(ii)=koord(j,2);
    end
749
    %x-Koordinaten er- bzw. einsetzen
    for j=1:length(Index)
        ii=Such_x_k==Index(j,1);
        Such_x_k(ii)=koord(j,3);
754    end

naeherung_r=[Such_y_k(1:anzR_support,1) Such_x_k(1:anzR_support,1) Such_y_k(1:anzR_support,2) Such_x_k(1:anzR_support
,2)]; %Struktur: [y-Koord.Standpkt. x-Koord.Standpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

for i=1:anzR_support
759    t_support(i,1)=naeherung_r(i,3)-naeherung_r(i,1); %Zähler für Richtungswinkel tSZ
    t_support(i,2)=naeherung_r(i,4)-naeherung_r(i,2); %Nenner für Richtungswinkel tSZ
end

for i=1:length(t_support)
764    %Berechnung der Richtungswinkel mit
    %Quadrantenzuordnung
    if t_support(i,1)> 0 && t_support(i,2)< 0
        t_support(i,3)=(atan(t_support(i,1)/t_support(i,2)))*(200/pi)+200;
    end
    if t_support(i,1)<0 && t_support(i,2)<0
        t_support(i,3)=(atan(t_support(i,1)/t_support(i,2)))*(200/pi)+200;
769    end
    if t_support(i,1)<0 && t_support(i,2)>0
        t_support(i,3)=(atan(t_support(i,1)/t_support(i,2)))*(200/pi)+400;
    end
    if t_support(i,1)>0 && t_support(i,2)>0
774    t_support(i,3)=(atan(t_support(i,1)/t_support(i,2)))*(200/pi);
    end
end

o_support=o_probe;
779    for i=1:length(o_support)
        if o_support(i,1)<0
            o_support(i,1)=o_support(i,1)+400;
        end
    end
end

```

```

784   for i=1:length(o_support)           %für korrekte Berechnung des Mittelwerts
      if o_support(i,1)<0.9995
        o_support(i,1)=o_support(i,1)+400;
      end
    end
789
    %Strecken
    naeherung_s=[Such_y_k(anzR_support+1:length(support),1) Such_x_k(anzR_support+1:length(support),1) Such_y_k(
      anzR_support+1:length(support),2) Such_x_k(anzR_support+1:length(support),2)]; %Struktur: [y-Koord.Stdpkt. x-
      Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

794   for i=1:length(naeherung_s)
      L0_s_support(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
    end

    %%Aufstellen der A-Matrix
    %y-Koordinaten er- bzw. einsetzen für Ri & Str.
799   Such_y_a=[alleR2(:,1:2);sGK_imp_sup(:,1:2)];
    Such_x_a=[alleR2(:,1:2);sGK_imp_sup(:,1:2)];

    for j=1:length(Index)
      ii=Such_y_a==Index(j,1);
804     Such_y_a(ii)=koord(j,2);
    end

    %x-Koordinaten er- bzw. einsetzen für Ri & Str.
809   for j=1:length(Index)
      ii=Such_x_a==Index(j,1);
      Such_x_a(ii)=koord(j,3);
    end

    naeherung_r_sup=[Such_y_a(1:anzR_support,1) Such_x_a(1:anzR_support,1) Such_y_a(1:anzR_support,2) Such_x_a(1:
814     anzR_support,2)]; %Struktur: [y-Koord.Standpkt. x-Koord.Standpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

    %Ableitungen nach Richtungen berechnen
    for i=1:anzR_support
      ableitungen_r_sup(i,1)=(-(naeherung_r_sup(i,4)-naeherung_r_sup(i,2))/((naeherung_r_sup(i,4)-naeherung_r_sup(i,2))
        ^2 + (naeherung_r_sup(i,3)-naeherung_r_sup(i,1))^2))*(200/pi); %dR/dyS
      ableitungen_r_sup(i,2)=(naeherung_r_sup(i,3)-naeherung_r_sup(i,1))/((naeherung_r_sup(i,4)-naeherung_r_sup(i,2))^2
        + (naeherung_r_sup(i,3)-naeherung_r_sup(i,1))^2))*(200/pi); %dR/dxS
819     ableitungen_r_sup(i,3)=(naeherung_r_sup(i,4)-naeherung_r_sup(i,2))/((naeherung_r_sup(i,4)-naeherung_r_sup(i,2))^2
        + (naeherung_r_sup(i,3)-naeherung_r_sup(i,1))^2))*(200/pi); %dR/dyZ
      ableitungen_r_sup(i,4)=(-(naeherung_r_sup(i,3)-naeherung_r_sup(i,1))/((naeherung_r_sup(i,4)-naeherung_r_sup(i,2))
        ^2 + (naeherung_r_sup(i,3)-naeherung_r_sup(i,1))^2))*(200/pi); %dR/dxZ
    end

    naeherung_s_sup=[Such_y_a(length(alleR2)+1:length(support),1) Such_x_a(length(alleR2)+1:length(support),1) Such_y_a(
824     length(alleR2)+1:length(support),2) Such_x_a(length(alleR2)+1:length(support),2)]; %Struktur: [y-Koord.Stdpkt.
      x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

    %Ableitungen nach Strecken berechnen
    for i=1:anzS_support
      ableitungen_s_sup(i,1)=(naeherung_s_sup(i,1)-naeherung_s_sup(i,3))/sqrt(((naeherung_s_sup(i,4)-naeherung_s_sup(i
        ,2))^2 + (naeherung_s_sup(i,3)-naeherung_s_sup(i,1))^2)); %dS/dyS
      ableitungen_s_sup(i,2)=(naeherung_s_sup(i,2)-naeherung_s_sup(i,4))/sqrt(((naeherung_s_sup(i,4)-naeherung_s_sup(i
        ,2))^2 + (naeherung_s_sup(i,3)-naeherung_s_sup(i,1))^2)); %dS/dxS
829     ableitungen_s_sup(i,3)=(naeherung_s_sup(i,3)-naeherung_s_sup(i,1))/sqrt(((naeherung_s_sup(i,4)-naeherung_s_sup(i
        ,2))^2 + (naeherung_s_sup(i,3)-naeherung_s_sup(i,1))^2)); %dS/dyZ
      ableitungen_s_sup(i,4)=(naeherung_s_sup(i,4)-naeherung_s_sup(i,2))/sqrt(((naeherung_s_sup(i,4)-naeherung_s_sup(i
        ,2))^2 + (naeherung_s_sup(i,3)-naeherung_s_sup(i,1))^2)); %dS/dxZ
    end

    %A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
834   A_support=zeros(length(L_support),length(Koord));

    % Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Standpunkt gehören
    for i=1:anzR_support
      for j=1:length(Index)
839         if alleR2(i,1)==Index(j,1)
            A_support(i,Index(j,2))=ableitungen_r_sup(i,1); %Ableitung nach der y-Koordinate des Standpunktes
            einfüllen
            A_support(i,(Index(j,2)+1))=ableitungen_r_sup(i,2); %Ableitung nach der x-Koordinate des Standpunktes
            einfüllen
          end
        end
844     end

    %Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt gehören
    for i=1:anzR_support
      for j=1:length(Index)
849         if alleR2(i,2)==Index(j,1)

```

```

                A_support(i, Index(j,2))=ableitungen_r_sup(i,3);      %Ableitung nach der y-Koordinate des Zielpunktes
                einsetzen
                A_support(i, (Index(j,2)+1))=ableitungen_r_sup(i,4);  %Ableitung nach der x-Koordinate des Zielpunktes
                einsetzen
            end
        end
854    end

    % Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
    for i=1:anzS_support
        for j=1:length(Index)
859            if sGK_imp_sup(i,1)==Index(j,1)
                A_support(anzR_support+i, Index(j,2))=ableitungen_s_sup(i,1);
                A_support(anzR_support+i, (Index(j,2)+1))=ableitungen_s_sup(i,2);
            end
        end
864    end

    %Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
    for i=1:anzS_support
        for j=1:length(Index)
869            if sGK_imp_sup(i,2)==Index(j,1)
                A_support(anzR_support+i, Index(j,2))=ableitungen_s_sup(i,3);
                A_support(anzR_support+i, (Index(j,2)+1))=ableitungen_s_sup(i,4);
            end
        end
874    end

    %eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
    saetze_sup(1,3)=0;
    for i=1:length(saetze_sup)-1
879        saetze_sup(i+1,3)=saetze_sup(i,3)+saetze_sup(i,2);    %Spaltenpos. für die Ableitung nach der O.unbek. berechnen
    end

    A0_support=zeros(anzR_support, length(X0_o_support));
    for i=1:length(saetze_sup)
884        n=saetze_sup(i,3);
        for j=1:saetze_sup(i,2)
            [row]=find(alleR2(:,1)==saetze_sup(i,1));
            [s]=find(alleR2(row(1,1):row(length(row),1),7)==j);
            offset=row(1,1)-1;
889            s=s+offset;
            n=n+j;
            for k=s(1,1):s(length(s),1)
                A0_support(k,n)=-1;
            end
894            n=n-j;
        end
    end

    %A0 um Nullbereich für die Streckenbeobachtungen erweitern
899    A0_support=[A0_support; zeros(length(sGK_imp_sup), length(X0_o_support))];
    A_support=[A_support A0_support];

    %X0 aus vorheriger Iteration übernehmen
904    X0_support=Xd_support;

    %Berechnung von L0_support für Richtungen mit den O.unbek. aus letzter Iteration
    L0_r_support=t_support(:,3)-o_probe(:,1);

909    end

    for i=1:length(L0_r_support)
        if L0_r_support(i,1)<0
            L0_r_support(i,1)=L0_r_support(i,1)+400;
        else
914            L0_r_support(i,1)=L0_r_support(i,1);
        end
    end

    L0_support=[L0_r_support; L0_s_support];
919    L_support=L_support-L0_support;

    S11_support=eye(length(L_support), length(L_support));    %P-Matrix aufstellen
    for i=1:length(L_support)
        S11_support(i,i)=sigma2_support(i);
924    end
    sigma02=1;
    Q11_support=(1/sigma02)*S11_support;
    P_support=Q11_support^-1;
929

```



```

%Ränderungsmatrix G aufstellen
G1_support=repmat([0 1],1,length(Koord)/2);
G2_support=repmat([1 0],1,length(Koord)/2);
null=zeros(1,length(X0_o_support));
934 G_support=[G1_support null;G2_support null];

for i=1:2:length(Koord)
    G_support(3,i+1)=Koord(i,1);
end
939 for i=2:2:length(Koord)
    G_support(3,i-1)=-Koord(i,1);
end

%Ausgleich
944 N_support=[A_support'*P_support*A_support G_support';G_support null_N];
xk_support=N_support^-1*[A_support'*P_support*I_support;null_xk];
x_support=xk_support(1:length(X0_support),1);

v_support=A_support*x_support-I_support;
949 Ld_support=L_support+v_support;
Xd_support=X0_support+x_support;

s02=(v_support'*P_support*v_support)/(beob-unk+d);

954 %% Hauptprobe
Xd_koord=Xd_support(1:length(G1_support),1);
Xd_hilfe_r=reshape(Xd_koord,2,length(G1_support)/2);
Xd_hilfe_r=Xd_hilfe_r';
959 Xd_hilfe_r=[Index(1:length(G1_support)/2,1) Xd_hilfe_r];

%ausgeglichene Orientierungen den Standpunkten zuweisen
Xd_o=Xd_support(length(G1_support)+1:length(Xd_support),1);

m=1;
964 for i=1:length(saetze_sup)
    for j=1:saetze_sup(i,2)
        [row]=find(alleR2(:,1)==saetze_sup(i,1));
        [s]=find(alleR2(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
969 s=s+offset;
        for k=s(1,1):s(length(s),1)
            o_probe(k,1)=Xd_o(m,1);
            end
            m=m+1;
974 end
        end
    end

Xd_probe_r_y=alleR2(:,1:2);
Xd_probe_r_x=alleR2(:,1:2);
979

%y-Koordinaten er- bzw. einsetzen
for i=1:length(alleR2(:,1))
    for j=1:length(Xd_hilfe_r)
        jj=Xd_probe_r_y==Xd_hilfe_r(j,1);
984 Xd_probe_r_y(jj)=Xd_hilfe_r(j,2);
    end
end

%x-Koordinaten er- bzw. einsetzen
989 for i=1:length(alleR2(:,1))
    for j=1:length(Xd_hilfe_r)
        jj=Xd_probe_r_x==Xd_hilfe_r(j,1);
        Xd_probe_r_x(jj)=Xd_hilfe_r(j,3);
    end
994 end

%Zuordnung der ausgeglichenen Koordinaten zu den ausgeglichenen Orientierungen
Ld_probe_r=[Xd_probe_r_y(:,1) Xd_probe_r_x(:,1) Xd_probe_r_y(:,2) Xd_probe_r_x(:,2) o_probe(:,1)]; %Schema:[yS xS yZ xZ o]

999 %Berechnung der Richtungen aus den ausgeglichenen Koord & Orientierungen
%Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:anzR_support
    Ld_HP_r(i,1)=Ld_probe_r(i,3)-Ld_probe_r(i,1); %Zähler der Richtungswinkel
    Ld_HP_r(i,2)=Ld_probe_r(i,4)-Ld_probe_r(i,2); %Nenner der Richtungswinkel
1004 end

for i=1:length(Ld_HP_r)
    if Ld_HP_r(i,1)> 0 && Ld_HP_r(i,2)< 0
        Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;
1009 end
    if Ld_HP_r(i,1)< 0 && Ld_HP_r(i,2)< 0
        Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+200;

```

```

end
1014 if Ld_HP_r(i,1)<0 && Ld_HP_r(i,2)>0
      Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi)+400;
end
if Ld_HP_r(i,1)>0 && Ld_HP_r(i,2)>0
      Ld_HP_r(i,3)=(atan(Ld_HP_r(i,1)/Ld_HP_r(i,2)))*(200/pi);
end
1019 end

for i=1:anzR_support
      Ld_HP_r(i,4)=Ld_HP_r(i,3)-o_probe(i,1);
end
1024

for i=1:length(Ld_HP_r)
      if Ld_HP_r(i,4)<0
            Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;
      end
end
1029

for i=1:length(Ld_HP_r)      %wenn Richtung um Ogon ist, 400gon für richtigen Vergleich mit Ld addieren
      if Ld_HP_r(i,4)<0.9995
            Ld_HP_r(i,4)=Ld_HP_r(i,4)+400;
      end
end
1034

%%Strecken für HP vorbereiten
Xd_strecken=Ld_support(anzR_support+1:length(Ld_support),1);
1039

Xd_probe_s_y=sgk_imp_sup(:,1:2);
Xd_probe_s_x=sgk_imp_sup(:,1:2);

%ausgeglichene y-Koordinaten einsetzen
1044 for i=1:anzS_support
      for j=1:length(Index)
            ll=Xd_probe_s_y==Xd_hilfe_r(j,1);
            Xd_probe_s_y(ll)=Xd_hilfe_r(j,2);
      end
end
1049

%ausgeglichene x-Koordinaten einsetzen
for i=1:anzS_support
      for j=1:length(Index)
1054          mm=Xd_probe_s_x==Xd_hilfe_r(j,1);
            Xd_probe_s_x(mm)=Xd_hilfe_r(j,3);
      end
end

1059 Ld_probe_s=[Xd_probe_s_y(:,1) Xd_probe_s_x(:,1) Xd_probe_s_y(:,2) Xd_probe_s_x(:,2) Xd_strecken];

for i=1:length(Xd_strecken)
      Ld_HP_s(i,1)=sqrt((Ld_probe_s(i,3)-Ld_probe_s(i,1))^2 + (Ld_probe_s(i,4)-Ld_probe_s(i,2))^2);
end
1064

for i=1:anzR_support
      if Ld_support(i,1)<0
            Ld_support(i,1)=Ld_support(i,1)+400;
      else
1069          Ld_support(i,1)=Ld_support(i,1);
      end
end

for i=1:anzR_support      %Anpassungen für dieses Netz
1074 if Ld_support(i,1)<0.9995 && Ld_HP_r(i,4)>390.9928
            Ld_p(i,1)=Ld(i,1)+2*pi;
      elseif Ld_support(i,1)>390.9928 && Ld_HP_r(i,4)<0.9995
            Ld_p(i,1)=abs(Ld_support(i,1)-2*pi);
      else
1079          Ld_p(i,1)=Ld_support(i,1);
      end
end

HP=[Ld_p;Ld_support(anzR_support+1:length(Ld_support),1)]-[Ld_HP_r(:,4);Ld_HP_s];
1084

HPabs=abs(HP);
maximum=max(HPabs);
maximum_r=max(HPabs(1:anzR_support,1));
maximum_s=max(HPabs(anzR_support+1:length(HP),1));
1089

koord=Xd_hilfe_r;
Koord=Xd_support(1:length(G1_support));

```

```

1094     if iteration_l==1
           Xd_sammel(:,1)=Xd_support;
        else
           Xd_sammel(:,iteration_l)=Xd_support; %letzte Spalte in Xd_sammel enthält die finalen ausgeglichenen Unbekannten
           diff_Xd=abs(Xd_sammel(:,iteration_l-1)-Xd_support);
1099     max_diff_k=max(diff_Xd(1:length(G1_support),1));
           max_diff_o=max(diff_Xd(length(G1_support):length(Xd_support),1));
        end
    end
1104 %% Verbesserungen zur endgültigen Lösung berechnen (geschieht anhand derselben Funktion wie oben, nur mit anderen
           Variablenbezeichnungen)
    [iteration fehler p ] = verbesserungen_final (Xd_support, SGK_imp, G1, anzR_alle, saetze, alleR, Index, alleB, alle, iteration
    );
    [zeile]=find(fehler==1) %grobe Fehler (v > 3sigma) am Bildschirm ausgeben
    toc %Ende der Zeitmessung

```

## F.2 Funktion für die Verbesserungen

```

function [iteration wahr_sample p w ] = verbesserungen5 (Xd_sample, SGK_imp, G1, anzR_alle, saetze, alleR, Index, alleB, alle,
    iteration)
%Verbesserungen aller Beobachtungen zur aktuellen Lösung berechnen
3
global wahr_sample p v_alle schwelle r_genauigkeit
lS=length(SGK_imp); %Anzahl aller ursprünglichen Strecken
lR=anzR_alle; %Anzahl aller ursprünglichen Richtungen

8 %% Strecken
[uniqueS]=unique(SGK_imp(:,1:2),'rows'); %findet alle unterschiedlichen Visuren

Xd_koord=Xd_sample(1:length(G1),1);
Xd_hilfe=reshape(Xd_koord,2,length(G1)/2);
13 Xd_hilfe=Xd_hilfe';
Xd_hilfe=[Index(1:length(G1)/2,1) Xd_hilfe];

Xd_s_y=uniqueS;
Xd_s_x=uniqueS;
18

%ausgegliche y-Koordinaten einsetzen
for i=1:length(uniqueS)
    for j=1:length(Index)
        ll=Xd_s_y==Xd_hilfe(j,1);
23 Xd_s_y(ll)=Xd_hilfe(j,2);
    end
end

%ausgegliche x-Koordinaten einsetzen
28 for i=1:length(uniqueS)
    for j=1:length(Index)
        mm=Xd_s_x==Xd_hilfe(j,1);
        Xd_s_x(mm)=Xd_hilfe(j,3);
    end
33 end

%% Strecken
yx_s=[Xd_s_y(:,1) Xd_s_x(:,1) Xd_s_y(:,2) Xd_s_x(:,2)]; %Schema: [yStdpkt xStdpkt yZielpkt yZielpkt]

38 for i=1:length(uniqueS)
    S(i,1)=sqrt((yx_s(i,3)-yx_s(i,1))^2 + (yx_s(i,4)-yx_s(i,2))^2); %Strecken aus den ausgeglichenen Koordinaten berechnen
end

for i=1:length(uniqueS)
43 [r]=find(SGK_imp(:,1)==uniqueS(i,1) & SGK_imp(:,2)==uniqueS(i,2)); %ursprünglichen Visuren suchen
    Ld_s(r(1,1):r(length(r),1),3)=S(i,1); %und die ausgeglichenen Strecken in die entsprechenden Zeilen schreiben
end

Ld_s(:,1:2)=SGK_imp(:,1:2);
48 v_s=Ld_s(:,3)-SGK_imp(:,3); %Verbesserungen für alle Strecken

%% Richtungen
Xd_o=Xd_sample(length(G1)+1:length(Xd_sample),1); %ausgegliche Orientierungen extrahieren

53 m=1;
for i=1:length(saetze)
    for j=1:saetze(i,2)
        [row]=find(alleR(:,1)==saetze(i,1));
        [s]=find(alleR(row(1,1):row(length(row),1),7)==j);
58 offset=row(1,1)-1;
        s=s+offset;
    end
end

```

```

        for k=s(1,1):s(length(s),1)
            o_alle(k,1)=Xd_o(m,1);
        end
63     m=m+1;
    end
end

Xd_r_y=alleR(:,1:2);
68 Xd_r_x=alleR(:,1:2);

%y-Koordinaten er- bzw. einsetzen
for i=1:length(alleR(:,1))
    for j=1:length(Xd_hilfe)
73         jj=Xd_r_y=Xd_hilfe(j,1);
            Xd_r_y(jj)=Xd_hilfe(j,2);
        end
    end
end

78 %x-Koordinaten er- bzw. einsetzen
for i=1:length(alleR(:,1))
    for j=1:length(Xd_hilfe)
        jj=Xd_r_x=Xd_hilfe(j,1);
83         Xd_r_x(jj)=Xd_hilfe(j,3);
    end
end

%Zuordnung der ausgeglichenen Koordinaten zu den ausgeglichenen Orientierungen
yxr=[Xd_r_y(:,1) Xd_r_x(:,1) Xd_r_y(:,2) Xd_r_x(:,2) o_alle(:,1)]; %Schema: [yS xS yZ xZ o]
88

%Berechnung der Richtungen aus den ausgeglichenen Koord & Orientierungen
%Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:1R
    Ld_r(i,1)=yx_r(i,3)-yx_r(i,1); %Zähler der Richtungswinkel
93     Ld_r(i,2)=yx_r(i,4)-yx_r(i,2); %Nenner der Richtungswinkel
end

for i=1:length(Ld_r)
    if Ld_r(i,1)> 0 && Ld_r(i,2)< 0
98         Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+200;
    end
    if Ld_r(i,1)<0 && Ld_r(i,2)<0
        Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+200;
    end
103    if Ld_r(i,1)<0 && Ld_r(i,2)>0
        Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+400;
    end
    if Ld_r(i,1)>0 && Ld_r(i,2)>0
108         Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi);
    end
end

for i=1:1R
113     Ld_r(i,4)=Ld_r(i,3)-o_alle(i,1);
end

for i=1:length(Ld_r)
    if Ld_r(i,4)<0
118         Ld_r(i,4)=Ld_r(i,4)+400;
    end
end

v_r=alleR(:,6)-Ld_r(:,4);

123 %% Schwellenwerte und Wahrscheinlichkeit bestimmen
schwelle_r=sqrt(r_genauigkeit);
schwelle_s=5/1000*ones(length(sGK_imp),1);
schwelle=[schwelle_r;schwelle_s];
schwelle(:,2)=schwelle(:,1)*3;
128 v_alle(:,iteration)=[v_r;v_s];

for i=1:length(alleB)
    if abs(v_alle(i,iteration))<schwelle(i,2) %wenn die Verbesserung unter dem Schwellenwert liegt, wird die Anzahl der
        unterstützenden Beob. um 1 erhöht
        wahr(i,1)=1;
133     else
        wahr(i,1)=0; %wenn die Verbesserung die Lösung nicht unterstützt
    end
end

138 wahr_sample(1:length(alleB),iteration)=wahr;
p(iteration,1)=sum(wahr)/length(alle); %Bestimmung des Quotienten
end

```

## G LTS

Bei der Umsetzung von LTS werden einige Rechenschritte mehrmals wiederholt und deshalb in eigene Funktionen ausgelagert; die nachfolgenden Abschnitte stellen diese vor.

### G.1 Hauptprogramm

```
clc           %Hauptfenster leeren
clear all    %alle Variablen löschen
tic         %Start für Zeitmessung der Rechendauer

5  global Xd_sample anzR_alle v_h3 Q Q1 Q2 Q3 %globale Variablen (für ausgelagerte Funktionen)

%alle Beobachtungen einlesen
sGK_imp=textread('sGK_acorr.txt');
richtungen=textread('richtungen_acorr.txt. ');
10 koord=textread('naeherung_a.txt');
%str=textread('strecken_a.txt');

richtungen_1=richtungen; %Hilfsvariable für Auswahlverfahren
richtungen_1(:,8)=1:length(richtungen);
15 richtungen_0=richtungen;
richtungen_0(:,8)=1:length(richtungen);
alle=[richtungen(:,1:2) richtungen(:,7);sGK_imp(:,1:2) sGK_imp(:,4)];
alle(:,3)=1:length(alle);
alleR=richtungen; %alle Richtungen im Original speichern, für Berechnung der Verbesserungen
20 alleB=alle; %alle Beobachtungen im Original speichern, zum Reseten für neue Iteration

%Definitionen für die Zuordnung in der A-Matrix
verwendete_pkte=unique(alle(:,1:2));
index=(1:2:2*length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Koordinaten verwendet
25 ind=(1:1:length(verwendete_pkte))'; %für Positionierung der Ableitung nach den Orientierungsunbekannten verwendet
Index=[verwendete_pkte index ind];

%Matrix mit Stand- u. Zielpunktsnr.
Such_y=alle(:,1:2);
30 Such_x=alle(:,1:2);

%Vektorgrößen definieren
gr=size(richtungen);
anzR_alle=gr(1,1); %Anzahl aller Richtungsmessungen, für endgültige Lösung benötigt
35

%Matrix erstellen, die die Standpunkte und ihre Anzahl von Sätzen zeigt
for i=1:gstdpk_r(1,1)
40 [row]=find(richtungen(:,1)==standpunkte_r(i,1));
bet=richtungen(row(1,1):row(length(row),1),7);
anzahl=unique(bet);
anzahl2=anzahl(length(anzahl),1);
saetze(i,1)=standpunkte_r(i,1);
45 saetze(i,2)=anzahl2;
end

%% L aufstellen
standpunkte_r =unique(richtungen(:,1));
50 gstdpk_r=size(standpunkte_r);

%Matrix erstellen, die die Standpunkte der RICHTUNGEN und ihre Anzahl von Sätzen zeigt
for i=1:gstdpk_r(1,1)
[ row]=find(richtungen(:,1)==standpunkte_r(i,1));
55 bet_r=richtungen(row(1,1):row(length(row),1),7);
anzahl_r=unique(bet_r);
anzahl2_r=anzahl_r(length(anzahl_r),1);
saetze_r(i,1)=standpunkte_r(i,1);
saetze_r(i,2)=anzahl2_r;
60 end

%Trimmingkonstante h und Bruchpunkt definieren
unbek=length(Index(:,1))*2+sum(saetze_r(:,2));
beob=length(alle(:,1));
65 h=round((beob+unbek+1)/2);
breakdown=(beob-h+1)/beob;

standpunkte_s =unique(sGK_imp(:,1));
gstdpk_s=size(standpunkte_s);
70
```

```

%Matrix erstellen, die die Standpunkte der STRECKEN und ihre Anzahl von Sätzen zeigt
for i=1:gstdpk_s(1,1)
    [row]=find(sGK_imp(:,1)==standpunkte_s(i,1));
    bet_s=sGK_imp(row(1,1):row(length(row),1),4);
75     anzahl_s=unique(bet_s);
    anzahl2_s=anzahl_s(length(anzahl_s),1);
    saetze_s(i,1)=standpunkte_s(i,1);
    saetze_s(i,2)=anzahl2_s;
end
80
wahl=unbek-sum(saetze_s(:,2))-sum(saetze_r(:,2)); %Anzahl der benötigten Beobachtungen für eine eindeutig bestimmte Lösung

L_r=richtungen(:,6);
L=[L_r;sGK_imp(:,3)];
85 L_alle=L; %für Berechnung der endgültigen Lösung
L_1=L; %Hilfsvariable für Auswahlverfahren
L_1(:,2)=1:length(L_1); %Nummerierung der Messungen in 2.Spalte
L_0=L;
L_0(:,2)=1:length(L_0); %Nummerierung der Messungen in 2.Spalte
90

%% L0 aufstellen
%Näherungen für Orientierungen berechnen
%Richtungswinkel berechnen
Such_y_k=Such_y;
95 Such_x_k=Such_x;

%y-Koordinaten er- bzw. einsetzen
for j=1:length(Index)
    ii=Such_y==Index(j,1);
100     Such_y_k(ii)=koord(j,2);
end

%x-Koordinaten er- bzw. einsetzen
for j=1:length(Index)
    ii=Such_x==Index(j,1);
105     Such_x_k(ii)=koord(j,3);
end

naeherung_r=[Such_y_k(1:gr(1,1),1) Such_x_k(1:gr(1,1),1) Such_y_k(1:gr(1,1),2) Such_x_k(1:gr(1,1),2)]; %Struktur: [y-Koord.
110     Standpkt. x-Koord.Standpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

for i=1:gr(1,1)
    t(i,1)=naeherung_r(i,3)-naeherung_r(i,1); %Zähler für Richtungswinkel tSZ
    t(i,2)=naeherung_r(i,4)-naeherung_r(i,2); %Nenner für Richtungswinkel tSZ
end
115

%Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:length(t)
    if t(i,1)> 0 && t(i,2)< 0
120         t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
    end
    if t(i,1)<0 && t(i,2)<0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+200;
    end
    if t(i,1)<0 && t(i,2)>0
125         t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi)+400;
    end
    if t(i,1)>0 && t(i,2)>0
        t(i,3)=(atan(t(i,1)/t(i,2)))*(200/pi);
    end
130 end

t_alle=t; %für Berechnung der endgültigen Lösung
t_1=t; %Hilfsvariable für Eliminieren
t_1(:,4)=1:length(t); %Nummerierung der Messungen in 2.Spalte
135 t_0=t;
t_0(:,4)=1:length(t); %Nummerierung der Messungen in 2.Spalte

for i=1:length(t) %Berechnung der Einzelorientierungen
140     o(i,1)=t(i,3)-L_r(i);
end

for i=1:length(o)
    if o(i,1)<0
        o(i,1)=o(i,1)+400;
145     end
end

for i=1:length(o) %für korrekte Berechnung des Mittelwerts
    if o(i,1)<0.9995
        o(i,1)=o(i,1)+400;
150     end
end
end

```

```

o_alle=o;           %für Berechnung der endgültigen Lösung
o_1=o;             %Hilfsvariable für Auswahlverfahren
o_1(:,3)=1:length(o); %Numerierung der Messungen in 2.Spalte
155 o_0=o;
o_0(:,3)=1:length(o); %Numerierung der Messungen in 2.Spalte

m=1;
for i=1:length(saetze_r)
160   for j=1:saetze_r(i,2)
       [row]=find(richtungen(:,1)==saetze_r(i,1));
       [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
       offset=row(1,1)-1;
       s=s+offset;
165       mittelo=mean(o(s(1,1):s(length(s)),1));
       X0_o(m,1)=mittelo;
       m=m+1;
       for k=s(1,1):s(length(s),1)
           o(k,2)=mittelo;
170       end
       end
end

%Berechnung von L0
175 % RICHTUNGEN
L0_r=t(:,3)-o(:,2);

for i=1:length(L0_r)
    if L0_r(i,1)<0
180       L0_r(i,1)=L0_r(i,1)+400;
    else
        L0_r(i,1)=L0_r(i,1);
    end
end
185

% STRECKEN
naeherung_s=[Such_y_k(gr(1,1)+1:length(alle),1) Such_x_k(gr(1,1)+1:length(alle),1) Such_y_k(gr(1,1)+1:length(alle),2) Such_x_k
    (gr(1,1)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt. x-Koord.Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]

for i=1:length(naeherung_s)
190   L0_s(i,1)=sqrt((naeherung_s(i,3)-naeherung_s(i,1))^2 + (naeherung_s(i,4)-naeherung_s(i,2))^2);
end

L0=[L0_r;L0_s];
L0_alle=L0;           %für Berechnung der endgültigen Lösung
195 L0_1=L0;           %Hilfsvariable für Auswahlverfahren
L0_1(:,2)=1:length(L0); %Numerierung der Messungen in 2.Spalte
L0_0=L0;
L0_0(:,2)=1:length(L0); %Numerierung der Messungen in 2.Spalte

200 %% Genauigkeit (P-Matrix)
% verwendete Geräte mit unterschiedlicher Genauigkeit beachten
% s_genauigkeit=[sGK_imp(:,3) str(:,14:15)];
% for i=1:length(s_genauigkeit)
%     s_genauigkeit(i,4)=(s_genauigkeit(i,2))^2+((s_genauigkeit(i,1)/1000)*s_genauigkeit(i,3))^2*10^-6;
205 % end
% r_genauigkeit=richtungen(:,5);
% for i=1:length(r_genauigkeit)
%     r_genauigkeit(i,2)=(r_genauigkeit(i,1)/1000)*(pi/200))^2;
% end

210 for i=1:length(sGK_imp)
    s_genauigkeit(i,1)=(5)^2+((sGK_imp(i,3)/1000)*2)^2*10^-6;
end
for i=1:length(richtungen)
215   r_genauigkeit(i,1)=(0.5/1000)^2;
end

sigma2=[r_genauigkeit(:,1);s_genauigkeit(:,1)];
sigma2_alle=sigma2;           %für Berechnung der endgültigen Lösung
220 sigma2_1=sigma2;           %Hilfsvariable für Auswahlverfahren
sigma2_1(:,2)=1:length(sigma2); %Numerierung der Messungen in 2.Spalte
sigma2_0=sigma2;
sigma2_0(:,2)=1:length(sigma2); %Numerierung der Messungen in 2.Spalte

225 %% Unbekanntenvektor X0 erstellen
koord_t=koord(:,2:3)';
Koord=reshape(koord_t, 2*length(koord),1);

%% Ableitungen für die A-Matrix berechnen
230 %y-Koordinaten er- bzw. einsetzen für Ri & Str.
Such_y_a=Such_y;
Such_x_a=Such_x;

```

```

for j=1:length(Index)
  ii=Such_y==Index(j,1);
235  Such_y_a(ii)=koord(j,2);
end

%x-Koordinaten er- bzw. einsetzen für Ri & Str.
for j=1:length(Index)
240  ii=Such_x==Index(j,1);
  Such_x_a(ii)=koord(j,3);
end

naeherung_r=[Such_y_a(1:gr(1,1),1) Such_x_a(1:gr(1,1),1) Such_y_a(1:gr(1,1),2) Such_x_a(1:gr(1,1),2)]; %Struktur: [y-Koord.
  Standpkt. x-Koord.Standpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]
245

%Ableitungen nach Richtungen berechnen
for i=1:gr(1,1)
  ableitungen_r(i,1)=(-(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dyS
  ableitungen_r(i,2)=(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dxS
250  ableitungen_r(i,3)=(naeherung_r(i,4)-naeherung_r(i,2))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dyZ
  ableitungen_r(i,4)=(-(naeherung_r(i,3)-naeherung_r(i,1))/((naeherung_r(i,4)-naeherung_r(i,2))^2 + (naeherung_r(i,3)-
    naeherung_r(i,1))^2))*(200/pi); %dR/dxZ
end

naeherung_s=[Such_y_a(length(richtungen)+1:length(alle),1) Such_x_a(length(richtungen)+1:length(alle),1) Such_y_a(length(
  richtungen)+1:length(alle),2) Such_x_a(length(richtungen)+1:length(alle),2)]; %Struktur: [y-Koord.Stdpkt. x-Koord.
  Stdpkt. y-Koord.Zielpkt. x-Koord.Zielpkt.]
255

%Ableitungen nach Strecken berechnen
for i=1:length(sGK_imp)
  ableitungen_s(i,1)=(naeherung_s(i,1)-naeherung_s(i,3))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dyS
  ableitungen_s(i,2)=(naeherung_s(i,2)-naeherung_s(i,4))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dxS
260  ableitungen_s(i,3)=(naeherung_s(i,3)-naeherung_s(i,1))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dyZ
  ableitungen_s(i,4)=(naeherung_s(i,4)-naeherung_s(i,2))/sqrt(((naeherung_s(i,4)-naeherung_s(i,2))^2 + (naeherung_s(i,3)-
    naeherung_s(i,1))^2)); %dS/dxZ
end

%% Ränderungsmatrix G aufstellen
265 G1= repmat([0 1],1,length(Koord)/2);
G2= repmat([1 0],1,length(Koord)/2);
null=zeros(1,length(X0_o));
G=[G1 null;G2 null];

270 for i=1:2:length(Koord)
  G(3,i+1)=Koord(i,1);
end
for i=2:2:length(Koord)
  G(3,i-1)=-Koord(i,1);
275 end

null_N=zeros(3,3);
null_xk=zeros(3,1);

280 %% A-Matrix aufstellen
%A-Matrix-Größe definieren - ohne Ableitungen für Orientierungsunbek.
A=zeros(length(L),length(Koord));

%Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Standpunkt gehören
285 for i=1:gr(1,1)
  for j=1:length(Index)
    if richtungen(i,1)==Index(j,1)
      A(i,Index(j,2))=ableitungen_r(i,1); %Ableitung nach der y-Koordinate des Standpunktes einfüllen
      A(i,(Index(j,2)+1))=ableitungen_r(i,2); %Ableitung nach der x-Koordinate des Standpunktes einfüllen
290    end
  end
end

%Auffüllen der A-Matrix mit den Richtungsableitungen, die zum jeweiligen Zielpunkt gehören
295 for i=1:gr(1,1)
  for j=1:length(Index)
    if richtungen(i,2)==Index(j,1)
      A(i,Index(j,2))=ableitungen_r(i,3); %Ableitung nach der y-Koordinate des Zielpunktes einsetzen
      A(i,(Index(j,2)+1))=ableitungen_r(i,4); %Ableitung nach der x-Koordinate des Zielpunktes einsetzen
300    end
  end
end
end

```



```

305 % Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Standpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,1)==Index(j,1)
            A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,1);
            A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,2);
310        end
    end
end

315 %Auffüllen der A-Matrix mit den Streckenableitungen, die zum jeweiligen Zielpunkt gehören
for i=1:length(sGK_imp)
    for j=1:length(Index)
        if sGK_imp(i,2)==Index(j,1)
            A(gr(1,1)+i, Index(j,2))=ableitungen_s(i,3);
320            A(gr(1,1)+i, (Index(j,2)+1))=ableitungen_s(i,4);
        end
    end
end

325 %eigene Teilmatrix A0 für das Auffüllen mit den Ableitungen nach den Orientierungsunbekannten
saetze(1,3)=0;
for i=1:length(saetze)-1
    saetze(i+1,3)=saetze(i,3)+saetze(i,2);    %Spaltenposition für die Ableitung nach der O.unbek. berechnen
end

330 A0=zeros(gr(1,1), length(X0_o));
for i=1:length(saetze)
    n=saetze(i,3);
    for j=1:saetze(i,2)
335        [row]=find(richtungen(:,1)==saetze(i,1));
        [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
        s=s+offset;
        n=n+j;
340        for k=s(1,1):s(length(s),1)
            A0(k,n)=-1;
        end
        n=n-j;
    end
end
345

%A0 um Nullbereich für die Streckenbeobachtungen erweitern
A0=[A0;zeros(length(sGK_imp), length(X0_o))];
A=[A A0];
350 gA=size(A);
    A_alle=A;    %für Berechnung der endgültigen Lösung
    A_1=A;    %Hilfsvariable für Eliminieren
    A_1(:, unbek+1)=1:length(A);    %Nummerierung der Messungen in 2.Spalte
    A_0=A;
355 A_0(:, unbek+1)=1:length(A);    %Nummerierung der Messungen in 2.Spalte

iteration=1;
abbruch=0;    %Zähler für den Abbruch wegen nicht ausreichender Konditionierung

360 while iteration ~=501
    %1. Auswahlverfahren
    for i=1:length(saetze_r)
        for j=1:saetze_r(i,2)
365            [row]=find(richtungen(:,1)==saetze_r(i,1));    %Zeilen, die einen gemeinsamen Standpunkt haben
            [s]=find(richtungen(row(1,1):row(length(row),1),7)==j);    %Zeilen, die einen gemeinsamen Satz haben
            offset=row(1,1)-1;
            s=s+offset;
            abstand=s(1,1)-1;
            auswahl=randsample(length(s),1);    %1 RICHTUNG auswählen, die verwendet werden soll
370            alle(auswahl+abstand,4)=1;    %ausgewählte RICHTUNG mit "1" markieren
        end
    end

    for i=1:length(saetze_s)
375        for j=1:saetze_s(i,2)
            [row]=find(sGK_imp(:,1)==saetze_s(i,1));    %Zeilen, die einen gemeinsamen Standpunkt haben
            [s]=find(sGK_imp(row(1,1):row(length(row),1),4)==j);    %Zeilen, die einen gemeinsamen "Satz" haben
            offset=anzR_alle + row(1,1)-1;
            s=s+offset;
            abstand=s(1,1)-1;
            auswahl=randsample(length(s),1);    %1 STRECKE aussuchen, die verwendet werden soll
380            alle(auswahl+abstand,4)=1;    %ausgewählte STRECKE mit "1" markieren
        end
    end
end
385

```

```

for i=0:length(alle)-1      %von der letzten zur ersten Messung, weil so die falsche Zeilenzuordnung umgangen wird
    if alle(length(alle)-i,4)==0
        A_1(length(alle)-i,:)=[]; %nicht ausgewählte "0" Beob. werden gelöscht
        sigma2_1(length(alle)-i,:)=[];
390         L_1(length(alle)-i,:)=[];
        L0_1(length(alle)-i,:)=[];
    else
        A_0(length(alle)-i,:)=[];
        sigma2_0(length(alle)-i,:)=[];
395         L_0(length(alle)-i,:)=[];
        L0_0(length(alle)-i,:)=[];
    end
end

400 alle_r=alle(1:anzR_alle,:);      %alleB aufteilen in Ri u Str
alle_s=alle(length(alle_r)+1:length(alle),:);

for i=0:anzR_alle-1
    if alle_r(anzR_alle-i,4)==0
405         richtungen_1(anzR_alle-i,:)=[]; %nicht ausgewählte Ri löschen
            o_1(anzR_alle-i,:)=[];          %dementsprechende Orientierungen
            t_1(anzR_alle-i,:)=[];          %dementsprechende Richtungswinkel
    else
        richtungen_0(anzR_alle-i,:)=[]; %ausgewählte Beob. werden in gegenteiliger Matrix gelöscht
410         o_0(anzR_alle-i,:)=[];
            t_0(anzR_alle-i,:)=[];
    end
end

415 alle_rs_h=[alle_r;alle_s];
alle_rs=[alle_r;alle_s];
alle_rs_rest=[alle_r;alle_s];      %Matrix duplizieren, damit für weitere Auswahl verwendet werden kann

for i=0:length(alleB)-1
420     if alle_rs_h(length(alleB)-i,4)==0      %alle nicht ausgewählte Zeilen löschen
        alle_rs(length(alleB)-i,:)=[];
    else
        alle_rs_rest(length(alleB)-i,:)=[]; %Matrix mit den verbleibenden Beob.
    end
425 end

%% 2.Auswahlverfahren (3 Beobachtungen werden ausgesucht)
alle_rs_rest=alle_rs_rest(:,1:3);      %letzte Spalte löschen, damit Beob. neu markiert werden können
auswahl=randsample(length(alle_rs_rest),wahl,false); %3 Beobachtungen auswählen damit insgesamt 42 Beob.; auswahl=
    Zeilennr NICHT Beob.nummer

430 for i=1:length(auswahl)
    alle_rs_rest(auswahl(i,1),4)=1; %neu ausgewählte Beob. markieren
end

435 for i=0:length(alle_rs_rest)-1
    %von der letzten zur ersten Messung, weil so die falsche Zeilenzuordnung
    umgangen wird
    if alle_rs_rest(length(alle_rs_rest)-i,4)==0
        A_0(length(alle_rs_rest)-i,:)=[];      %nicht ausgewählte "0" Beob. werden gelöscht
        sigma2_0(length(alle_rs_rest)-i,:)=[];
440         L_0(length(alle_rs_rest)-i,:)=[];
        L0_0(length(alle_rs_rest)-i,:)=[];
    end
end

445 aR=0;
    %zählen, wieviele von den nach der 1.Auswahl übrigen Beob. Ri sind
for i=1:length(alle_rs_rest)
    if alle_rs_rest(i,3)<=anzR_alle      %alle mit Zeilennr. <anzR_alle sind Richtungen
        aR=aR+1;
    end
450 end

rest_r=alle_rs_rest(1:aR,:);
rest_s=alle_rs_rest(aR+1:length(sigma2_0),:);

455 for i=0:length(rest_r)-1
    if rest_r(length(rest_r)-i,4)==0
        richtungen_0(length(rest_r)-i,:)=[]; %nicht ausgewählte Ri herauschneiden
        o_0(length(rest_r)-i,:)=[];          %dementsprechende Orientierungen
        t_0(length(rest_r)-i,:)=[];          %dementsprechende Richtungswinkel
460     end
end

A_sample=[A_1;A_0];
A_sample=sortrows(A_sample,unbek+1);
465 sigma2_sample=[sigma2_1;sigma2_0];

```

```

sigma2_sample=sortrows(sigma2_sample,2);
L0_sample=[L0_1;L0_0];
L0_sample=sortrows(L0_sample,2);
L_sample=[L_1;L_0];
470 L_sample=sortrows(L_sample,2);
o_sample=[o_1;o_0];
o_sample=sortrows(o_sample,3);
t_sample=[t_1;t_0];
t_sample=sortrows(t_sample,4);
475 richtungen_sample=[richtungen_1;richtungen_0];
richtungen_sample=sortrows(richtungen_sample,8);

%ausgewählte Strecken suchen
platz=1;
480 for i=1:unbek
    if L_sample(i,2) > anzR_alle
        s_sample(platz,3)=L_sample(i,1);
        s_sample(platz,1:2)=alle(L_sample(i,2),1:2);
        s_sample(platz,8)=alleB(L_sample(i,2),3);
485        platz=platz+1;
    end
end

aR=0; %zählen, wieviele von den ausgewählten Beob. Ri sind
490 for i=1:length(L0_sample)
    if L0_sample(i,2)<=anzR_alle %alle mit Zeilennr. <anzR_alle sind Richtungen
        aR=aR+1;
    end
end
495

%L0_r mit den ausgewählten Ri neu berechnen
standpunkte_r_sample =unique(richtungen_sample(:,1));

%neu Matrix erstellen, die die verwendeten Standpunkte und ihre Anzahl von Sätzen zeigt
500 for i=1:length(standpunkte_r_sample)
    [row]=find(richtungen_sample(:,1)==standpunkte_r_sample(i,1));
    bet_sample=richtungen_sample(row(1,1):row(length(row),1),7);
    anzahl_sample=unique(bet_sample);
    anzahl2_sample=anzahl_sample(length(anzahl_sample),1);
505    saetze_sample(i,1)=standpunkte_r_sample(i,1);
    saetze_sample(i,2)=anzahl2_sample;
end

m=1;
510 for i=1:length(saetze_sample)
    for j=1:saetze_sample(i,2)
        [row]=find(richtungen_sample(:,1)==saetze_sample(i,1));
        [s]=find(richtungen_sample(row(1,1):row(length(row),1),7)==j);
        offset=row(1,1)-1;
515        s=s+offset;
        mittelo_sample=mean(o_sample(s(1,1):s(length(s),1)));
        X0_o_sample(m,1)=mittelo_sample;
        m=m+1;
        for k=s(1,1):s(length(s),1)
520            o_sample(k,2)=mittelo_sample;
        end
    end
end
525

%Berechnung von L0 für Richtungen
L0_r_sample=t_sample(:,3)-o_sample(:,2);

for i=1:length(L0_r_sample)
530    if L0_r_sample(i,1)<0
        L0_r_sample(i,1)=L0_r_sample(i,1)+400;
    else
        L0_r_sample(i,1)=L0_r_sample(i,1);
    end
end
535

L0_sample=[L0_r_sample;L0_sample(aR+1:length(L0_sample),1)];
l_sample=L_sample(:,1)-L0_sample(:,1);

540 %P-Matrix aufstellen
S11=eye(length(L_sample),length(L_sample));
for i=1:length(L_sample)
    S11(i,i)=sigma2_sample(i,1);
end
545 sigma02=1;
Q11=(1/sigma02)*S11;
P_sample=Q11^-1;

```

```

X0_sample=[Koord;X0_o_sample];
N_sample=[A_sample(:,1:unbek)'+P_sample*A_sample(:,1:unbek) G';G null_N];
550 condi=rcond(N_sample);

if condi<=10^-15
    disp('System nicht ausreichend konditioniert. Ergebnis ist nicht zuverlässig und wird verworfen.')
    abbruch=abbruch+1;
555 else
    xk_sample=N_sample^-1*[A_sample(:,1:unbek)'+P_sample*I_sample;null_xk];
    x_sample=xk_sample(1:length(X0_sample),1);

    Xd_sample=X0_sample+x_sample;
560 v_sample=A_sample(:,1:unbek)*x_sample-l_sample;
    Ld_sample=L_sample(:,1)+v_sample;

    samples(1:length(sigma2_sample),iteration)=sigma2_sample(:,2);

565 % Iterativer Ausgleich & Hauptprobe
    [alle_l2lts] = iterativ(koord, richtungen_sample, s_sample);

    %1. Durchlauf (Q1)
    [v_sort1, v_h1 ] = verbesserungenlts1 (sGK_imp, G1, saetze, alleR, Index, iteration, h);
570 vH_1(:,iteration)=v_h1(:,2);

    %2. Durchlauf (Q2)
    [abbruch2, richtungen2, orig_satz2, saetze2, alle_l2lts] = L2lts2(richtungen, v_h1, sGK_imp, koord);
    if abbruch2==0 %sofern L2-Ausgleich erfolgreich und ausreichend konditioniert
575 [yx_r, Id_r, v_sort2, v_h2] = verbesserungenlts2 (sGK_imp, G1, Index, iteration, beob, orig_satz2, richtungen, h);
        vH_2(:,iteration)=v_h2(:,2);

        %3.Durchlauf (Q3)
        [abbruch3, richtungen3, orig_satz3, alle_l2lts] = L2lts3(richtungen, v_h2, sGK_imp, koord);
580 if abbruch3==0
            [v_sort3] = verbesserungenlts3 (sGK_imp, G1, Index, iteration, beob, orig_satz3, richtungen, h);
            vH_3(:,iteration)=v_sort3(1:h,2);

            anzahlverb(iteration,1)=length(v_sort1); %zeigt, für wieviele Beob. Verbesserungen berechnet werden können
585 anzahlverb(iteration,2)=length(v_sort2);
            anzahlverb(iteration,3)=length(v_sort3);
            iteration=iteration+1;
        end
    end
590 end

%Matrizen reseten für neuen Durchlauf
A_1=A;
A_1(:,unbek+1)=1:length(A);
595 A_0=A;
A_0(:,unbek+1)=1:length(A);
sigma2_1=sigma2;
sigma2_1(:,2)=1:length(sigma2);
sigma2_0=sigma2;
600 sigma2_0(:,2)=1:length(sigma2);
L0_1=L0;
L0_1(:,2)=1:length(L0);
L0_0=L0;
L0_0(:,2)=1:length(L0);
605 o_1=o;
o_1(:,3)=1:length(o);
o_0=o;
o_0(:,3)=1:length(o);
t_1=t;
610 t_1(:,4)=1:length(t);
t_0=t;
t_0(:,4)=1:length(t);
L_1=L;
L_1(:,2)=1:length(L_1);
615 L_0=L;
L_0(:,2)=1:length(L_0);
richtungen_1=richtungen;
richtungen_1(:,8)=1:length(richtungen);
richtungen_0=richtungen;
620 richtungen_0(:,8)=1:length(richtungen);
alle=alleB;
clearvars s_sample
end

625 %% Weitere Berechnung mit den 10 kleinsten Q3
Q3(:,2)=1:length(Q3);
Q3_sort=sortrows(Q3,1);
Q3_min=Q3_sort(1:10,1:2);

```

```

630 Xd_Q=zeros(unbek,10);
    r_sammel=zeros(beob,10);
    vH_Q2=zeros(beob,10);

    for q_ind=1:10
635
        schleife=1;           %Schleifenzähler bis zur Konvergenz von Q
        iter=Q3_min(q_ind,2); %betreffende Iteration suchen
        v_hQ=v_h3(:,iter);
        diff_Q=1;
640

        while diff_Q > 0.00000001
            [Qxx, Ql1dd, Qvv, r, s02, sld, richtungenQ, orig_satzQ, alle_l2lts] = L2ltsQ3(v_hQ, richtungen, sGK_imp, koord);
            [v_alle_aktuell, v_sortQ] = verbesserungenltsQ3 (sGK_imp, Gl, Index, beob, schleife, richtungen, orig_satzQ, h);
645            sld_sammel(:,q_ind)=sld;
            vH_Q(:,q_ind)=v_sortQ(1:h,2);
            vH_Q2(1:length(v_alle_aktuell),q_ind)=v_alle_aktuell(:,1); %alle Verbesserungen, für 3sigma-Vergleich, in
                Beobachtungsreihenfolge
            s02_sammel(q_ind,1)=s02;
            r_sammel(1:length(r),q_ind)=r;
650

            if schleife==1
                diff_Q=Q3_min(q_ind,1)-Q(schleife,1);
                Q_finals(q_ind,1)=Q(schleife,1);
                schleife=schleife+1;
655                Xd_Q(1:length(Xd_sample),q_ind)=Xd_sample;
            else
                diff_Q=Q(schleife-1,1)-Q(schleife,1);
                Q_finals(q_ind,1)=Q(schleife,1);
                schleife=schleife+1;
660                Xd_Q(1:length(Xd_sample),q_ind)=Xd_sample;
            end
        end
    end

665 minQ=min(Q_finals);
    row_minq=find(Q_finals==minQ); %zeigt Zeile des kleinsten Q-konvergiert
    Xd_solve=Xd_Q(:,row_minq); %enthält die ausgeglichenen Unbekannten des minimalsten Q-konvergiert

    %% Grobfehlersuche
670 s02_final=s02_sammel(row_minq(1,1),1);
    sld_final=sld_sammel(:,row_minq(1,1));

    %zählen, wieviele von den Beob. Ri sind
    aR=0;
675 for i=1:length(vH_Q(:,row_minq(1,1)))
        if vH_Q(i,row_minq(1,1))<=anzR_alle %alle mit Zeilennr. <anzR_alle sind Richtungen
            aR=aR+1;
        end
    end
680

    for i=1:anzR_alle
        if vH_Q2(i,row_minq(1,1))>0.0015 %testen, ob Verbesserung größer als 3sigma, Richtungen
            err(i,1)=1;
        end
    end
685

    for i=anzR_alle+1:beob
        if vH_Q2(i,row_minq(1,1))*10<0.015 %testen, ob Verbesserung größer als 3sigma, Strecken
            err(i,1)=0;
        else
690            err(i,1)=1;
        end
    end

    end

    toc %Ende der Zeitmessung

```

## G.2 Funktion für die Verbesserungen

Im Hauptprogramm wird mehrmals eine Funktion für die Berechnung der Verbesserungen aufgerufen. Die Funktion „verbesserungenlts1“ entspricht der Funktion, welche bereits bei RANSAC eingesetzt wird. Die weiteren Funktionen „verbesserungenlts2“, „verbesserungenlts3“ und „verbesserungenltsQ3“ basieren ebenfalls auf der Grundfunktion von RANSAC und unterscheiden sich untereinander nur in der Benennung der Variablen. Sie beinhalten zudem die in Kapitel 4.7 erwähnte Erweiterung für eine reduzierte Anzahl an Orientierungsunbekannten (siehe nachfolgender Matlab-Code).

```

1 function [yx_r, Ld_r, v_sort2, v_h2] = verbesserungenlts2 (sGK_imp, G1, Index, iteration, beob, orig_satz2, richtungen, h)
   global alle_l2lts Xd_sample anzR_alle v_h3 Q Q1 Q2 Q3

   %% Strecken
6  [uniqueS]=unique(sGK_imp(:,1:2),'rows'); %findet alle unterschiedlichen Visuren

   Xd_koord=Xd_sample(1:length(G1),1);
   Xd_hilfe=reshape(Xd_koord,2,length(G1)/2);
   Xd_hilfe=Xd_hilfe';
11  Xd_hilfe=[Index(1:length(G1)/2,1) Xd_hilfe];

   Xd_s_y=uniqueS;
   Xd_s_x=uniqueS;

16  %ausgeglichene y-Koordinaten einsetzen
   for i=1:length(uniqueS)
       for j=1:length(Index)
           ll=Xd_s_y==Xd_hilfe(j,1);
           Xd_s_y(ll)=Xd_hilfe(j,2);
21  end
   end

   %ausgeglichene x-Koordinaten einsetzen
26  for i=1:length(uniqueS)
       for j=1:length(Index)
           mm=Xd_s_x==Xd_hilfe(j,1);
           Xd_s_x(mm)=Xd_hilfe(j,3);
           end
       end
31  end

   yx_s=[Xd_s_y(:,1) Xd_s_x(:,1) Xd_s_y(:,2) Xd_s_x(:,2)]; %Schema yStdpkt xStdpkt yZielpkt yZielpkt

   for i=1:length(uniqueS)
       S(i,1)=sqrt((yx_s(i,3)-yx_s(i,1))^2 + (yx_s(i,4)-yx_s(i,2))^2); %Strecken aus den ausgeglichenen Koordinaten berechnen
36  end

   for i=1:length(uniqueS)
       [r]=find(sGK_imp(:,1)==uniqueS(i,1) & sGK_imp(:,2)==uniqueS(i,2)); %ursprünglichen Visuren suchen
       Ld_s(r(1,1):r(length(r),1),3)=S(i,1); %und die ausgeglichenen Strecken in die entsprechenden
           Zeilen schreiben
41  end

   Ld_s(:,1:2)=sGK_imp(:,1:2);

46  v_s=Ld_s(:,3)-sGK_imp(:,3); %Verbesserungen für alle Strecken

   %% Richtungen
   Xd_o=Xd_sample(length(G1)+1:length(Xd_sample),1); %ausgeglichene Orientierungen extrahieren

   %Beginn neuer Abschnitt: falls Zahl der Orientierungsunbekannte durch Auswahl der h Beobachtungen verringert wird
51  standpunkte_r2 =unique(orig_satz2(:,1));
   gstdpk_r2=size(standpunkte_r2);

   %Matrix erstellen, die die Standpunkte und ihre Anzahl von Sätzen zeigt
   saetze_neu=zeros(5,9);
56  for i=1:gstdpk_r2(1,1)
       [row]=find(orig_satz2(:,1)==standpunkte_r2(i,1));
       bet=orig_satz2(row(1,1):row(length(row),1),3);
       anzahl=unique(bet);
       saetze_neu(i,1)=standpunkte_r2(i,1);
61  saetze_neu(i,2:length(anzahl)+1)=anzahl';
       saetze_neu(i,10)=length(anzahl);
   end

```

```

saetze_neu(~any(saetze_neu,2),:)=[]; %Nullzeilen löschen (wenn nicht alle Standpunkte verwendet)
saetze_neu(:,~any(saetze_neu))=[]; %Nullspalten löschen (je nach dem wieviele Sätze pro Standpunkt verwendet werden)
66
o_alle2=zeros(anzR_alle,1);
ssn=size(saetze_neu);
m=1;
for i=1:ssn(1,1)
71   for j=1:saetze_neu(i,ssn(1,2))
       [row]=find(richtungen(:,1)==saetze_neu(i,1));
       [s]=find(richtungen(row(1,1):row(length(row),1),7)==saetze_neu(i,j+1));
       offset=row(1,1)-1;
       s=s+offset;
76   for k=s(1,1):s(length(s),1)
       o_alle2(k,1)=Xd_o(m,1);
       end
       m=m+1;
81   end

richtungenv2=richtungen;
richtungenv2(:,8)=1:length(richtungen(:,1));

86   for i=0:anzR_alle-1
       if o_alle2(length(o_alle2)-i,1)==0
           richtungenv2(length(o_alle2)-i,:)=[]; %Vektor, der die noch verwendbaren (da vorhandene Ori.unbek) Ri anzeigt
       end
91   end

o_alle2(~any(o_alle2,2),:)=[]; %Nullzeilen löschen (wenn nicht alle Standpunkte verwendet)
%Ende neuer Abschnitt

Xd_r_y=richtungenv2(:,1:2);
96   Xd_r_x=richtungenv2(:,1:2);

%y-Koordinaten er- bzw. einsetzen
for i=1:length(richtungenv2(:,1))
    for j=1:length(Xd_hilfe)
101     jj=Xd_r_y==Xd_hilfe(j,1);
        Xd_r_y(jj)=Xd_hilfe(j,2);
    end
end

106 %x-Koordinaten er- bzw. einsetzen
for i=1:length(richtungenv2(:,1))
    for j=1:length(Xd_hilfe)
        jj=Xd_r_x==Xd_hilfe(j,1);
        Xd_r_x(jj)=Xd_hilfe(j,3);
111     end
end

%Zuordnung der ausgeglichenen Koordinaten zu den ausgeglichenen Orientierungen
yx_r=[Xd_r_y(:,1) Xd_r_x(:,1) Xd_r_y(:,2) Xd_r_x(:,2) o_alle2(:,1)]; %Schema: [yS xS yZ xZ o]
116

%Berechnung der Richtungen aus den ausgeglichenen Koord & Orientierungen
%Berechnung der Richtungswinkel mit Quadrantenzuordnung
for i=1:length(richtungenv2(:,1))
    Ld_r(i,1)=yx_r(i,3)-yx_r(i,1); %Zähler der Richtungswinkel
121     Ld_r(i,2)=yx_r(i,4)-yx_r(i,2); %Nenner der Richtungswinkel
end

for i=1:length(Ld_r)
    if Ld_r(i,1) > 0 && Ld_r(i,2) < 0
126     Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+200;
    end
    if Ld_r(i,1) < 0 && Ld_r(i,2) < 0
        Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+200;
    end
131     if Ld_r(i,1) < 0 && Ld_r(i,2) > 0
        Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi)+400;
    end
    if Ld_r(i,1) > 0 && Ld_r(i,2) > 0
        Ld_r(i,3)=(atan(Ld_r(i,1)/Ld_r(i,2)))*(200/pi);
136     end
end

for i=1:length(richtungenv2(:,1))
141     Ld_r(i,4)=Ld_r(i,3)-o_alle2(i,1);
end

```

```

146 for i=1:length(Ld_r)
      if Ld_r(i,4)<0
          Ld_r(i,4)=Ld_r(i,4)+400;
      end
    end
151 v_r=richtungenv2(:,6)-Ld_r(:,4);
    v_alle_aktuell=[v_r;v_s/10];
    num2=[richtungenv2(:,8);(anzR_alle+1:beob)'];
    v_alle_aktuell=[abs(v_alle_aktuell) num2];
156 v_sort2=sortrows(v_alle_aktuell,1);
    v_h2=v_sort2(1:h,:);

    Q2(iteration,1)=sum(v_sort2(1:h,1).^2);
161 end

```

### G.3 Funktion für den Ausgleich nach der L2-Norm

Die aufgerufenen Funktionen „iterativ“ und „L2lts2/3/Q3“ sind im Wesentlichen ident. Für „L2lts2“ und „L2lts3“ kommt ein kurzer Abschnitt über die Prüfung einer ausreichenden Konditionierung der  $N$ -Matrix hinzu. Der zugehörige Rechengang ist bereits aus den vorangegangenen Codeausschnitten bekannt und wird an dieser Stelle nicht mehr explizit angeführt.