

Studienrichtung Vermessungswesen
Technische Universität Wien

**G E O W I S S E N S C H A F T L I C H E
M I T T E I L U N G E N**

Heft 36

**o. Univ.-Prof. Dr. Hans Schmid
zum 70. Geburtstag**

Veröffentlichung des Institutes für Landesvermessung u. Ingenieurgeodäsie
Abt. Landesvermessung

Studienrichtung Vermessungswesen
Technische Universität Wien

**G E O W I S S E N S C H A F T L I C H E
M I T T E I L U N G E N**

Heft 36

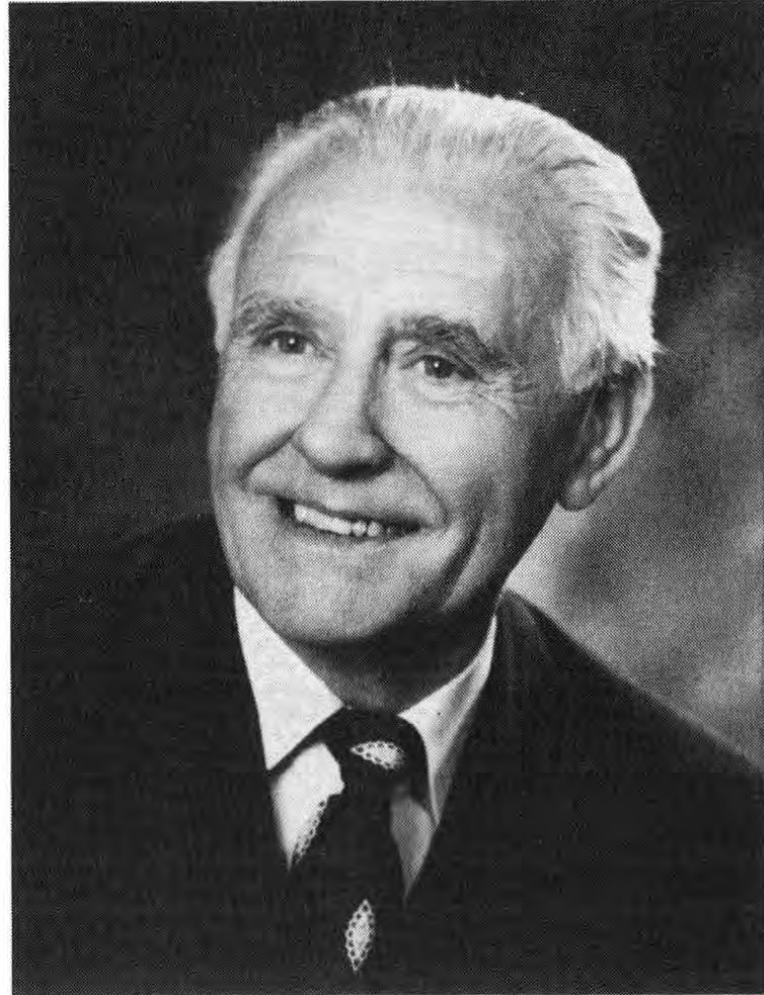
**o. Univ.-Prof. Dr. Hans Schmid
zum 70. Geburtstag**

Veröffentlichung des Institutes für Landesvermessung u. Ingenieurgeodäsie
Abt. Landesvermessung

Herausgeber und Verleger: O.Rat Dipl.-Ing. E.Korschineck
interim.Leiter der Abteilung Landesvermessung
des Instituts für Landesvermessung und Ingenieurgeodäsie
der Technischen Universität Wien
A-1040 Wien, Gußhausstraße 27-29

Die Kosten für den Druck wurden aus der ordentlichen Dotation
des Instituts für Landesvermessung und Ingenieurgeodäsie
der TU Wien getragen.

Druck: HTU Wirtschaftsbetriebe Ges.m.b.H.
1040 Wien
Auflage: 150 Stück



Inhaltsverzeichnis

KORSCHINECK, E.:	Vorwort	V
BRETTTERBAUER, K.:	Laudatio	VII
STANEK, H.:	Analyseausgleichung zur Interpretation geodätischer Netze	1-133
TÖPFER, H.:	Transformation von geodätischen Programmen	1-114

V O R W O R T

Am 26. April 1990 beging

Em.O.Univ.Prof.Dipl.-Ing.Dr.techn. Hans SCHMID

seinen 70. Geburtstag. Aus diesem Anlaß widmet die Abteilung Landesvermessung des Instituts für Landesvermessung und Ingenieurgeodäsie der Technischen Universität Wien ihrem ehemaligen Institutsvorstand ein Sonderheft der Geowissenschaftlichen Mitteilungen.

Anstatt, wie üblich, in einem Vorwort den Lebenslauf des Jubilars zu schildern, habe ich mir erlaubt, die von Univ.Prof. Dr. Bretterbauer geistvoll und launig gehaltene Laudatio hier wiederzugeben.

Die Autoren der beiden wissenschaftlichen Beiträge dieses Heftes sind ehemalige Schüler und wissenschaftliche Mitarbeiter von Professor Schmid. Ihre hier in gekürzter Fassung publizierten bemerkenswerten Dissertationen sind über Anregung und unter wissenschaftlichen Betreuung von Professor Schmid entstanden.

Sein Wirken als Ordinarius fiel in eine Zeit der stürmischen Entwicklung der Geodäsie sowohl auf technologischem als auch theoretischem Gebiet. Dem Rechnung tragend, modernisierte er die Ausstattung des Instituts und gestaltete den Vorlesungsstoff neu. Er verstand es, wie kein anderer, neue theoretische Erkenntnisse in die Praxis umzusetzen. Diese Praxisnähe, interessante Wissensvermittlung und ausgezeichnete Rethorik wurden von den Studenten stets durch gut besuchte Vorlesungen honoriert.

Seinen Mitarbeitern war Professor Schmid ein konzilianter Vorgesetzter, der sie zu motivieren wußte, für ihre Anliegen stets ein offenes Ohr hatte, ihre wissenschaftlichen Ambitionen förderte und sie von seinem Wissen und praktischen Erfahrung profitieren ließ.

Leider wurde er allzufrüh von einem heimtückischen Leiden befallen, das seinen Tatendrang hemmte und schließlich 1988 zu seiner Emeritierung führte.

Abschließend möchte ich Herrn Professor Schmid, den ich 35 Jahre als Mitarbeiter und Freund auf seinem Lebensweg begleiten durfte, meinen ganz persönlichen Dank aussprechen, für all die Jahre der Zusammenarbeit, die für mich anfangs eine harte Schule, später dann eine gedeihliche Auseinandersetzung mit seinen fordernden Ideen war.

Für die weitere Zukunft wünsche ich meinem Lehrer und Freund noch viele erfüllte Jahre in bestmöglicher Gesundheit im Kreise seiner Familie und Freunde.

E. Korschineck

Laudatio

auf Univ. Prof. Dr. Hans Schmid

gehalten im Rahmen des Festkolloquiums zu seinem 70. Geburtstag
an der TU Wien.

Verehrte Festgäste! Wir sind hier zusammengekommen, um einem bemerkenswerten Kollegen und hochgeschätzten Freund zu seinem 70. Geburtstag unsere Ehre zu erweisen. Deshalb gilt mein ganz besonderer Willkommensgruß unserem Jubilar Hans Schmid. Gerne habe ich als Vorsitzender der Fachgruppe "Geowissenschaften" die schwierige und ehrenvolle Aufgabe übernommen, die Verdienste von em. Univ. Prof. Dr. techn. Dipl. Ing. Hans Schmid zu würdigen. Der 70. Geburtstag war schon immer ein besonderes Datum im Leben eines Universitätsprofessors, bedeutete es doch die Befreiung von der Lehrverpflichtung. Prof. Schmid mußte diese Befreiung schon etwas früher beantragen, war ihm doch die Lehrverpflichtung nicht etwa lästig, aber zunehmend belastend geworden. Bis dahin aber kann Prof. Schmid auf eine nicht alltägliche Hochschulkarriere verweisen. Es möge daher zunächst der Lebensweg und die berufliche Laufbahn unseres Jubilars kurz in Erinnerung gerufen werden.

Eine solche Schilderung ist im Grunde anmaßend, heißt es doch ein wenig, einem erfolgreichen Manne sein Leben zu rechtfertigen. Bedarf schon das Leben an sich keiner Rechtfertigung, so am allerwenigsten das von Prof. Schmid. Es ist bekannt, daß Hans Schmid seit langem und mit leidenschaftlichem Interesse sich den Problemen und Erkenntnissen der Astrophysik und Kosmologie zugewendet hat. Ich will deshalb in Anspielung auf diese Wissenschaften und auf die Relativitätstheorie seinen Lebensweg in dem vierdimensionalen Raum-Zeit-Kontinuum, in dem wir leben, eine Minkowski'sche Weltlinie nennen. Diese Weltlinie hatte ihren Startpunkt am 26. April 1920 in Olmütz. Zwischen diesem Weltpunkt und jenem, an dem Prof. Schmid heute steht, hätte seine Weltlinie viele Formen annehmen können. Es gibt aber unter allen möglichen Weltlinien eine ausgezeichnete. Hans Schmid hat jene Form gewählt, bei der, in der Sprache der Relativitätstheorie, der zeitartige Abstand zwischen den zwei genannten Weltpunkten ein Maximum wird, der raumartige Abstand dagegen ein Minimum! Eine solche Weltlinie nennt man eine geodätische Linie. Ist es allein schon naheliegend, den Lebensweg eines Geodäten als geodätische Linie zu deuten, drängt sich diese Deutung im Falle von Hans Schmid ganz besonders auf. Die beiden genannten Extrema bedürfen noch einer Interpretation. Für Hans Schmid bedeutet der maximale zeitartige Abstand, er hat in die Spanne seines bisherigen Lebens ein Maximum an Aktivitäten und Erfolgen gepreßt, das raumartige Minimum aber besagt, er hat seine Ziele immer auf direktem und kürzestem Wege angesteuert.

Seine geodätische Linie mit Ausgangspunkt Olmütz hatte ein Anfangsazimut, das nach Wien wies. Dort besuchte er die Volksschule und die Realschule. Im geschichtsträchtigen Frühjahr 1938 maturierte er mit Auszeichnung, im Herbst desselben Jahres begann er das Studium des Vermessungswesens an der TH Wien. Damit hat eine nicht alltägliche akademische Laufbahn begonnen, die durch Krieg und Gefangenschaft verzögert, aber nicht ab-

gebrochen wurde. Nach der 1. Staatsprüfung, wieder mit Auszeichnung, kam die Einberufung zur Deutschen Wehrmacht mit Fronteinsätzen in Rußland und Italien verbunden mit Kriegsverwundung. Im Februar 1946 konnte Hans Schmid aus amerikanischer Kriegsgefangenschaft heimkehren und das Studium schon im Oktober als Dipl. Ing. abschließen. Wie? Mit Auszeichnung natürlich.

Um bei dem Gleichnis der geodätischen Linie zu bleiben: Sehr glatt ist deren Verlauf bis dahin wohl nicht gewesen, bedenkt man die äußeren Verhältnisse der Jahre 1920 bis 1946. Eine Schilderung dieser Zeitumstände sei hier vermieden; die älteren Menschen haben sie erlebt, das Vorstellungsvermögen der jungen aber würde überfordert.

Hans Schmid ging seinen Weg mit den extremalen Eigenschaften weiter. 1948 Hochschulassistent, Juni 1949 Promotion mit Auszeichnung zum Doktor der technischen Wissenschaften. Auf grund dieser herausragenden Studienerfolge wurde Schmid 1950 die "Karoline und Guido Krafft-Medaille" verliehen, eine Auszeichnung, die der heutigen Promotion sub auspiciis praesidentis entspricht. Weiter ging's mit der Erwerbung der Befugnis als Ingenieurkonsulent 1950 und der Habilitation in den Fächern Photogrammetrie und Katastertechnik 1955. 1964, als erster Höhepunkt der Hochschulkarriere, kam die Ernennung zum a. o. Prof., was damals nicht vielen Dozenten beschieden war. Nebenbei hat Schmid zielstrebig sein Ingenieurbüro aufgebaut.

Manch einer hätte nun begonnen, sich mit dem Erreichten zufrieden zu geben und die Zügel schleifen zu lassen. Das hätte einen unschönen Knick in der geodätischen Linie seines Lebens ergeben. Dies ließ Schmid nicht zu, er baute nur eine kleine raumartige Schleife ein. 1968 nämlich wurde er als Ordinarius an das damalige III. Geodätische Institut der TH Graz berufen, wo er aber nur drei Jahre verblieb. Denn schon 1971 kam die Berufung an jenes Institut, an dem er viele Jahre als Assistent und Dozent gewirkt hat. Nur wer Ähnliches erlebt hat, kann ermessen, welche Freude und Genugtuung es bedeutet, an jenes Institut berufen zu werden, an dem man einst studiert und gearbeitet hat.

Sofort hat Prof. Schmid das Institut für Landesvermessung radikal modernisiert, neue Instrumente angeschafft und die Vorlesungen auf den letzten Stand der geodätischen Wissenschaft gebracht. Ein Hauptanliegen dabei war ihm die rigorose Durchsetzung des Matrizenkalküls in der Ausgleichsrechnung, was bis dahin an unserer Hochschule vernachlässigt worden war. Ein so dynamischer Mensch kann sich akademischen Ämtern nicht entziehen. Prof. Schmid war durch viele Jahre Vorsitzender der 1. und der 2. Staatsprüfungskommission bzw. der Diplomprüfungskommission, der Studienkommission und der Fachgruppe. Als schwierigstes Amt fiel ihm in den Jahren 1974 - 1977 das Dekanat der Technisch-Naturwissenschaftlichen Fakultät zu. Schwierig deshalb, weil in seine Amtszeit die Durchführung des UOG und der Übergang der Fakultät von einer reinen Professorenkurie zu einem Kollegialorgan aller Hochschulangehörigen fiel. Durch sein Geschick in der Menschenführung, durch seine feste Haltung verbunden mit Konzilianz konnte Prof. Schmid heikle Situationen meistern und anfängliche Spannungen abbauen.

Das bisher Gesagte zeigt nur einen Teil des Wesens und Wirkens von Prof. Schmid. In all diesen Jahren hat er sein Ingenieurbüro zu einem der größten seiner Art in Öster-

reich ausgebaut. Dazu gehört ein bewundernswertes Maß an Unternehmergeist, kluger Menschenführung und Managementqualitäten. Schmid hat als erster Ingenieurkonsulent in Österreich die digital gesteuerte Orthophotoherstellung angewendet. Sein Büro hat zahllose bedeutende Projekte im In- und im Ausland ausgeführt. Erwähnt seien nur: Die Bauüberwachung des Wiener Donauturmes, die Brückenvermessung der Schnellstraße über die Donau bei Deutsch-Altenburg, Vermessungen an der West- und Südautobahn, photogrammetrische Planerstellung für die Jauntalbahn, Vermessungsarbeiten für die U-Bahn und das Radstadion in Wien, Nivellement des gesamten Hochwasserdammsystems, Leitungskataster für Wien, sowie umfangreiche photogrammetrische und geodätische Arbeiten in Lambarene (Afrika) und im Iran. Diese reiche Erfahrung eines praktisch tätigen Geodäten konnte er dann in der Lehre an der Universität an die Studenten weitergeben.

Für solchen Einsatz innerhalb und außerhalb der Hochschule wurde Prof. Schmid mehrfach geehrt. Durch viele Jahre war er Vorstandsmitglied der Bundesingenieurkammer, Mitglied der Prüfungskommission für Ziviltechniker, Mitglied der Österreichischen Kommission für die Internationale Erdmessung und des Vorstandes des Österreichischen Vereins für Vermessungswesen und Photogrammetrie. Seit 1988 ist er dessen Ehrenmitglied. 1979 erhielt er das Goldene Ehrenzeichen für Verdienste um das Land Wien und 1981 erfolgte die Verleihung des Österreichischen Ehrenkreuzes für Wissenschaft und Kunst I. Klasse durch Frau Minister Firnberg. Über 40 wissenschaftliche Publikationen und zahlreiche Fachvorträge ergänzen das Bild eines erfolgreichen Universitätsprofessors.

Man würde der Persönlichkeit von Prof. Hans Schmid nicht gerecht, wollte man nur die äußeren Erfolge schildern. Da meine eigene Weltlinie die geodätische von Hans Schmid seit mehr als 40 Jahren oftmals gekreuzt hat, glaube ich, ihn auch als Mensch zu kennen. Leistung und Erfolg waren für ihn stets Lebenselixier. Er war äußerst dynamisch, sportlich und impulsiv, konnte manchmal herrisch reagieren, war aber nie nachtragend. Was ihn über ähnlich tüchtige Technokraten aber hinaushebt ist die Tatsache, daß Hans Schmid über den Geschäften nicht die höheren geistigen Dinge vergessen hat. Das zeigte sich unter anderem schon seit langem in der bereits erwähnten leidenschaftlichen Zuwendung zu den Problemen und Erkenntnissen der Astrophysik und der Kosmologie. In mehreren Vorträgen hat er versucht, seine Begeisterung dafür einer großen Zuhörerschaft mitzuteilen. Höchste Bewunderung verdient Hans Schmid für die Art, wie er eine schwere Krankheit erträgt. Erst kürzlich hat er gemeint, er hätte für seine 70 Jahre Leben 20 % Mehrwertsteuer bezahlt, nämlich 4 Jahre Kriegsdienst und 10 Jahre Krankheit. Seit 1946 verheiratet, hat er in seiner Frau Gudrun eine tatkräftige und kongeniale Partnerin gefunden. Drei äußerst wohlgeratene Kinder und acht Enkelkinder erlauben Hans Schmid auch in diesem Bereich auf ein erfülltes Leben zurückzublicken. Soweit mir bekannt, ist ihm nur ein Wunsch nicht erfüllt worden, die Nachfolge im Ordinariat durch seinen geistigen Ziehsohn Prof. Peters.

Lieber, hochgeschätzter Prof. Hans Schmid! Einem schönen akademischen Brauch folgend, wird Dir zu Deinem 70. Geburtstag ein Festkolloquium mit vier Fachvorträgen gewidmet. Die Veranstalter dieses Kolloquiums hatten den guten Gedanken, als Vortragende nicht renommierte, womöglich gleichaltrige, Kollegen zu gewinnen, die ja nur sich selbst darstellen würden, sondern sie haben dazu die jüngsten Nachwuchsforscher unserer Fachgruppe aufgefordert. Diese werden zwar auch sich selbst und ihre Arbeiten darstel-

len, ihre Arbeiten jedoch basieren zum nicht geringsten Teil auf dem, was Du ihnen gelehrt hast. Kann es Schöneres für einen Hochschullehrer geben, als zu erleben, wie seine Schüler flügge werden und die Wissenschaft weitertreiben?

Lieber und liebenswerter Hans Schmid, ich nehme mir die Freiheit, mich zum Sprecher aller Teilnehmer dieses Festkolloquiums zu machen und Dir unsere Glückwünsche zum 70. Geburtstag und unsere besondere Wertschätzung auszudrücken. Möge es Dir und auch uns, Deiner Familie, Deinen Kollegen und Freunden, vergönnt sein, Deine raumartige Weltlinie noch viele Jahre verfolgen zu können, möglichst wenig beeinträchtigt durch körperliche Fährnisse. Deine zeitartige Weltlinie aber, dessen sind wir gewiß, wird in dem unergründlichen, von Dir so geliebten Kosmos in Ewigkeit weiterlaufen.

Wien, 4. Mai 1990

Kurt Bretterbauer

ANALYSEAusGLEICHUNG ZUR INTERPRETATION

GEODÄTISCHER LAGENETZE

von

H. Stanek

Kurzfassung:

Ein Programmsystem zur umfassenden Analyse der Qualitätskriterien von geodätischen Lagenetzen wird vorgestellt. Dabei werden neben den üblichen Genauigkeits- und Zuverlässigkeitskriterien auch punktbezogene Qualitätsaussagen formuliert und in einem System zusammengefaßt.

Die schrittweise Aufdeckung "grober" Beobachtungsfehler und die Überprüfung der mathematischen Modellbildung, insbesondere der Datumsvorgabe mittels statistischer Hypothesentests bilden weitere Schwerpunkte.

Die dabei angewandte Variante der Modellbildung stellt eine universelle Bearbeitungsform geodätischer Netze zur Verfügung.

Eine, in vier Stufen gegliederte, Strategie zur Analyse hierarchischer Netze wird formuliert und auf ein synthetisches und drei reale Beispiele angewendet.

Abstract:

A programming system for the extensive analysis of quality criteria of 2D geodetic networks is presented. Apart from the common criteria of accuracy and reliability, point-related quality statements are being defined as well and integrated in one application.

The gradual disclosure of "gross" errors in observations and the investigation of mathematical model formation and especially the verification of the networks-date by means of statistical hypothesis tests represent further curcial points.

A universal processing type for geodetic networks is made available with a variant of model formation applied herewith.

A four-step strategy for analysis of hierarchic geodetic networks is being formed and applied to on synthetic and three concrete examples.

Inhaltsverzeichnis

1.	Problemanalyse, Zielvorgabe	1
2.	Mathematische Modellbildung und Parameterschätzung	3
2.1	Funktionales Modell	3
2.2	Stochastisches Modell	7
2.3	Ausgleichung nach vermittelnden Beobachtungen	9
2.4	Lagerung von geodätischen Netzen	11
3.	Möglichkeiten der Datumsvorgabe in Lagenetzen	15
3.1	Hierarchische Datumsfestsetzung	16
3.2	Dynamische Datumsvorgabe	17
3.3	Datumsvorgabe mittels Ersatzbeobachtungen	21
4.	Statistische Hypothesentests zur Modellüberprüfung	24
4.1	Grobe Fehler in den Beobachtungen	25
4.1.1	DATA SNOOPING nach BAARDA	26
4.1.2	Tau - Kriterium nach POPE	28
4.1.3	t - Test zur Grobfehlersuche	29
4.1.4	Multipler t - Test	29
4.2	Grobfehlersuche in den Ersatzbeobachtungen	31
4.2.1	Lineare eindimensionale Hypothesentests	32
4.2.2	Mehrdimensionale Hypothesentests	33
4.3	Überprüfung des stochastischen Modells	34
4.4	Varianzkomponentenuntersuchung	35
5.	Qualitätskriterien geodätischer Lagenetze	37
5.1.1	globale Genauigkeitsangaben	40
5.1.2	lokale Genauigkeitsangaben	43
5.1.3	funktionale Genauigkeitsangaben	45
5.2	Zuverlässigkeitsmaße	47
5.2.1	innere Zuverlässigkeit	49
5.2.2	äußere Zuverlässigkeit	51

5.3	Hauptkomponentenuntersuchung	53
5.4	Lokale Einzelpunktuntersuchung	55
6.	a priori Untersuchung durch Simulationsausgleichung	58
6.1	Planung von Netzverdichtungen	62
6.2	Abschätzung beliebiger Beobachtungskombinationen	63
7.	Realisierung einer umfassenden Analyseausgleichung	65
7.1	Programmablauf	70
7.2	stufenweiser Analyseablauf	74
8.	Testberechnungen	79
8.1	schematisches Netz - T2	79
8.2	Datumsüberprüfung - EP Pottendorf	106
8.3	Zuverlässigkeitsüberprüfung - B 79 Sieggraben	114
8.4	Gesamtanalyse - EP Brand Nagelberg	119
9.	Ausblicke, Weiterentwicklungen	125
	Literaturverzeichnis	127
	Anhang	133

1. Problemanalyse, Zielvorgabe

Zur Erfüllung Ihrer Vermessungsaufgaben müssen den Anwendern eine große Anzahl von flächendeckend verteilten amtlichen Festpunkten zur Verfügung gestellt werden. Dazu wurde eine Folge von hierarchischen Netzverdichtungen von der 1. Ordnung mit Visurlängen bis etwa 30 km bis zur Ordnung der EP - Felder angelegt. Die ursprünglich ausschließlich angewandte Beobachtungsform der Winkel- bzw. Richtungsmessungen und die Ausmessung einiger weniger Invardrahtbasen zur Maßstabsbestimmung bedingte eine optimale Form des Netzaufbaues in Form von Dreiecksmaschen. In diesen wurden durch weitere Beobachtungen Verdichtungen in mehreren Hierarchiestufen vorgenommen, bis schließlich EP- Netze mit Punktabständen zwischen 500 und 700 m erstellt werden konnten. Diesen Punkten gilt das Hauptinteresse der Anwender. Etwaige Genauigkeitsaussagen sind, falls sie überhaupt verfügbar sind zu optimistisch da ja innerhalb der einzelnen Hierarchiestufen die höherwertigen Festpunkte als fehlerfreie, rein deterministische "Datumspunkte" behandelt werden. In der Praxis wird die Hierarchieabfolge bei der Lagerung von Folgenetzen nicht streng durchgehalten. So können in einer Ausgleichung der 5. Ordnung Punkte der 1. bis 4. Ordnung als Datumspunkte herangezogen werden. Die Begründung dieser Vorgangsweise ist in der enormen Datenmenge zu finden. Auch die zeitliche Abfolge für die Ersterstellung und Aktualisierung von Netzteilen spricht neben der erwähnten Beobachtungs- und Auswertemethodik für hierarchische Vorgangsweise.

Mit der Verfügbarkeit von EDM- Geräten wurde kleinräumig eine Überprüfung des Netzmaßstabes möglich. Durch Einbeziehung von Streckenbeobachtungen zur Punktbestimmung wurde nach und nach die Netzform in Dreiecksmaschen zu Gunsten von Polygonzügen verändert. Die Auswirkung auf die Kontrollierbarkeit der Beobachtungen bzw. der Zuverlässigkeiten wurde dabei häufig nicht beachtet. Insbesondere bei den häufig mit modernen Geräten nachbeobachteten Netzen der 4. und 5. Ordnung und im speziellen den EP- Netzen wird dieser Effekt besonders deutlich. Die Lagerung bei der Ausgleichung dieser Netze erfolgt auf, aus älteren Operaten entstammenden "Datumspunkten". Die hohe Genauigkeit der Nachbarschaftsbeziehung der neu zu erstellenden Punkte wird durch die hierarchische Datumsfestsetzung mit nicht-stochastischen Datumspunkten beeinflusst. Die Angabe von Genauigkeitsmaßen wird damit schwer interpretierbar.

Es sollte ein Programmsystem erstellt und erprobt werden, welches einerseits dem aktuellen Stand der Ausgleichsrechnung entspricht und andererseits eine umfassende Analyse von Modellbildung und Netzqualität ermöglicht.

An der hierarchischen Datumsfestsetzung muß bei der endgültigen Auswertung festgehalten werden. Die Frage nach einer optimalen Auswahl der Festpunkte für diese Form der Datumsfestsetzung wurde als besonders dringliches Problem eingestuft.

Die Bereitstellung von Informationen zur Zuverlässigkeit von Beobachtungen und deren Auswirkungen sollte neben der umfassenden Darstellung der Genauigkeitssituation die Analyse einer Reihe von Beurteilungskriterien ermöglichen.

Eine weitere Anforderung war die Interpretation der Modellüberprüfung durch statistische Aussagen. Weiters sollten dem Bearbeiter einzelne schlecht bestimmbare Einzelpunkte oder "schwimmende" Punktgruppen, sowie deren Auswirkungen auf lokale Genauigkeitsmaße aufgezeigt werden. Ein besonderes Anliegen stellte die übersichtliche Gestaltung der Benutzeroberfläche sowie die interaktive Bearbeitungsform mit Berücksichtigung eines entsprechenden Antwortzeitverhaltens dar. Die Datenübernahme in andere Programmentwicklungen sollte in einfacher Form möglich sein.

Herrn Prof.Dipl.-Ing.Dr. H. Schmid danke ich für die Unterstützung und seinen Zuspruch zur Durchführung dieser Arbeit. Besonderen Dank habe ich auch an seine Magnifizenz Prof.Dipl.-Ing.Dr. K. Kraus für die Übernahme des Koreferates zu richten. *)

In Gesprächen mit Dipl.-Ing. Dr. E. ERKER und Dipl.-Ing. E. IMREK vom BA f.E u.Vw Abteilung K 2 konnten wesentliche Anregungen für die Formulierung der Zielsetzung dieser Arbeit gewonnen werden. Weiters konnte bei der Überprüfung der Entwicklung auf Praxistauglichkeit mit Hilfe von bereitgestellten Testoperaten zurückgegriffen werden. Für diese Kooperation möchte ich mich bei beiden Herrn an dieser Stelle herzlich bedanken.

*) Diese Arbeit ist eine approbierte Dissertation, welche an der Technisch- Naturwissenschaftlichen Fakultät der Technischen Universität Wien zur Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften eingereicht wurde.

2. Mathematische Modellbildung und Parameterschätzung

Die gegenseitige Lage von Punkten eines geodätischen Netzes wird durch Beobachtungen bestimmt. Jede dieser Beobachtungen ist auf Grund der vorgegebenen Gerätegenauigkeit und der Beobachtungsmethodik eine Beobachtungsgenauigkeit zuordenbar. Zur Schätzung der Parameter, sowie zur Überprüfung von Genauigkeiten und Zuverlässigkeiten werden zusätzliche (überschüssige) Beobachtungen durchgeführt. Es stehen damit n Beobachtungen u gesuchten Größen gegenüber. Die sich auf Grund zufälliger Meßfehler ergebenden Widersprüche werden nach der Methode der kleinsten Quadrate derart verteilt, daß plausibelste und eindeutige Werte für die Beobachtungen und den daraus abgeleiteten Unbekannten folgen. Die Grundlage dieser Parameterschätzung ist die Erstellung eines mathematischen Modelles. Dieses muß die geometrischen und wahrscheinlichkeitstheoretischen Gegebenheiten in seinen Zusammenhängen in vollständiger Weise beschreiben. An Beobachtungen treten dabei Richtungen, Strecken, beobachtete Orientierungen und später zu definierende Ersatzbeobachtungen auf. Die u unbekannt Parameter sind dabei Punktkoordinaten, Orientierungsunbekannte bei Richtungsbeobachtungen in Satzform und eventuell eine Maßstabsunbekannte der Streckenbeobachtungen.

2.1 Funktionales Modell

In diesem Modellanteil wird die funktionale Relation zwischen den Beobachtungen und den unbekannt Parametern formuliert.

Die n Beobachtungen werden in einem Beobachtungsvektor zusammengefaßt:

$$L^T = (L_1, L_2, L_3, \dots, L_n) \quad (2-1)$$

Der Beobachtungsvektor (2-1) besteht komponentenweise aus der Realisierung von Zufallsgrößen. Sein Erwartungswert

$$E(L) = \Sigma \quad (2-2)$$

ist durch den Vektor der "wahren Werte" bestimmt. Implizit sind damit systematische und grobe Beobachtungsfehler ausgeschlossen. Die u Parameter X_i bilden den Unbekannten-Vektor \mathbf{X} :

$$\mathbf{X}^T = (X_1, X_2, X_3, \dots, X_u) \quad (2-3)$$

$$n > u$$

$$f = n - u \quad \dots \quad \text{Anzahl der Freiheitsgrade}$$

(Anzahl der überschüssigen Beobachtungen)

Die mathematische Beziehung zwischen dem Beobachtungsvektor und dem gesuchten Parametervektor wird als das nichtlineare funktionale Modell bezeichnet:

$$\mathbf{L} = f(\mathbf{X}) \quad (2-4)$$

\mathbf{X} = Vektor der wahren Werte der Unbekannten

Der Übergang von den wahren Werten \mathbf{X} zu den Schätzwerten $\hat{\mathbf{X}}$ gelingt durch Erweiterung des Beobachtungsvektors um den Vektor der Verbesserungen beziehungsweise Residuen. Die Form (2-4) wird damit zu

$$\hat{\mathbf{l}} = \mathbf{L} + \mathbf{v} = f(\hat{\mathbf{X}}) \quad (2-5)$$

Dieses System der nichtlinearen Verbesserungsgleichungen wird nunmehr zum Zwecke der Behandlung nach Verfahren der linearen Algebra in ein überbestimmtes lineares Gleichungssystem übergeführt. Dazu wird in $\hat{\mathbf{X}}$ ein Anteil wegen Näherungskoordinaten \mathbf{X}^0 abgespalten:

$$\hat{\mathbf{X}} = \mathbf{X}^0 + \mathbf{x} \quad (2-6)$$

Die Beziehung (2-6) wird in die Form (2-4) eingesetzt und damit auch der Beobachtungsvektor zerlegt:

$$\begin{aligned} \mathbf{L}^0 &= f(\mathbf{X}^0) \\ \mathbf{L} &= \mathbf{L}^0 + \mathbf{l} \end{aligned} \quad (2-7)$$

Jede einzelne nichtlineare Beobachtungsgleichung wird nun nach TAYLOR in eine Potenzreihe entwickelt. Unter der Voraussetzung, daß $\mathbf{x} \ll \mathbf{x}^0$ gilt, kann man sich dabei auf Glieder erster Ordnung beschränken:

$$\mathbf{f}^0 = f(\mathbf{x}^0) + (\delta f / \delta \mathbf{x})_0 (\mathbf{x} - \mathbf{x}^0) + \dots \quad (2-8)$$

Mit (2-6) und (2-7) ergibt sich das linearisierte funktionale Modell:

$$\mathbf{f} = \mathbf{A} \mathbf{x} \quad (2-9)$$

Die Elemente der Matrix A bestehen aus den Differentialquotienten aus (2-8):

$$\mathbf{A} = \left[\frac{\delta f(\mathbf{x})}{\delta \mathbf{x}} \right]_0 = \begin{bmatrix} \left[\frac{\delta f_1(\mathbf{x})}{\delta \mathbf{x}} \right]_0 & \left[\frac{\delta f_1(\mathbf{x})}{\delta x_2} \right]_0 & \dots & \left[\frac{\delta f_1(\mathbf{x})}{\delta x_u} \right]_0 \\ \left[\frac{\delta f_2(\mathbf{x})}{\delta x_1} \right]_0 & \left[\frac{\delta f_2(\mathbf{x})}{\delta x_2} \right]_0 & \dots & \left[\frac{\delta f_2(\mathbf{x})}{\delta x_u} \right]_0 \\ \vdots & \vdots & \ddots & \vdots \\ \left[\frac{\delta f_n(\mathbf{x})}{\delta x_1} \right]_0 & \left[\frac{\delta f_n(\mathbf{x})}{\delta x_2} \right]_0 & \dots & \left[\frac{\delta f_n(\mathbf{x})}{\delta x_u} \right]_0 \end{bmatrix} \quad (2-10)$$

Die Matrix A wird als Designmatrix bzw. Formmatrix mit n Zeilen und u Spalten bezeichnet. Die Form (2-9) beschreibt das GAUSS-MARKOV - Modell. Nunmehr sollen die Gleichungen gemäß (2-5) linearisiert werden. Das System der linearisierten Verbesserungsgleichungen läßt sich dann wie folgt angeben:

$$\mathbf{f} = \mathbf{l} + \mathbf{v} = \mathbf{A} \mathbf{x} \quad (2-11)$$

Die nachfolgenden Tabelle zeigt eine Zusammenstellung der Koeffizienten der Designmatrix A gegliedert nach Beobachtungsarten für Lagenetze:

$$\begin{aligned}
 dx &= x_{z^0} - x_{s^0} & y_{s^0}, x_{s^0} & \dots \text{Näherungskoord. Standpkt.} \\
 dy &= y_{z^0} - y_{s^0} & y_{z^0}, x_{z^0} & \dots \text{Näherungskoord. Zielpkt.} \\
 e & \dots 200 / \pi * (1000 / s[m]) & & s^2 = dx^2 + dy^2
 \end{aligned}$$

Beob. art	x_z	y_z	o_s	m
Richtung	$-e * dy / s$	$e * dx / s$	-1	0
Strecke	dx / s	dy / s	0	$-s/1000$
Orientierung	0	0	-1	0
x-Koord.diff.	1	0	0	0
y-Koord.diff.	0	1	0	0

Tab. 2.1

Die Koeffizienten in den Spalten x_z und y_z beziehen sich jeweils auf die Komponenten der Koordinaten des Zielpunktes und die Komponente o_s auf den Standpunkt der Beobachtung. Die Koeffizienten der Komponenten der Standpunktskoordinaten berechnen sich gleichartig nach Multiplikation mit dem Faktor (-1).

Die Beobachtungsarten beinhalten neben den in Lagenetzen üblichen Formen der Richtungs- und Streckenbeobachtung auch die Koeffizienten für Beobachtungsgleichungen von beobachteten Orientierungen und Koordinatendifferenzen. Die letzten beiden Formen von Ersatzbeobachtungen werden im Abschnitt 3.3 benötigt. Gemessene Orientierungen werden etwa für die Einbeziehung von Kreiselazimuten bei der Ausgleichung von Tunnelnetzen benötigt.

In einer Verbesserungsgleichung und damit in einer Zeile der Matrix A ergeben sich also maximal fünf Koeffizienten ungleich Null. Dieser Umstand wird bei der Erstellung, Speicherung und Weiterverarbeitung der Designmatrix berücksichtigt.

Die Komponenten des Unbekanntenvektors \mathfrak{X} bestehen zum einen aus den Koordinatenunbekannten der gesuchten Neupunkte des Netzes und zum anderen aus den Orientierungsunbekannten der Satzbeobachtungen und eventuell einer Maßstabsunbekannten. Das funktionale Modell beschreibt die Zusammenhänge zwischen Beobachtungen und den Unbekannten in deterministischer Weise. Informationen über die Beobachtungsgenauigkeiten der einzelnen Beobachtungen werden durch diesen Modellteil nicht beschrieben.

2.2 Stochastisches Modell

Dieser Modellanteil setzt die Bedeutung der einzelnen Beobachtungen zueinander fest. Eine einfache Vorstellung dabei ist die Einführung sogenannter Wiederholungszahlen zu jeder Beobachtung. Werden dabei auch nicht ganzzahlige Werte zugelassen, spricht man im speziellen von der Vergabe von Gewichten. Nunmehr sind Beobachtungen durch einen Meßfehler charakterisiert. Diese Meßfehler können entweder aus Herstellerangabe bekannt sein oder aus Mehrfachbestimmungen der Beobachtungswerte geschätzt werden. In Sonderfällen ist auch eine direkte Vorgabe von Gewichten denkbar, etwa wenn diese auf Grund von Vergleichsoperaten übernommen werden soll (BA f.E u.Vw). Das gesamte stochastische Verhalten des Beobachtungsvektors L läßt sich in einer Varianz-Kovarianz-Matrix Σ_{LL} beschreiben.

Die Abweichungen zwischen dem Beobachtungsvektor L und den entsprechenden wahren Werten \mathbf{L} bezeichnen wir als Abweichungsvektor \mathbf{v}_L . Die Komponenten dieses Abweichungsvektors stellen die wahren Fehler ϵ dar. Die stochastischen Eigenschaften des Abweichungsvektors beschreiben dessen Kovarianzmatrix Σ_{LL} .

$$\Sigma_{LL} = E(\epsilon\epsilon^T) = \begin{bmatrix} \sigma_1^2 & r_{12} \sigma_1 \sigma_2 & \dots & r_{1n} \sigma_1 \sigma_n \\ r_{21} \sigma_2 \sigma_1 & \sigma_2^2 & \dots & r_{2n} \sigma_2 \sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} \sigma_n \sigma_1 & r_{n2} \sigma_n \sigma_2 & \dots & \sigma_n^2 \end{bmatrix} \quad (2-12)$$

σ_j : Standardabweichung der Beobachtung L_j , $j=1,2,\dots,n$,
 r_{jk} : Korrelationskoeffizient zwischen L_j und L_k , $j < k$.

Die Elemente der obigen Matrix lassen sich nur bis auf eine Multiplikationskonstante angeben (σ_0^2):

$$\Sigma_{LL} = \sigma_0^2 Q_{LL} \quad Q_{LL} = \frac{1}{\sigma_0^2} \Sigma_{LL} \quad (2-13)$$

Diese, bis auf die Konstante bestimmbare Matrix Q_{LL} beinhaltet in den Hauptdiagonalen die jeweiligen Beobachtungsvarianzen und in den restlichen Gliedern die Kovarianzen nach Multiplikation mit σ_0^2 . Sie

stellt eine symmetrische, in den hier besprochenen Fällen stets positiv definite und damit invertierbare Matrix der Dimension (n,n) dar. Es existiert also eine symmetrische Kehrmatrix von gleicher Dimension.

$$P = Q_{LL}^{-1} \quad (2-14)$$

Die Matrix P beschreibt die vollständige Gewichtssituation der Beobachtungen.

$$P = \begin{bmatrix} p_{11} & r_{12} \sigma_1 \sigma_2 & \dots & r_{1n} \sigma_1 \sigma_n \\ r_{21} \sigma_2 \sigma_1 & p_{22} & \dots & r_{2n} \sigma_2 \sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} \sigma_n \sigma_1 & r_{n2} \sigma_n \sigma_2 & \dots & p_{nn} \end{bmatrix} \quad (2-15)$$

Im häufig angenommenen Sonderfall unabhängiger Beobachtungen wird die Q_{LL} - Matrix zur Diagonalmatrix und es ergibt sich damit die P - Matrix mit den Diagonalelementen:

$$p_{jj} = \frac{\sigma_0^2}{\sigma_j^2} \quad j = 1, 2, 3, \dots, n \quad (2-16)$$

Die Matrix Σ_{LL} beschreibt das stochastische Modell und ist ident mit der entsprechenden Varianz- Kovarianzmatrix Σ_{11} des gekürzten Beobachtungsvektors l. Die Gewichtsrelationen der einzelnen Beobachtungen zueinander sind in der Gewichtsmatrix (2-14) festgelegt.

Alternativ zur Bestimmung der Beobachtungsgewichte nach (2-16) erfolgt die Bestimmung bei Ausgleichungen des BA f.E u.Vw durch direkte Angabe von Gewichten. Dabei wird das Gewicht von Streckenbeobachtungen $p_s = 3/m_s^2$ ($m_s = \pm 2\text{cm}$) und das der Richtungsbeobachtungen entfernungsproportional bei Visurweiten ≤ 2 km mit $p_R = s[\text{km}]/m_R^2$ und Visurweiten > 2 km $p_R = (3 - 2/s[\text{km}])/m_R^2$ ($m_R = \pm 10''$) gewählt. In der Programmentwicklung wurde auch diese Gewichts Vorgabe berücksichtigt.

Während es im allgemeinen gelingt, die Gewichtsverhältnisse innerhalb einer Beobachtungsgruppe genau genug anzugeben, ist die Annahme der Genauigkeitsrelation zwischen verschiedenen Gruppen von Beobachtungsarten fehleranfällig. Durch Überprüfung des Verhältnisses mit Hilfe der Varianzkomponentenuntersuchung (Abschnitt 4.4) kann das stochastische Modell in dieser Hinsicht überprüft werden.

2.3 Ausgleichung nach vermittelnden Beobachtungen

Durch die Vorgabe des linearen funktionalen Modells (2-11) und des stochastischen Modells (2-12) ist die Modellbildung eines Ausgleichungsproblems vollständig beschrieben. Das überbestimmte lineare Gleichungssystem wird nunmehr durch Vorgabe einer Nebenbedingung in eindeutiger Weise nach der Methode der kleinsten Quadrate durch Minimierung der Zielfunktion $v^T Q_{LL}^{-1} v = \text{Minimum}$ aufgelöst. Dies führt zu dem Normalgleichungssystem (Koch, 1980)

$$A^T P A \hat{x} - A^T P l = 0 \quad (2-17)$$

$$N = (A^T P A)$$

Die Matrix N wird als Normalgleichungsmatrix bezeichnet. Ihre Inverse existiert, wenn u unabhängige die Form des Netzes beschreibende Unbekannte ausgewählt wurden und kein Konfigurationsdefekt vorliegt. Die Schätzwerte für die Zuschläge zu den Näherungen für die Unbekannte berechnen sich zu:

$$\hat{x} = Q A^T P l \quad (2-18)$$

$$Q = (A^T P A)^{-1} = N^{-1}$$

$$\Sigma_{\hat{x}\hat{x}} = s_0^2 Q$$

Die Kovarianzmatrix $\Sigma_{\hat{x}\hat{x}}$ beschreibt die Genauigkeitseigenschaften der berechneten Parameter.

Aus der Ausgleichung erhält man den Schätzwert für den a posteriori Varianzfaktor bzw. die Standardabweichung der Gewichtseinheit a posteriori:

$$s_o^2 = \frac{v^T P v}{n - u} \quad (2-19)$$

Dieser Wert stellt nach (Koch, 1987) einen erwartungstreuen Schätzwert für σ_o^2 dar. Aus der a posteriori Varianz Kovarianz - Matrix $\Sigma_{xx} = s_o^2 Q$ kann die Genauigkeit der Unbekannten und Funktionen der Unbekannten berechnet werden. Insbesondere stellen ja die Beobachtungen (nach der Ausgleichung) l bzw. \hat{l} Funktionen der Unbekannten dar, deren Fehler nach der Ausgleichung im weiteren benötigt werden. Für einen beliebigen Zufallsvektor h , der als lineare Funktion des (gekürzten) Beobachtungsvektors l dargestellt werden kann (Pelzer, 1985).

$$h = H l \quad (2-20)$$

Mit einer gegebenen Matrix H enthält die Kovarianzmatrix alle stochastischen Informationen. Sie läßt sich mit Hilfe des Varianzfortpflanzungsgesetzes angeben:

$$\Sigma_{hh} = H \Sigma_{ll} H^T = \sigma_o^2 H Q_{ll} H^T = \sigma_o^2 Q_{hh} \quad (2-21)$$

Die Kofaktorenmatrix Q_{hh} des Funktionsvektors h lautet

$$Q_{hh} = H Q_{ll} H^T \quad (2-22)$$

Fassen wir den Vektor h nunmehr in den Komponenten \hat{x} (Lösungsvektor), \hat{l} (gekürzter Beobachtungsvektor) nach (2-11), v (Verbesserungen) und \hat{y} (= beliebige linearisierte Funktionen der Unbekannten \hat{x}), kann man die Gesamtkofaktorenmatrix Q_{hh} angeben:

$$h = \begin{bmatrix} \hat{x} \\ \hat{l} \\ v \\ \hat{y} \end{bmatrix} = \begin{bmatrix} Q A^T P \\ A Q A^T P \\ A Q A^T P - E \\ Q A^T P \end{bmatrix} l \quad (2-23)$$

$$v = A \hat{x} - l = A Q A^T P l - l$$

$$\hat{y} = F \hat{x} = F Q A^T P l$$

$$Q_{hh} = \begin{bmatrix} Q_{xx} & Q_{xz} & Q_{xv} & Q_{xy} \\ Q_{zx} & Q_{zz} & Q_{zv} & Q_{zy} \\ Q_{vx} & Q_{vz} & Q_{vv} & Q_{vy} \\ Q_{yx} & Q_{yz} & Q_{yv} & Q_{yy} \end{bmatrix} = \begin{bmatrix} Q & Q A^T & 0 & Q F^T \\ A Q & A Q A^T & 0 & A Q F^T \\ 0 & 0 & Q_{111} - A Q A^T & 0 \\ F Q & F Q A^T & 0 & F Q F^T \end{bmatrix}$$

Bemerkenswert ist das Verschwinden der Anteile Q_{xv} und Q_{zv} . Sie zeigt die stochastische Unabhängigkeit des Verbesserungsvektors v von den übrigen Ausgleichungsergebnissen.

2.4 Lagerung von geodätischen Netzen

Geodätische Beobachtungen wie Richtungen, Strecken, gemessene Orientierungen vermitteln nur die Beziehung zu Koordinatenunterschieden, während der absolute Koordinatenbezug, das sogenannte geodätische Datum, undefiniert bleibt. Die Designmatrix A besitzt bei Einführung von Koordinaten als Unbekannte den Spaltendefekt d . Dieser Spaltendefekt d bewirkt in der Normalgleichungsmatrix ein Rangdefizit von eben dieser Größe. Auf Grund dieses Rangdefizites ist eine Inversion der Normalgleichungsmatrix gemäß (2-18) nicht möglich. Es sind d Parameter zur Festlegung des Datums bzw. zur Lagerung, Orientierung und Maßstabsgebung des geodätischen Netzes vorzugeben. Im Falle eines Triangulationsnetzes sind $d = 4$ Bedingungen zur Beseitigung des Datumdefektes vorzugeben. Bei Trilaterationsnetzen und Kombinationen aus Triangulationen und Trilaterationen ist die Anzahl der einzuführenden Restriktionen abhängig von der Maßstabsgebung im Netz. Bei Ableitung der Netzausdehnung aus den Streckenbeobachtungen selbst, sind $d = 3$ Bedingungen notwendig. Soll eine Maßstabsanpassung zwischen den Streckenmessungen aus dem Festpunktfeld erfolgen, so ist über $d = 4$ Parameter zu verfügen. Werden d Koordinaten vorgegeben, bleiben datumsinvariante Größen wie die Standardabweichungen der Gewichtseinheit oder die Netzgeometrie ohne

den Einfluß von Zwängen frei und man spricht von zwangsfreien Netzen. Allerdings sind der Koordinatenvektor \mathbf{x} und dessen Kovarianzmatrix von der Wahl der festgehaltenen Koordinaten abhängig. Diesen Mangel behebt (Meissl, 1969) durch Übergang auf die "innere Fehlertheorie". Dabei wird der Rangdefekt durch folgende Bedingung behoben: Die Spur der sich ergebenden Kofaktorenmatrix der Unbekannten Q_{xx} bzw. der Kovarianzmatrix Σ_{xx} soll zum Minimum werden. Es ergeben sich damit die kleinstmöglichen Fehlermaße für die Unbekannten.

$$\text{spur} (Q_{xx}) \dots \text{Minimum} \quad (2-24)$$

Die Erfüllung dieser Forderung resultiert aus der Minimumsbedingung:

$$\mathbf{x}^T \mathbf{x} \dots \text{Minimum} \quad (2-25)$$

Für den berechneten Parametervektor gilt damit, daß seine euklidische Norm zum Minimum wird. Zur Auflösung wird das Normalgleichungssystem (2-17) um vier linearisierte Bedingungsgleichungen (Restriktionen) erweitert:

$$\begin{bmatrix} N & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} = \begin{bmatrix} A^T P l \\ 0 \end{bmatrix} \quad (2-26)$$

Die Inverse der erweiterten Normalgleichungsmatrix läßt sich formal wie folgt angeben:

$$\begin{bmatrix} N & G \\ G^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} Q & G(G^T G)^{-1} \\ (G^T G)^{-1} G^T & 0 \end{bmatrix}^{-1} \quad (2-27)$$

Man spricht dabei auch von einer Ränderung der Matrix N mit einer Matrix G . Für den Maximalrangdefekt $d = 4$ in einem Lagenetz mit p Neupunkten hat die Matrix G folgende Gestalt:

$$G^T = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 \\ -y_1^0 & x_1^0 & -y_2^0 & x_2^0 & \dots & -y_p^0 & x_p^0 \\ x_1^0 & y_1^0 & x_2^0 & y_2^0 & \dots & x_p^0 & y_p^0 \end{bmatrix} \begin{array}{l} \text{Transl. in } x \\ \text{Transl. in } y \\ \text{Rotation} \\ \text{Maßstab} \end{array} \quad (2-28)$$

Soll der Maßstab aus Streckenbeobachtungen beibehalten werden, entfällt für $d = 3$ die letzte Bedingungsgleichung. Wird die Matrix G in der Form (2-28) benutzt, so ist Q die sogenannte Pseudoinverse von N mit minimaler Spur:

$$Q = N^+ \quad (2-29)$$

Setzt man die Pseudoinverse Q in (2-18) ein, ergibt sich der Lösungsvektor gemäß der Forderung (2-25). Die Theorie der Pseudoinversen wurde von Rao und Mitra (1971) aus mathematischer Sicht zur direkten Auflösung formuliert. Bjerhammer (1958) gibt eine spezielle Normalinverse an, durch die ein Gleichungssystem mit liegender Koeffizientenmatrix unter der Nebenbedingung (2-25) unmittelbar aufgelöst werden kann:

$$\begin{matrix} C & \hat{x} & = & l \\ m,u & u,1 & & m,1 \end{matrix} \quad m < u \quad (2-30)$$

C entsteht aus einer quadratischen N - Matrix durch Streichen von d beliebigen Zeilen. Der eindeutige Lösungsvektor \hat{x} ergibt sich:

$$\hat{x} = C^T (C C^T)^{-1} l \quad (2-31)$$

unter der Bedingung (2-25). Der Matrizenausdruck $C^T (C C^T)^{-1}$ in (2-31) stellt keine Pseudoinverse dar. Die Form ist von der Auswahl der d gestrichenen Anteile abhängig. Die Pseudoinverse Q^+ ergibt sich als stochastische Ringinverse:

$$Q^+ = C^T (C C^T)^{-1} C^T (C C^T)^{-1} C \quad (2-32)$$

Diese Matrix ist symmetrisch, eindeutig und erfüllt die Bedingung (2-24) nach minimaler Spur. Erfüllt ein Lösungsvektor die Bedingung (2-25), spricht man von einer freien Netzausgleichung. Alle Punkte nehmen an der Festlegung des geodätischen Datums des Lagenetzes teil. Soll lediglich eine Untermenge der behandelten Neupunkte zur Datumsfestsetzung herangezogen werden, spricht man von einer Teilspurminimierung. Die Bedingungen (2-24) und (2-25) gelten dann nur anteilig für diese Neupunktsunbekannten.

Wird das Rangdefizit durch Vorgabe von genau d Koordinatenunbekannten festgesetzt, spricht man von einer zwangsfreien Ausgleichung. Zwischen diesen drei Datumsfestlegungen besteht die Möglichkeit einer S - (similarity) Transformation. Diese Transformation leistet den Übergang von einer Datumsfestlegung zu einer anderen. Transformiert werden dabei der berechnete Parametervektor und dessen Kofaktorenmatrix.

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{S}_1 \mathbf{x}_k \\ \mathbf{Q}_{11} &= \mathbf{S}_1 \mathbf{Q}_{kk} \mathbf{S}_1^T \end{aligned} \quad (2-33)$$

Diese Transformation geht auf Baarda (1973) zurück. Die (u,u) Transformationsmatrix S kann aus der Ränderungsmatrix G (2-28) angegeben werden:

$$\mathbf{S} = \mathbf{I} - \mathbf{G} (\mathbf{G}^T \mathbf{E}_1 \mathbf{G})^{-1} \mathbf{G}^T \mathbf{E}_1 \quad (2-34)$$

\mathbf{I} . . . Einheitsmatrix (u,u)

\mathbf{E}_1 . . . Diagonalmatrix (u,u) die an den Stellen der Koordinatenunbekannten, die der Datumsfestlegung dienen, 1 enthält.

Durch eine S - Transformation wird die innere Geometrie eines Netzes nicht gestört. Lediglich die Ausgleichung unter Zwang kann nicht in ein S - System einbezogen werden. Diese Datumsfestlegung liegt vor, wenn mehr als d Koordinatenunbekannte (= Restriktionen bzw. Bedingungen) festgesetzt wurden. Dabei werden Festpunktzwänge auf die Beobachtungen übertragen. Die innere Geometrie des Netzes wird durch den Festpunktzwang gestört.

3. Möglichkeiten der Datumsvorgabe in Lagenetzen

Wie in Abschnitt 2 beschrieben, stellen Beobachtungen Relationen zwischen den gesuchten Parametern dar. Durch Einbeziehung von Festpunkten wird das Lagenetz in Beziehung zum umgebenden Festpunktrahmen gesetzt. Dabei stellen die Festpunktkoordinaten korrelierte Ergebnisse einer oder mehrerer vorangegangener Ausgleichungen dar. Unter der "Datumsfestlegung" versteht man die Beseitigung des in Abschnitt 2.4 angegebenen Datumdefektes d durch Einbeziehung von Festpunkten. Dabei bedeuten für ein Lagenetz die Vorgabe eines Festpunktes die Festlegung von 2 Bedingungsgleichungen. Werden genau d Festpunktunbekannte vorgegeben, haben wir gemäß Abschnitt 2.4 eine freie Datumsfestlegung durchgeführt. Werden mehr als d Festpunktkoordinaten vorgegeben, spricht man von einer gezwängten Ausgleichung mit quasi überbestimmter Datumsvorgabe. In "freien" Netzen wird die Datumsvorgabe durch innere Bedingungen (2-28) festgelegt. Ein Bezug zu den äußeren Koordinatenrahmen ist dann nicht erfolgt.

Im Bereich der Zielrichtung dieser Arbeit ist zumeist eine gezwängte Datumsvorgabe vorgesehen. Der bisher zumeist eingeschlagene Lösungsweg wird als Ausgleichung mit "hierarchischer" Datumsfestlegung bezeichnet. Dabei werden die datumsbestimmenden Festpunktkoordinaten als unveränderliche und fehlerfreie Größen einbezogen. Etwaige "Spannungen" werden innerhalb des Festpunktkoordinatenrahmens auf die gesuchten Ausgleichungsergebnisse übertragen. Die Vernachlässigung der stochastischen Informationen des Festpunktfeldes bewirkt zu optimistische Genauigkeitsaussagen der Ausgleichungsparameter.

Alternativ dazu werden bei der "dynamischen" Datumsvorgabe die Festpunktkoordinaten als veränderliche und stochastische Größen in die Ausgleichung einbezogen. Die Festpunktkoordinaten als Unbekannte der Netzausgleichung der höheren Ordnung y resultieren daraus als korrelierte Größen mit der Kofaktorenmatrix Q_{yy} . In den wenigsten Fällen liegt die Kofaktorenmatrix Q_{yy} zur Berücksichtigung in der nächsten Verdichtungsstufe vor. Insbesondere könnten die Festpunktkoordinaten aus verschiedenen Vorausgleichungen entstammen. Als Näherung muß daher eine idealisierte Ersatzmatrix K_{yy} , eine sogenannte

Kriteriummatrix verwendet werden. In der Ersatzkovarianzmatrix muß nur die Stochastizität dieser Punktgruppe berücksichtigt werden, wobei nur diese als veränderlich gelten sollen. In der Literatur wird diese Ersatzkovarianzmatrix durch verschiedene Modelle angenähert.

3.1 Hierachische Datumsfestsetzung

Für den "üblichen" Fall der Netzverdichtung wird also der Vektor mit den Komponenten der Festpunktskoordinaten \hat{y} wie folgt angenommen:

$$\begin{aligned} \hat{y} &= \mathbf{Y} \quad \text{bzw.} \quad \hat{y} = 0 \\ Q_{\hat{y}\hat{y}} &= 0 \end{aligned} \quad (3-1)$$

Der Zuschlagsvektor $\hat{y} = 0$ stellt eine Gruppe von Bedingungsgleichungen dar. Diese werden im System der linearen Verbesserungsgleichungen (2-11) durch Vorweglimination (Streichen der entsprechenden Spalten in der A - Matrix) behandelt. Formal läßt sich diese Vorgangsweise wie folgt beschreiben:

$$\begin{bmatrix} A_1^T P_{11} A_1 & A_1 P_{11} A_2 \\ A_2^T P_{11} A_1 & A_2 P_{11} A_2 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} - \begin{bmatrix} A_1^T P_{11} l \\ A_2^T P_{11} l \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3-2)$$

Mit \hat{x} werden die unbekanntenen Koordinaten der zu bestimmenden Neupunkte bezeichnet.

Mit \hat{y} die unbekanntenen Koordinaten der Anschlußpunkte.

Damit läßt sich das Normalgleichungssystem (2-17) formal wie folgt aufspalten:

$$\begin{aligned} l + v &= A_1 \hat{x} + A_2 \hat{y} \\ 0 &= 0 \hat{x} + I \hat{y} = B \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \end{aligned} \quad (3-3)$$

Dieses gemischte Modell wird nach (Koch, 1987) als GAUSS-MARKOV - Modell mit Restriktionen bezeichnet. Dabei müssen mindestens d Unbekannte als nichtstochastische Größen vorgegeben sein.

Das zugehörige Formalgleichungssystem lautet:

$$\begin{bmatrix} A_1^T P_{11} A_1 & A_1 P_{11} A_2 & 0 \\ A_2^T P_{11} A_1 & A_2 P_{11} A_2 & I \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{k} \end{bmatrix} - \begin{bmatrix} A_1^T P_{11} l \\ A_2^T P_{11} l \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3-4)$$

\hat{k} bezeichnet den Vektor der Korrelaten. Die formale Auflösung des Systems (3.4) für den interessierenden Vektor \hat{x} lautet nach der direkten Berücksichtigung der Auswirkung der Restriktionen:

$$\begin{aligned} I \hat{y} &= 0 \\ \hat{x} &= (A_1^T P A)^{-1} A_1^T P_{11} l \end{aligned} \quad (3-5)$$

(3-5) steht mit (2-18) in Übereinstimmung. Zu beachten ist, daß s_0^2 (2-19) in der Verbesserungsquadratsumme sowohl Anteile aus den zufälligen Fehlern der Beobachtungen als auch aus den durch die Bedingungen verfälschten Beobachtungen Unsicherheiten enthält und damit schwer interpretierbar wird. Auf Grund veränderter Beobachtungsmethodiken, insbesondere den verstärkten Einsatz von EDM - Geräten und der damit verbundenen Möglichkeit der kleinräumigen Netzmaßstabsüberprüfung, muß die Form der hierarchischen Datumsvorgabe in Frage gestellt werden. Mehrere Autoren (Peters 1971, Augath 1985) haben gezeigt, daß die Vorgangsweise, vom kleinräumigen Netz das übergeordnete Netz aufzubauen, bessere Koordinatengenauigkeiten und vor allem Nachbarschaftsgenauigkeiten erwarten läßt. Schädlich könnte auch der Einfluß von aus verschiedenen Vorausgleichungen entstammenden Festpunkten oder Festpunktgruppen auf die Netzverdichtung sein. Ein Beispiel zu dieser Problematik wird in Abschnitt 8.2, EP - Pottendorf, angegeben.

3.2 Dynamische Datumsvorgabe

Dabei wird die ursprüngliche Ausgleichung mit dem nachfolgenden Verdichtungsnetz zu einem Gesamtausgleich kombiniert.

Der Unbekanntenvektor wird formal in drei Gruppen aufgeteilt:

- x : die zu bestimmenden Neupunktkoordinaten
- y_1 : die durch Verdichtungsmessungen mit den
Neupunkten verknüpften Anschlußpunktunbekannten
- y_2 : die nur von Beobachtungen l_1 im übergeordneten Netz
betroffenen Anschlußpunktunbekannten.

Die Beobachtungen l_1 mit der zugehörigen Gewichtsmatrix P_{11} betreffen die Netzbeobachtungen der ursprünglichen Ausgleichung. Die Verdichtungsmessungen verknüpfen die Neupunkte des Verdichtungsnetzes untereinander. Die Gewichtsmatrix P_{22} beschreibt die Gewichtssituation der zusätzlichen Beobachtungen l_2 . Zwischen den Beobachtungsgruppen l_1 und l_2 kann Korrelation ausgeschlossen werden. Die Gewichtsmatrix und damit das stochastische Modell lautet

$$P = \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix} \quad (3-6)$$

Die linearen Verbesserungsgleichungen lassen sich folgendermaßen beschreiben:

$$\begin{aligned} l_1 + v_1 &= A_4 y_2 + A_3 y_1 + 0 x \\ l_2 + v_2 &= 0 y_2 + A_2 y_1 + A_1 x \end{aligned} \quad (3-7)$$

Das zusammengefaßte Normalgleichungssystem zur Ausgleichung mit dynamischer Datumsvorgabe lautet:

$$\begin{bmatrix} A_4^T P_{11} A_4 & A_4^T P_{11} A_3 & 0 \\ A_3^T P_{11} A_4 & A_1^T P_{11} A_3 + A_2^T P_{22} A_2 & A_2^T P_{22} A_1 \\ 0 & A_1^T P_{22} A_2 & A_1^T P_{11} A_1 \end{bmatrix} \begin{bmatrix} y_2 \\ y_1 \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} A_4^T P_{11} l_1 \\ A_3^T P_{11} l_1 + A_2^T P_{22} l_2 \\ A_1^T P_{22} l_2 \end{bmatrix} \quad (3-8)$$

Unter der Bedingung der Unabhängigkeit der Beobachtungsgruppen l_1 und l_2 kann dieses System nach Bähr (1973) unter Vermeidung einer Totalinversion durch Anwendung einer Reihenentwicklung (Neumann'sche Reihe) aufgelöst werden. Das Normalgleichungssystem der hierarchischen Netzausgleichung stellt einen Sonderfall des Systems (3-8) dar. In Bill (1984) wird ein nach Meissl generiertes Netz mit hierarchischer und mit dynamischer Datumsvorgabe gegenübergestellt. Dabei wird eine Netzhierarchie in zwei Ordnungen angenommen.

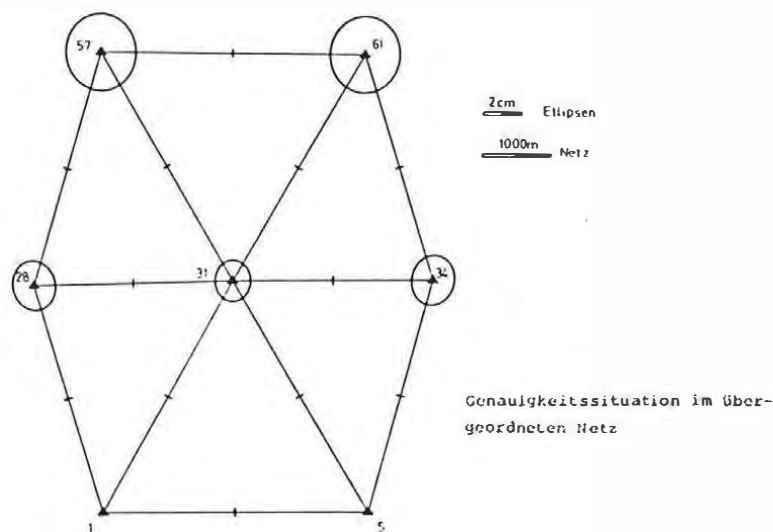


Abb. 3.1

Das Netz höherer Ordnung besteht aus 7 Punkten, das Verdichtungsnetz aus 55 weiteren Neupunkten. Tabelle 3.1 zeigt für das übergeordnete Netz charakteristische Qualitätskriterien (Genauigkeit und Zuverlässigkeit):

	Genauigkeit	Zuverlässigkeit
Richtungen	$0.36 < \bar{\sigma}_1 < 0.55$	$1.9 < \bar{\sigma}_1 < 3.8$
Strecken	$0.50 < \bar{\sigma}_1 < 0.75$	$2.4 < \bar{\sigma}_1 < 3.8$
Genauigkeit (datumsabhängig)		
	Fehlerellipse	Punktfehler
	$\sigma_{11} < \sigma_{22} < \sigma_{33}$	$\sigma_{11} < \sigma_{22} < \sigma_{33}$

Übergeordnetes Netz an zwei Punkte angeschlossen mit Maßstabsunbekannte ($\sigma^2 = 0.001$, $d_1 = 2.1 \cdot 10^{-6}$, $d = 4$)

Tab. 3.1

In Tabelle 3.2 werden für das Verdichtungsnetz Genauigkeits- und Zuverlässigkeitsmaße der Auswertung nach hierarchischer und dynamischer

Datumsvorgabe gegenübergestellt:

	0.001, $\sigma_0 = 0.001$					re $\overline{\sigma_{0,1}}$ [cm, mgon]	äußere $\overline{\sigma_{0,1}}$ [-]	Beobach- tungstyp
	$\hat{\sigma}$ [10 ⁻³]	global λ_{max} λ_{min} [-]	lokal $\hat{\sigma}_p$ [cm]	-	-			
Hier- arch. Netz	1.1	232.0	6.1	0.8	.36-0. .43-0	2.6-3.5 3.7-5.0	2.3-4.3 2.0-4.6	Richtung Strecke
Dyna- misch. Netz	1.0	962.2	1.3	1.4	.33-0. .42-0	2.5-3.5 3.7-5.0	2.4-4.4 2.1-4.7	Richtung Strecke

Vergleich der Genauigkeits- und Zuverlässigkeitsmaße im hierarchischen und dynamischen Netz

Tab.3.2

Der Rangdefekt im dynamischen Netz wird mit $d = 4$ vorgegeben. Die Abbildungen 3.2 und 3.3 zeigen die Punktfehlerellipsen für die beiden Datumsvarianten.

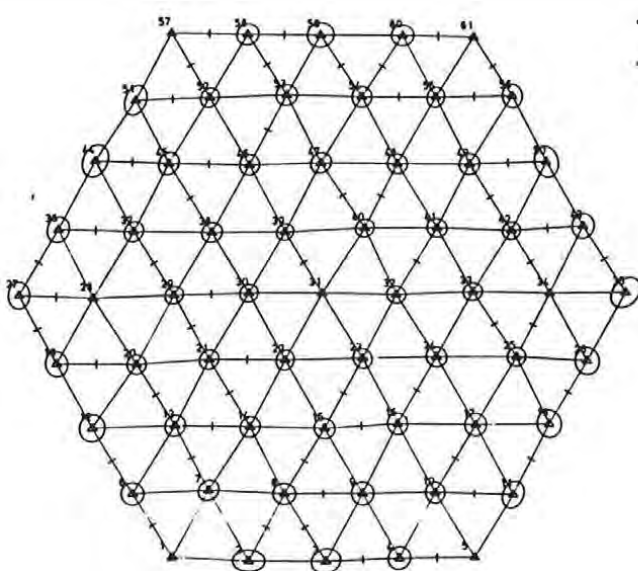


Abb. 3.2 Genauigkeitssituation bei hierarchischer Netzausgleichung

2 cm Ellipsen
1000 m Netz

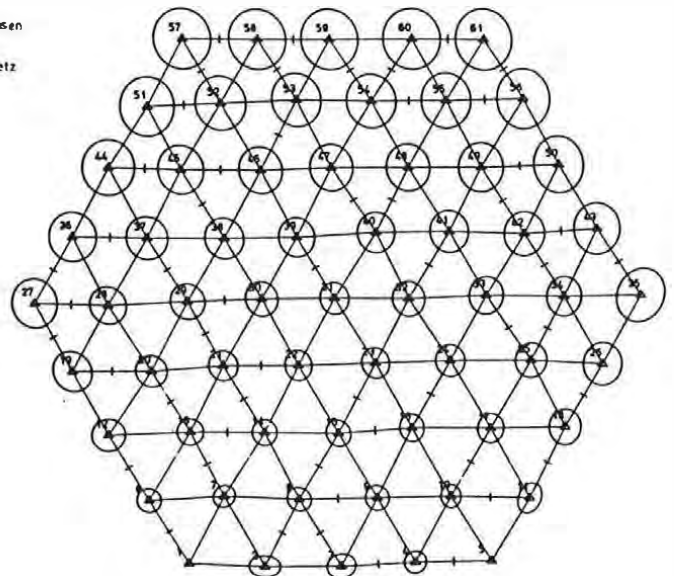


Abb. 3.3 Genauigkeitssituation bei dynamischer Netzausgleichung

Aus der Gegenüberstellung in Tabelle 3.1 bzw. Tabelle 3.2 lassen sich folgende Erkenntnisse zusammenfassen:

1. Die Genauigkeitsmaße bei der Ausgleichung mit hierarchischer Datumfestsetzung fallen zu optimistisch aus.

2. Eine wesentliche Erhöhung der Kontrollierbarkeit (Zuverlässigkeit) geodätischer Beobachtungen ist bei der dynamischen Datumsvorgabe nicht gegeben.
3. In schlecht kontrollierbaren übergeordneten Netzen werden die Beobachtungen des übergeordneten Netzes durch die Beobachtungen des Verdichtungsnetzes kontrolliert.

An Nachteilen der dynamischen Datumsvorgabe ist in erster Linie der große Berechnungsaufwand und die Notwendigkeit, große Beobachtungsmengen zur Verfügung zu halten, zu nennen. Üblicherweise müssen auch mehr als 2 Hierachiestufen von Netzen behandelt werden können.

3.3 Datumsvorgabe mittels Ersatzbeobachtungen

Der Nachteil des großen Berechnungsaufwandes bei der dynamischen Datumsvorgabe kann nach Pelzer (1980) reduziert werden. Sollen durch die Verdichtungsmessungen nur Änderungen in den Koordinaten und Genauigkeiten unmittelbar von Beobachtungen betroffener oder benachbarter Anschlußpunkte möglich sein, braucht in der Ersatzkovarianzmatrix nur die Stochastizität dieser Punktgruppe berücksichtigt werden. Das funktionale Modell bzw. das lineare Verbesserungsgleichungssystem:

$$l_2 + v_2 = A_1 x + A_2 y \quad (3-9)$$

wird um folgende Verbesserungsgleichungen erweitert:

$$l_y + v_y = 0 x + I y \quad (3-10)$$

Die Beobachtungsgruppe l_2 bezeichnet die Beobachtungen des Verdichtungsnetzes. I beschreibt eine Einheitsmatrix von der Dimension der Anzahl der Festpunktsunbekanntenkoordinaten. Die Beobachtungen gemäß (3-10) werden als Ersatzbeobachtungen bezeichnet. Die Beobachtungen der Gruppe l_2 gelten als von den Ersatzbeobachtungen unabhängig. Die Gewichtsmatrix kann daher angegeben werden zu:

$$P = \begin{bmatrix} P_{11} & 0 \\ 0 & P_{yy} \end{bmatrix} \quad (3-11)$$

Damit ergibt sich das Normalgleichungssystem nach Koch (1984) zu

$$\begin{bmatrix} A_1^T P_{11} A_1 & A_1^T P_{11} A_2 \\ A_2^T P_{11} A_1 & A_2^T P_{11} A_2 + P_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} A_1^T P_{11} l_2 \\ A_2^T P_{11} l_2 + P_{yy} l_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3-12)$$

Bei der praktischen Durchführung dieses Verfahrens werden die l_y durch Gleichsetzen der Näherungswerte für die p Festpunktskoordinaten mit deren Eingangswerten zu Null gemacht. Damit entfällt im Vektor $A^T P l$ des Normalgleichungssystems (3-12) der Ausdruck $P_{yy} l_y$, und es verbleibt lediglich eine Modifikation eines Blockelementes der Normalgleichungsmatrix gemäß (3-12). Nunmehr steht die Frage nach dem stochastischen Modell der Ersatzbeobachtungen an. Als Ersatz für die nicht bekannte Varianz-Kovarianzmatrix Σ_{yy} der Anschlußunbekannten empfiehlt Baarda (1973) die Einführung einer Kriteriummatrix K_{yy} , die synthetisch, von wenigen Parametern abhängig, das Genauigkeitsverhalten idealisiert. Solche Kriteriummatrizen werden speziell auch bei der Optimierung geodätischer Netze angewandt.

$$K_{yy} = \begin{bmatrix} \sigma_{x_1}^2 & & & \\ & \sigma_{y_1}^2 & & \\ & & \ddots & \\ & & & \sigma_{x_p/2^2} \\ & & & & \sigma_{y_p/2^2} \end{bmatrix} \quad (3-13)$$

Die einfachste Vorgangsweise ist, dabei unkorrelierte Ersatzbeobachtungen anzunehmen und damit alle Kovarianzen in der Matrix K_{yy} zu Null zu machen. Die Werte der Varianzen können zum Beispiel einem Ausgleichsprotokoll der höheren Ordnung entnommen werden. Bei dieser Kriteriummatrix bleiben die Nachbarschaftsbeziehungen der Festpunkte unberücksichtigt.

Ein Vorteil der Annahme von unkorrelierten Ersatzbeobachtungen ist die Verwendbarkeit des Testverfahrens data snooping für unkorrelierte Beobachtungen nach Baarda (1968). Für diese Arbeit wurde

in weiterer Folge diese Art der Vorgabe der Kriteriummatrix gewählt. In Bill (1984) wurden darüber hinaus korrelierte Ersatzbeobachtungen untersucht und zu dem Modell (3-13) in Beziehung gebracht. Zur Angabe von vollbesetzten Kriteriummatrizen ist die Definition von Korrelationsfunktionen erforderlich. Zielvorstellungen bei dieser Vorgangsweise sind dabei:

1. Die Punktfehlerellipsen sind kreisförmig (Isotropie)
2. Die Punktfehlerellipsen sind gleich groß (Homogenität)
3. Die relativen Fehlerellipsen erfüllen die Forderungen nach Homogenität und Isotropie.

Werden die Bedingungen 1 und 2 erfüllt, spricht man von einer Kriteriummatrix mit TAYLOR-KARMANN- Struktur. Baarda (1973) erfüllt mit einer speziellen Kriteriummatrix auch die dritte Forderung.

In Bill (1984) ist eine Gegenüberstellung der Verwendung von verschiedenen Kriteriummatrizen gegeben. Für den hier im weiteren verwendeten Modellansatz der Diagonalkriteriummatrix gemäß (3-13) zeigt sich eine sehr gute Übereinstimmung mit den Ergebnissen der strengen Ausgleichung mit dynamischer Datumsvorgabe. Es handelt sich dabei um das Netz nach Abbildung 3.1 bzw. Abbildung 3.2 und 3.3.

Benning (1988) beschreibt eine ähnliche Vorgangsweise mit Ersatzbeobachtung als "weiche" Lagerung von Lagenetzen und gibt als wesentlichen Vorteil die Möglichkeit der Kompensation von vorhandenen Netzspannungen an. Eine interessante Variante zur Behandlung von freien Netzen geben Schmid, Scherrer (1980), die die Auflösung von freien Netzausgleichungen auf die Definitionen von Pseudo-beobachtungen zurückführen. Dabei werden die Koordinaten des Netzes als zusätzliche Beobachtungen mitgeführt. In der Photogrammetrie wird diese Methodik zur Kombination von beobachteten Koordinaten mit relationalen Beobachtungsformen in einem Ausgleichungsmodell verwendet (Kraus, 1984). Durch Variation der Festpunktauswahl kann ein Festpunktrahmen mit minimalem Festpunktzwang bestimmt werden und, falls gewünscht, abschließend mit hierarchischer Datumsvorgabe erneut ausgeglichen werden. Ein Beispiel in Abschnitt 8.2 soll diese Vorgangsweise an Hand des EP - Netzes Pottendorf demonstrieren.

4. Statistische Hypothesentests zur Modellüberprüfung

Die Parameterschätzung im GAUSS-MARKOV - Modell (2-9) beruht auf der Voraussetzung, daß sich die Erwartungswerte der Beobachtungen mit der Matrix A als Linearkombinationen der Parameter darstellen lassen. Weiters gilt, daß die Gewichtsmatrix der Beobachtungen bekannt ist. Nach Koch (1987) sollen 3 mögliche Modellabweichungen und ihre Konsequenzen beschrieben werden:

1. Die Anzahl der Parameter wurde zu klein gewählt.

Es wurde also ein zu einfaches Ausgleichungsmodell gewählt. Die Schätzwerte der Parameter sind dann im allgemeinen verzerrt, also nicht erwartungstreu. Varianzen der geschätzten Parameter ergeben sich im allgemeinen als zu kleine Werte. Ist die Varianz der Gewichtseinheit $\sigma^2 = 1$ bekannt, so kann aus einem Schätzwert $s_0^2 > 1$ nach der Ausgleichung auf zu wenige Parameter im Modell geschlossen werden.

2. Die Anzahl der Parameter ist zu groß.

Die Schätzwerte der Parameter ergeben sich erwartungstreu und deren Varianzen der geschätzten Parameter ergeben sich im allgemeinen als zu große Werte. Die Varianz der Gewichtseinheit s_0^2 wird erwartungstreu geschätzt.

3. Fehlerhafte Gewichtsmatrix der Beobachtungen.

Die Berechnung des Modelles ergibt für die Parameter eine erwartungstreue Schätzung. Verzerrt erhält man die Schätzung der Varianz der Gewichtseinheit.

Insbesondere der dritte Fall könnte bei üblichen Netzkonfigurationen eintreten. Es sollte daher eine Überprüfungsmöglichkeit der Gewichtsmatrix P bestehen. Diese wird in dem Abschnitt 4.3 global für das gesamte Modell und im Abschnitt 4.4 für Beobachtungsgruppen angegeben. Die ersten beiden Fälle dürften durch die übliche Vorgabe der Zielsetzung eines geodätischen Netzes umgangen werden können.

Bei der Formulierung der Methode der kleinsten Quadrate zur Bildung des GAUSS - MARKOV - Modells wurde folgende Forderung bezüglich des funktionalen Modells eingeführt:

Die Beobachtungen sollen Realisierungen normalverteilter Größen darstellen.

Es sollen nur zufällige Fehler in den Beobachtungswerten enthalten sein. Ein grober Fehler würde eine Verzerrung der angenommenen Normalverteilung je Beobachtung bewirken. Das Erkennen solch fehlerhafter Beobachtungen, seien es geodätische Beobachtungen oder Ersatzbeobachtungen, ist das weitere Ziel der Abschnitte 4.1 bzw. 4.2 . Sehr grobe Datenfehler lassen sich mit nicht statistischen Methoden, etwa Plausibilitätskontrollen, eliminieren. Ursachen sind dabei Punktverwechslungen, Ablesefehler, Reduktionsfehler usw. Nach Heck (1981) u. a. stehen 3 Teststatistiken zur Prüfung der unkorrelierten Beobachtungen zur Wahl, die sich im weiteren im verwendeten Varianzfaktor und der daraus resultierenden Verteilungsfunktion unterscheiden. In jedem Fall werden dazu die normierten Verbesserungen w_i als Testgrößen verwendet:

$$w_i = \frac{|v_i|}{\sigma_{v_i}} = \frac{|v_i|}{s_0 \sqrt{q_{v_i v_i}}} \sim N(0,1) \quad (4-1)$$

Diese Testgrößen folgen unter der Voraussetzung nicht korrelierter Beobachtungen einer normierten Normalverteilung. Die Elemente q_{vv} stellen die Hauptdiagonalglieder der Kofaktorenmatrix der Verbesserungen als Anteil aus (2-23) dar:

$$q_{vv} = q_{11} - A Q A^T \quad (4-2)$$

4.1 Grobe Fehler in den Beobachtungen

Jede Beobachtung l_i besitzt als stochastische Größe eine bestimmte Wahrscheinlichkeitsverteilung. Üblicherweise, unter der Annahme von

ausschließlich zufälligen Beobachtungsfehlern handelt es sich dabei um eine Normalverteilung. Kann der Meßwert nicht mehr als Stichprobe aus der Grundgesamtheit mit der Wahrscheinlichkeitsverteilung angesehen werden, ist also ein grober Beobachtungsfehler enthalten, gehört er einer nicht-zentralen Verteilungsfunktion an. Der grobe Fehler einer Funktion soll mit V_{11} bezeichnet werden.

4.1.1 Data snooping nach Baarda

Dabei wird in der richtungsweisenden Arbeit von Baarda (1968) die Kenntnis des a priori - Varianzfaktors σ_0^2 vorausgesetzt. Der sogenannte Globaltest

$$T = \frac{s_0^2}{\sigma_0^2} \sim F_{n-u, \infty} \quad (4-3)$$

folgt als Quotienten zweier X^2 - verteilter Testgrößen der F - Verteilung. Die Konsequenz der Ablehnung der Nullhypothese

$$H_0 : E(s_0^2) = \sigma^2 \quad (4-4)$$

ist die Annahme von Modellfehlern bzw. der Übergang von einer zentralen zu einer nicht zentralen F - Verteilung:

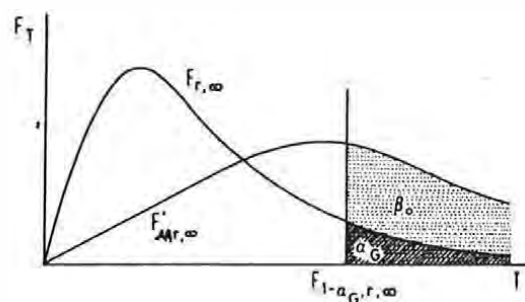


Abb. 4.1

Die Lage der Alternativhypothese ist abhängig vom Nichtzentralitätsparameter

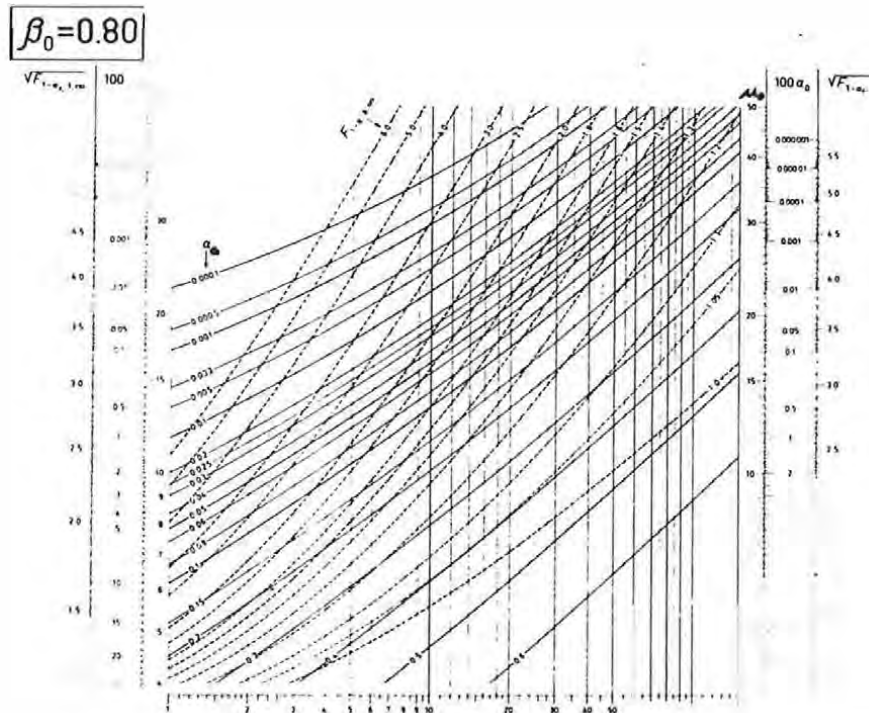
$$\mu_0 = f(\alpha_G, \beta_G, n-u, \infty) \quad (4-5)$$

Nach Baarda wird als Ursache für die Ablehnung der Nullhypothese die

Anwesenheit eines groben Fehlers V_{1i} in einer Beobachtung angenommen. Zur Aufdeckung der fehlerhaften Beobachtung wird für jede der n - Beobachtungen die Nullhypothese ($H_0 : E(V_{1i}) = 0$) gegen die Alternativhypothese ($H_A : E(V_{1i}) \neq 0$) geprüft. Als Testgröße für jede der n - Hypothesen dient unter den angegebenen Voraussetzungen die normierte Verbesserung (4-1) der einzelnen Beobachtungen. Ist w_i größer als das $\alpha/2$ - Fraktile der angenommenen Normalverteilung, wird in dieser Beobachtung der grobe Fehler vermutet. Wird α_0 etwa zu 0.1% gewählt, ergibt sich der Vergleichswert zu 3.3 (Reißmann 1980). Sollten mehrere grobe Fehler in den Beobachtungen aufgedeckt werden, wird die Beobachtung mit der größten Testgröße als erstes eliminiert und eine komplette Neuberechnung durchgeführt. Dies ist erforderlich, da ja eine Beobachtung auf alle anderen Beobachtungen über die Q_{vv} - Matrix rückwirkt. Eine Abweichung von diesem Grundsatz wird von manchen Autoren (Förstner, 1979) und anderen toleriert, wenn die grob fehlerhaft erkannten Beobachtungen nicht in funktionalem Zusammenhang aus dem Beobachtungsplan stehen. In der Originalableitung von Baarda (1968) wird die Irrtumswahrscheinlichkeit des mehrdimensionalen Globaltests (4-3) auf die Irrtumswahrscheinlichkeit des eindimensionalen Tests (4-1) bei gleichbleibender Güte bzw. Macht des Tests $\beta_G = \beta_0$ und gleichem Nichtzentralitätsparameter μ_0 abgestimmt.

$$\mu_0 = f(\alpha_0, \beta_0, 1, \infty) = f(\alpha_G, \beta_0 = \beta_G, n-u, \infty) \quad (4-6)$$

Dabei wird bei einer kleinen Anzahl von Überbestimmungen r der Vertrauensbereich groß und damit der Globaltest unempfindlich. Alternativ werden bei wachsender Redundanz r der Globaltest unter Umständen überempfindlich oder die Einzeltests verlieren an Aussagekraft. Als Abhilfe wird allgemein (Niemeier, 1985) die Partitionierung der zu untersuchenden Lagenetze in kleinere, funktional wohl zusammenhängende Netzteile empfohlen (Carosio, 1983). Die Wahl der Irrtumswahrscheinlichkeit α_0 der eindimensionalen Tests bzw. deren Sicherheitswahrscheinlichkeiten wird mit gleichbleibender Testgüte β_0 und gleichbleibendem Nichtzentralitätsparameter μ_0 festgelegt (B-method of testing). Das funktionale Zusammenwirken von (4-6) wurde von Baarda in Nomogrammform dokumentiert. Abb. 4.2 zeigt das für $\beta_0 = 80\%$ zu verwendende Nomogramm aus Baarda (1968):



$$\delta_o = \sqrt{\mu_o}$$

Abb. 4.2

Wird etwa $\alpha_o = 0.1\%$ gewählt, entspricht das einer Übereinstimmung $r = 12$ des Netzes eines $\alpha_G = 5\%$ und $\delta_o = \sqrt{16.9} = 4.11$.

4.1.2 Tau - Kriterium nach Pope

Nach Pope (1976), Höggerl (1986) läßt sich dieses Verfahren zur Aufdeckung grober Beobachtungsfehler im Falle eines wenig zuverlässigen Varianzfaktors s_o^2 anwenden. Ist die τ - verteilte Testgröße

$$\tau_{\perp} = \frac{\|v_{\perp}\|}{s_o \sqrt{Q_{v_{\perp} v_{\perp}}}} \sim \tau_{n-u} \quad (4-7)$$

größer als das α_{τ} - Fraktile der τ - Verteilung bei $(n - u)$ Freiheitsgraden wird in dieser Beobachtung ein grober Beobachtungsfehler vermutet. Die Irrtumswahrscheinlichkeit α_{τ} für den Test ist abhängig nach Heck (1981) von der Irrtumswahrscheinlichkeit des Gesamtsystems α_G :

$$\alpha_G = 1 - (1 - \alpha_{\tau})^{1/\bar{n}} \quad (4-8)$$

wobei \bar{n} die Anzahl der Beobachtungen bezeichnet, für die sich durch die Ausgleichung ein Genauigkeitsgewinn $q_{v_1 v_1} > 0$ ergibt. Auch bei diesem Verfahren muß bei Auftreten mehrerer grober Fehler ein schrittweises Suchen angewandt werden.

4.1.3 t - Test zur Grobfehlersuche

Eliminiert man den Verbesserungsquadratsummenanteil (2-18) der zu untersuchenden Beobachtung l_1 , wird der Schätzwert für den Varianzfaktor der Gewichtseinheit zu

$$\begin{aligned} \Omega &= v^T P v \\ \hat{s}_0^2 &= \frac{\Omega - v_1 v_1 / q_{v_1 v_1}}{n - u - 1} \end{aligned} \quad (4-9)$$

und ist damit unbeeinflusst von einem etwaigen groben Fehler in dieser Beobachtung l_1 . Die mit (4-9) gebildete normierte Verbesserung lautet somit:

$$t_1 = \frac{\|v_1\|}{\hat{s}_0 \sqrt{q_{v_1 v_1}}} \sim t_{n-u-1} \quad (4-10)$$

die Testgröße t_1 ist t - mit $(n - u - 1)$ Freiheitsgraden.

4.1.4 Multipler t - Test

Dieser beruht auf einer Varianzanalyse, der aus der Ausgleichung stammenden Verbesserungen, wobei gegenüber (4-9) die Rückwirkung eines Beobachtungswertes auf alle anderen Verbesserungen berücksichtigt wird. Der Einfluß eines groben Meßfehlers auf die gewichtete Verbesserungsquadratsumme läßt sich wie in (4-9) formulieren. Die Fehlerquadratsummen sind Realisierungen χ^2 - verteilter Größen:

$$\begin{aligned} v^T P v &\sim \chi^2_{\mathcal{E}} \\ v'^T P v' &\sim \chi^2_{\mathcal{E}-1} \quad (\text{excl. } l_1) \end{aligned} \quad (4-11)$$

Die Differenz zweier χ^2 - verteilter Zufallsgrößen ist wieder eine χ^2 - verteilte Zufallsgröße mit der Differenz der beiden Freiheitsgrade als neuem Freiheitsgrad.

$$v^T P v - v'^T P v' \sim \chi^2_1 \quad (4-12)$$

Der Quotient aus (4-12) und (4-9) stellt eine F - verteilte Prüfgröße mit dem Freiheitsgrad 1 im Zähler und (f - 1) im Nenner dar. Die Nullhypothese

$$P\{ F_1 \leq F_{1, f-1} \} = 1 - \alpha \quad (4-13)$$

ist für jede Beobachtung l_1 durchzuführen. Übersteigt F_1 den kritischen Wert der Verteilungsfunktion, wird die zugehörige Messung als grob fehlerhaft erkannt. Wie bei den anderen Verfahren bereits erwähnt, muß beim Auftreten mehrerer grober Fehler wieder schrittweise vorgegangen werden.

Der Test soll auf alle Beobachtungen gleichzeitig angewendet werden. Es ist daher ein multipler Test zu formulieren. Die Gesamtwahrscheinlichkeit für das gleichzeitige Auftreten aller F_1 ergibt sich als Produkt der Einzelwahrscheinlichkeit.

$$P\{F_1\} P\{F_2\} \dots P\{F_n\} = (1 - \alpha')^n = (1 - \alpha) \quad (4-14)$$

Die Einzelwahrscheinlichkeit α' wird als Auftreten von F_{\max} ist in Näherung nach Heck (1980) gleich α/n . Diese Größe kann mit zunehmender Anzahl von Beobachtungen sehr klein werden. Gemäß Heck (1980) sollte daher eine Unterschranke von $\alpha' = 0.001$ vorgesehen werden. Zwischen der F - Verteilung und der t - Verteilung besteht der Zusammenhang

$$t_{f-1, 1-\alpha/2}^2 = F_{1, f-1, 1-\alpha} \quad (4-15)$$

$$\sqrt{F_1} = \frac{\|v_1\|}{s_0 \sqrt{q_{v_1 v_1}}} \geq t_{f-1, 1-\alpha/2} \quad (4-16)$$

Dieser Test wird als "multipler t- Test" bezeichnet. Er unterscheidet sich von dem unter 4.1.1 beschriebenen data snooping nach Baarda durch die Verwendung der bereinigten empirischen Standardabweichung (4-9).

In der vorliegenden Arbeit wurde nach Vergleich der angeführten Testformen das verbreitete "data snooping" nach Baarda übernommen. Der Hauptnachteil dieses Verfahrens, ist die Voraussetzung der Kenntnis der Varianz der Gewichtseinheit, wird durch die Einführung eines Prüfverfahrens eben dieser Größe gemildert. In Übereinstimmung mit Benning (1988) und Förstner (1979) kann in geodätischen Netzwerken das stochastische Modell mittels Varianzkomponentenuntersuchung überprüft werden. Teilweise liegen auch für die speziellen Netzformen beobachtungsabhängige Gewichtsvorgaben vor (BA f.E u.Vw). Laut Grimm (1985) verliert der zuletzt angeführte multiple t - Test bei einer geringen Überbestimmung seine Tauglichkeit. Laut Bill (1984) zeigt data snooping bei Vorliegen sich gegenseitig beeinflussender grober Fehler bessere Testergebnisse.

4.2 Grobfehlersuche in den Ersatzbeobachtungen

Grobe Fehler der Ersatzbeobachtungen können zum einen aus fehlerhafter Eingabe und zum anderen aus fehlerhafter Vorausausgleichung entstammen. Als weitere systematische Ursache könnte die Herkunft der Datumsunkte aus verschiedenen Vorausausgleichungen angeführt werden. Diese Fehlerursachen führen im allgemeinen zu zweidimensionalen Punktlageänderungen, die durch geeignete zweidimensionale Tests nachgewiesen werden sollten. Im Abschnitt 4.1 wurden lediglich eindimensionale Teststellungen besprochen. Im Abschnitt 4.2.1 sollen diese Testformen auch auf Ersatzbeobachtungen angewendet werden. Im Abschnitt 4.2.2. sollen die eindimensionalen Tests durch mehrdimensionale Prüfgrößen erweitert werden. Abschließend soll eine kurze Gegenüberstellung der Testverfahren für Ersatzbeobachtungen angegeben werden.

4.2.1 Lineare eindimensionale Hypothesentests für Ersatzbeobachtungen

Die Anschlußpunktkoordinaten werden als fingierte Beobachtungen in die Ausgleichung des Verdichtungsnetzes eingeführt. Für den Erwartungswert und die Varianz gilt die Nullhypothese

$$H_0 : E(\mathbf{v}_Y) = 0; \quad V(\mathbf{v}_Y) = \sigma^2 Q_{YY} \quad (4-17)$$

Zur Prüfung eines einzelnen Koordinatenwertes wird der anteilige Verbesserungsvektor \mathbf{v}_Y mit einem Zeilenvektor einer Einheitsmatrix $\mathbf{e}_i^T = (0,0,0,1,0,\dots,0)$ multipliziert. Die Hypothese für die Ersatzbeobachtung i lautet:

$$H_0 : E(\mathbf{e}_i^T \mathbf{e}_i) = 0 \quad (4-18)$$

Für die Alternativhypothese gilt in Worten: "Eine Koordinate hat sich verändert, alle anderen Koordinaten blieben stabil". Für den Fall unkorrelierter Ersatzbeobachtungen, wie im weiteren angenommen, ergibt sich in gleicher Weise wie für geodätische Beobachtungen die Testgröße in Form der normierten Verbesserung (4-1).

$$w_{Yi} = \frac{\|v_{Yi}\|}{\sigma_{v_i}} = \frac{\|v_{Yi}\|}{s_0 \sqrt{q_{v_i v_i}}} \sim N(0,1) \quad (4-1)$$

Die Verallgemeinerung für korrelierte Ersatzbeobachtungen gibt Baarda (1968) an.

$$w_{Yi} = \frac{\mathbf{e}_i^T P_{YY} \mathbf{v}_Y}{\sigma \sqrt{(\mathbf{e}_i^T P_{YY} Q_{YY} P_{YY} \mathbf{e}_i)}} \quad (4-19)$$

Ist die Testgröße größer als $\alpha_0/2$ - Fraktile der Normalverteilung, wird in dieser Anschlußpunktkoordinate ein grober Fehler vermutet. Dabei erfolgt die Abstimmung von Irrtumswahrscheinlichkeit des Global - (4-3) und der eindimensionalen Tests der Punktkoordinaten bei gleichbleibender Güte β_0 und gleichen Nichtzentralitätsparameter μ_0 . Die Grenzwerte, ab denen die Werte der Ersatzbeobachtungen mit der Güte β_0 signifikant zur Alternativhypothese zugehörig ausgewiesen

werden können, entsprechend den Grenzwerten der Beobachtungen, und können in Abhängigkeit vom stochastischen Modell für unkorrelierte

$$V_{0y_1} = \sigma_{y_1} \sqrt{\mu_0} / r_1 \quad (4-20)$$

und für korrelierte

$$V_{0y_1} = \sigma_{y_1} \sqrt{\mu_0} / (e_1^T P_{yy} Q_{yy} P_{yy} e_1) \quad (4-21)$$

Ersatzbeobachtungen angegeben werden.

Die Verknüpfung zweier eindimensionaler Hypothesen für x- und y-Koordinaten eines Punktes läßt sich in der Form einer zweidimensionalen Grenzwertellipse in datumsabhängiger Weise formulieren (van Mierlo, 1978).

4.2.2 Mehrdimensionale Hypothesentests

Die Wahl der Verteilungsfunktion wird wieder abhängig von dem Umstand bestimmt, ob die Varianz der Gewichtseinheit a priori bekannt bzw. erprobt wurde oder unbekannt ist. Aus dem Normalgleichungssystem der hierarchischen Datumsvorgabe (3-2) läßt sich nach Koch (1987) die Prüfgröße T wie folgt angeben:

$$T = \frac{(\hat{Y} - \hat{Y})^T Q_{yy}^{-1} (\hat{Y} - \hat{Y})}{u_x \sigma^2} \sim \frac{\chi_{u_x}^2}{u_x} \sim F_{u_x, \infty} \quad (4-22)$$

Die Hypothese lautet dabei: "Alle Anschlußpunktkoordinaten können als identisch mit den Sollwerten aus der höheren Ordnung betrachtet werden". In Pelzer (1985) wird die Verfälschung des Lösungsvektors \hat{x} auf Grund grober Fehler in den Anschlußpunkten y bestimmt.

$$\nabla x = Q_{xx} N_{xy} V_y \quad (4-23)$$

und in ähnlicher Weise die Verfälschung der ausgeglichenen Beobachtungen berechnet.

$$V_l = (Q_{xx} N_{xy} - A_2) V_y \quad (4-24)$$

Siehe dazu Gleichungssystem (3-4).

Ist der a priori - Varianzfaktor der Gewichtseinheit nicht bekannt, wird folgende Fisher - verteilte Prüfgröße mit u_x und $n - u$ Freiheitsgraden nach Koch bestimmt.

$$U = \frac{\frac{(\hat{Y} - \hat{Y})_x Q_{YY}^{-1} (\hat{Y} - \hat{Y})}{u_x \sigma^2}}{n - u} \sim \frac{\chi_{u_x}^2}{n - u} \sim F_{u_x, n-u} \quad (4-25)$$

In Bill (1984) werden die Testgrößen gemäß (4-22) und (4-25) dem eindimensionalen Hypothesentest für Ersatzbeobachtungen in Form von Koordinatendifferenzen gegenübergestellt. Es zeigt sich dabei, daß bei großer Überbestimmung im Lagenetz sich die drei Testgrößen bezüglich ihrer Irrtumswahrscheinlichkeit und F - Fraktile kaum unterscheiden. Für die weitere Vorgangsweise konnte daher der eindimensionale Test für Ersatzbeobachtungen analog den eigentlichen Beobachtungen übernommen werden.

4.3 Überprüfung des stochastischen Modells

Der wesentlichste Test zur Überprüfung des stochastischen Modells im Ganzen wurde bereits angesprochen. Beim Globaltest gemäß (4-3) wird der Varianzfaktor der Gewichtseinheit mit dem sich a posteriori aus der Ausgleichung ergebenden Vergleichswert (2-19), in Beziehung gebracht.

$$T = \frac{s_0^2}{\sigma^2} \sim F_{n-u, \infty} \quad (4-3)$$

Die Hypothese lautet dabei, daß das stochastische Modell a priori die stochastischen Eigenschaften der Beobachtungswerte zueinander richtig und vollständig beschreibt. Ist (4-3) nicht erfüllt, muß die Nullhypothese H_0 abgelehnt werden. In großen Netzen sollte auch die punktweise Verteilung der lokalen Standardabweichungen der Gewichtseinheit betrachtet werden.

Für die Verteilung der lokalen Standardabweichungen der Gewichtseinheit l_{m_0} kann nach Ehlert (1984) kein allgemein gültiges Gesetz angegeben werden. In der Diagnoseausgleichung 1980 des deutschen

Hauptdreiecksnetzes wurde punktweise die Varianz der Gewichtseinheit l_{m_0} aus den den einzelnen Punkt betreffenden Beobachtungen berechnet. Sie zeigen damit Stellen auf, an denen die Beobachtungsgenauigkeiten den in der Gewichtsvergabe zugrunde liegenden Standard nicht erreichen. Die Angabe von l_{m_0} wird natürlich bei sehr geringer Anzahl von punktweiser Überbestimmung unsicher.

4.4 Varianzkomponentenuntersuchung

Wie in Abschnitt 2.2 beschrieben, wird das stochastische Modell aus Gerätegenauigkeiten oder anderen stochastischen Vorausinformationen zusammengesetzt. Insbesondere die Genauigkeitsrelation zwischen den einzelnen Beobachtungsarten kann dabei zu Problemen führen. Schließlich werden ja auch Ersatzbeobachtungen wie geodätische Beobachtungen behandelt und müssen daher im stochastischen Modell entsprechend berücksichtigt werden. Mit Hilfe des Verfahrens der Varianzkomponentenschätzung nach Welsch (1981) besteht die Möglichkeit, die stochastische Modellbildung in Gruppen zu verifizieren. Bei der Gruppenbildung sind zwei wesentliche Richtlinien zu berücksichtigen:

- die Beobachtungsgruppen sollten ausreichend überbestimmt sein und
- die Beobachtungsgruppen sollten über die Designmatrix A funktional im ausreichenden Maß in Beziehung stehen.

Üblicherweise werden Beobachtungsgruppen auf Grund der Beobachtungsart gewählt. Analog ist es aber auch denkbar, alle Beobachtungen eines Punktes oder einer Punktgruppe einer Beobachtungsgruppe zuzuordnen (siehe Abschnitt 5.4). Das System der linearisierten Verbesserungsgleichungen (2-11) soll in m - Gruppen aufgeteilt werden.

$$\begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} \hat{x} \quad (4-26)$$

Die Berechnung der Varianz der Gewichtseinheit für das Gesamtsystem ist in (2-19) angegeben. Nunmehr soll für jede der m - Gruppen dieser Wert mittels einer erwartungstreuen Schätzfunktion $s_{i,i}^2$ bestimmt werden:

$$\sigma_{i,i}^2 = \frac{E(v_i^T P_{i,i} v_i)}{\text{spur}(P_{i,i} Q_{v_i v_i})} \quad i = 1, 2, \dots, m \quad (4-27)$$

$$s_{i,i}^2 = \frac{v_i^T P_{i,i} v_i}{\text{spur}(P_{i,i} Q_{v_i v_i})}$$

Der Index i bezeichnet in diesem Falle alle Beobachtungen, die zur i -ten Gruppe von 1 bis m gehören. Der Doppelindex beider Matrizen bezeichnet alle zur Beobachtungsgruppe gehörigen Matrizenelemente. Diese Abschätzung, deren komplette Ableitung in Pelzer (1985) enthalten ist, gilt in dieser Form lediglich für den Fall, daß die Beobachtungsgruppen a priori von einander unabhängig sind. Da jedoch im stochastischen Modell Abschnitt 2.2 unkorrelierte Beobachtungen vorausgesetzt wurden, und dies für den Fall der Ersatzbeobachtungen auch angenommen wird, sind natürlich auch die einzelnen Beobachtungsgruppen voneinander unabhängig. Für jede der m - Beobachtungsgruppen kann die anteilige Varianz der Gewichtseinheit abgeschätzt werden. Bei korrekter Vorgabe des stochastischen Modells müßten sich für alle Beobachtungsgruppen gleich große Werte ergeben. Aus einer eventuellen Abweichung der einzelnen Beobachtungsgruppen voneinander bzw. zu dem globalen Wert (2-19), kann auf eine Über- bzw. Unterbewertung der entsprechenden Beobachtungsgruppe entsprechend ihren Residuenanteil geschlossen werden. Dabei werden im allgemeinen Abweichungen in der Größenordnung von einigen Prozent als unschädlich angesehen. Sollte die Übereinstimmung unzureichend sein, wird der aus (4-27) bestimmte Wert in einer neuerlichen stochastischen Modellbildung als Ausgangswert verwendet. Insbesondere bei kleinen Lagenetzen sollte diese Iteration nur bei freier Netzausgleichung durchgeführt werden. Anderenfalls könnten Festpunktzwänge die Schätzungen der gruppenweisen Varianzkomponenten verzerren. Die punktweise Definition von Gruppen führt zu dem im vorangegangenen Abschnitt angeführten Wert l_{m0} .

5. Qualitätskriterien geodätischer Lagenetze

Nachdem in den vorangegangenen Abschnitten die Modellbildung, die Datumsvorgabe und die Modellüberprüfung behandelt wurden, soll in diesem Abschnitt die Überprüfbarkeit der Qualität eines geodätischen Lagenetzes behandelt werden. Die Kriterien zur Beurteilung der Qualität wird üblicherweise in folgende Kategorien aufgeteilt:

- Genauigkeit
- Zuverlässigkeit
- Wirtschaftlicher Aufwand

In dieser Arbeit sollen die Kriterien der Genauigkeit und Zuverlässigkeiten behandelt werden. Die Einbeziehung der wirtschaftlichen Komponente in die Überprüfung der Netzqualität erfordert eine Einbeziehung der wirtschaftlichen Zusammenhänge in die mathematische Modellbildung. Versuche dazu sind etwa in Augath (1985) angeführt. Das Schlüsselproblem dabei ist, allgemein gültige Kostenfunktionen zu erstellen und in die Modellbildung miteinzubeziehen.

Das Qualitätskriterium der Genauigkeit soll im weiteren nach globalen, lokalen und funktionalen Genauigkeitsmaßen gegliedert werden. Der Begriff der Genauigkeit ist wohl der in der geodätischen Praxis verbreitete und allgemein bekannte Qualitätsmaßstab. Genauigkeitskriterien sind im allgemeinen geometrisch anschaulich und zum Teil bereits aus der Beobachtungsmethodik ableitbar. Literatur: Reißmann (1980), Höpcke (1980), Niemeier (1985), u.v.a.

Besonders berücksichtigt werden muß die Tatsache, daß die meisten Genauigkeitsangaben von der Datumsvorgabe (Abschnitt 3) abhängig sind. Die Kriterien der Zuverlässigkeit geodätischer Netze (Baarda 1968, Koch 1987, Niemeier 1985) sind demgegenüber in der Praxis zu- meist noch nicht enthalten. Eine große Anzahl von Publikationen zeigt die ungenügende Charakterisierung der Netzqualität durch Genauigkeitskriterien. In Abschnitt 8.3 soll dazu ein entsprechendes Beispiel vorgeführt werden. Alle Informationen zur Bildung der Kriterien nach Genauigkeiten und Zuverlässigkeiten sind in der Kofaktorenmatrix (2-23) enthalten:

$$Q_{xxx} = Q \quad (2-23)$$

$$Q_{xxx} = (A^T P A)^{-1} \quad (5-1)$$

Für gewisse Kriterien ist eine Spektralzerlegung (Zurmühlen-Falk, 1986) erforderlich.

$$Q_{xxx} = S D S^T \quad (5-2)$$

Die Spektralmatrix D beschreibt eine Diagonalmatrix mit den Elementen gleich den Eigenwerten Γ_i ($i=1,2,\dots,n$); in auf- oder absteigender Reihenfolge. Die Modalmatrix S enthält das System der zugehörigen normierten Eigenvektoren. Zu beachten ist, daß sowohl die Matrix Q_{xxx} als auch deren Spektralzerlegung nach Vorliegen der Punktkonfiguration des Beobachtungsplanes und der a priori Meßgenauigkeiten zu berechnen ist. Dieser Umstand ermöglicht in Abschnitt 6 die a priori Untersuchung in geodätischen Lagenetzen, wobei keine tatsächlichen Beobachtungswerte in die Berechnung einbezogen werden. Für die Berechnung der Spektralzerlegung nach (5-2) existiert im Bereich der numerischen Mathematik eine große Reihe von anwendbaren Verfahren. Es soll hier das im weiteren verwendete "klassische" Jacobiverfahren (Schwarz, 1986) wiedergegeben werden. Gegeben sei eine beliebige symmetrische Matrix M . Gesucht seien alle Eigenwerte (Spektralmatrix U) und die dazugehörigen Eigenvektoren (Modalmatrix D).

$$D = U^T M U \quad (5-3)$$

Die Diagonalelemente von D sind die Eigenwerte von M und die Spaltenvektoren von U bilden die zugehörigen orthogonalen Eigenvektoren. Das Jacobiverfahren stellt ein iteratives Verfahren dar. Die Diagonalfom wird durch eine Folge von orthogonalen Ähnlichkeitstransformationen angestrebt. Jede der i Transformationsmatrizen U_i hat folgende Form:

$$U_i = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \cos \mu & & & & \sin \mu & \\ & & & i & & & & \\ & & -\sin \mu & & & & \cos \mu & \\ & & & & 1 & & & \\ & & & & & & & 1 \\ & & & & & & & & 1 \end{bmatrix} \quad (5-4)$$

$$U = U_1 U_2 U_3 \dots U_n$$

Die Ähnlichkeitstransformation der Matrix M mit U_1 führt nur zu einer Änderung der Koeffizienten in den p -ten und q -ten Zeilen bzw. Spalten von M . Beim ursprünglichen Jacobiverfahren wird zuerst das betragsgrößte Außerdiagonalelement m_{qp} von M bestimmt. Dann wird die elementare Drehmatrix gemäß (5-4) aufgebaut, sodaß m_{qp} durch die Rotation verschwindet. Danach wird aus dem System die Gesamtrationsmatrix U berechnet und damit die Eigenwerte bestimmt. Als Abbruchkriterium für dieses iterative Verfahren kann die Eigenwertänderung von der Iterationsstufe i zur Iterationsstufe $i + 1$ benützt werden. In (Schwarz, 1986) und (Zurmühl -Falk, 1986) sind Modifikationen dieses klassischen Verfahrens mit Konvergenzbeschleunigung und Verfahren zur Behebung numerischer Berechnungsunschärfen enthalten.

Jede symmetrische Matrix der Ordnung n besitzt n - reelle Eigenwerte als Nullstellen des charakteristischen Polynoms. Im Falle mehrfacher Nullstellen dieses Polynoms sind damit die entsprechenden Eigenwerte mehrfach und die dazugehörigen Eigenwerte nicht mehr eindeutig. Jede normierte Linearkombination der gefundenen Eigenvektoren ist damit ebenfalls selbst Eigenvektor.

Dieses Phänomen tritt etwa bei der Berechnung der Punktfehlerellipsen auf: die Längen der Hauptachsen der Ellipsen (entsprechen den Quadratwurzeln der Eigenwerte von Q_{xx}) sind immer eindeutig bestimmbar; die Richtung der Achsen können jedoch unbestimmbar werden, wenn die Ellipse zu einem Kreis degeneriert (Hauptachsenproblem für eine Teilmatrix).

In Abb.5.1 aus (Niemeier, 1985) wird eine systematische Zuordnung zwischen einer symbolisierten Kofaktorenmatrix (beschränkt auf den Parametervektoranteil der Neupunktkoordinaten) und der einzelnen Genauigkeitskriterien symbolhaft aufgezeigt.

Die Beziehung zur entsprechenden Kovarianzmatrix lautet:

$$\Sigma_{xx} = \sigma_0^2 Q_{xx} \quad (5-5)$$

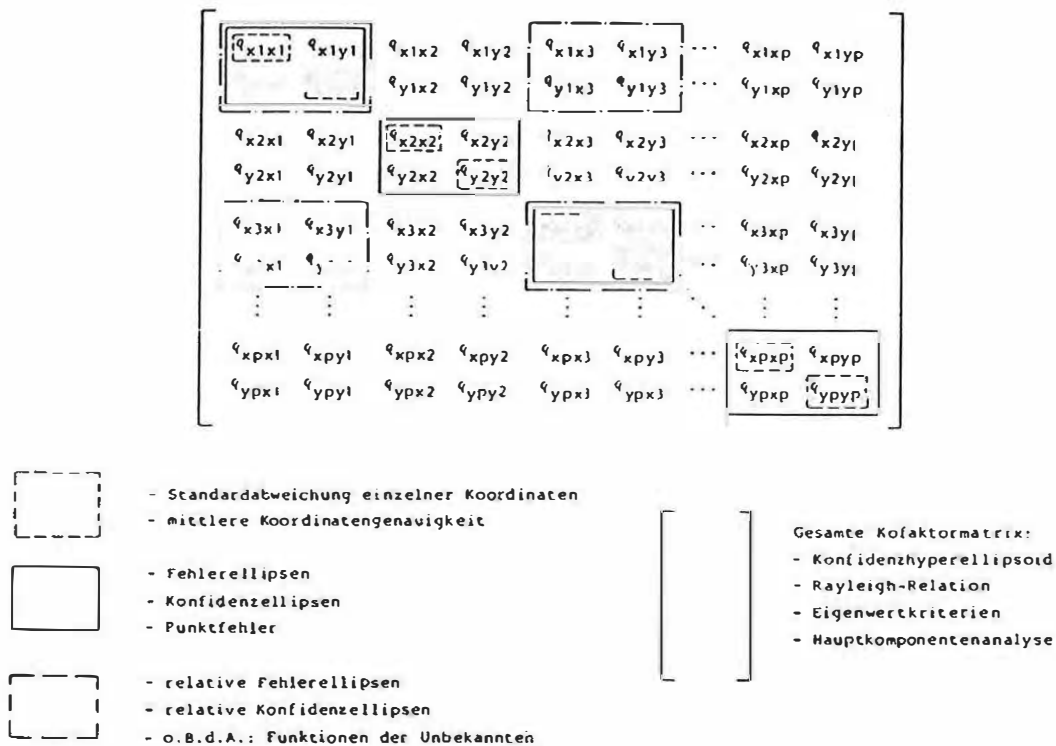


Abb. 5.1

Dabei ist zu unterscheiden, ob die Varianz der Gewichtseinheit σ_0^2 bekannt ist oder als verlässlicher Schätzwert s_0^2 nach (2-19) auf Grund ausreichender Überbestimmungen berechnet wurde. Wird in (5-5) σ_0^2 durch s_0^2 ersetzt, spricht man von einer empirischen Kovarianzmatrix. Im anderen Fall spricht man von einer theoretischen Kovarianzmatrix Σ_{xx} . Die Konsequenz der Verwendung von s_0^2 oder σ_0^2 liegt bei der Berechnung der Konfidenzbereiche. Wird eine empirische Kovarianzmatrix verwendet, gilt die F - Verteilung mit $f = (n - u)$ Freiheitsgraden; andernfalls gilt die χ^2 - Verteilung.

5.1.1 Globale Genauigkeitsangaben

Globale Genauigkeitsmaße beschreiben die Güte eines Netzes insgesamt. Sie hängen nach Abb. 5.1 von allen Elementen der Kofaktorenmatrix ab. Eine globale Genauigkeitsaussage kann in der Form eines u - dimensional Konfidenzhyperellipsoides berechnet werden. Es ist durch folgende Wahrscheinlichkeitsaussage definiert:

$$P\{(\mathbf{X} - \bar{\mathbf{X}})^T \Sigma_{xx} (\mathbf{X} - \bar{\mathbf{X}}) \leq \chi^2_{u, 1-\alpha}\} = 1-\alpha \quad (5-6)$$

$\chi^2_{u, 1-\alpha}$. . . Fraktile der χ^2 - Verteilung mit u Freiheitsgraden und einer Sicherheitswahrscheinlichkeit von $S = 1-\alpha$.

Es umhüllt die wahren Koordinaten sämtlicher Punkte mit der vorgegebenen Sicherheitswahrscheinlichkeit $1 - \alpha$, für den Fall, daß der theoretische Varianzfaktor σ^2 angewandt wird. Die Halbachsen dieses u - dimensionalen Konfidenzhyperellipsoides lassen sich mit Hilfe der Spektralzerlegung (5-2) angeben

$$\Sigma_{xx} = S D S^T \quad (5-7)$$

Die geordneten Eigenwerte als Elemente der diagonalen Spektralmatrix beschreiben die Quadrate der Halbachsen des Hyperellipsoides. Für die Halbachsen des Konfidenzhyperellipsoides gilt somit:

$$A_i^2 = \Gamma_i \chi^2_{u, 1-\alpha} \quad i=1 \dots u \quad (5-8)$$

Als Kriterium für die globale Netzgenauigkeit kann nun etwa das Volumen dieses Hyperellipsoides betrachtet werden. Es ist damit proportional dem Produkt der Halbachsen bzw. der Determinante der Kovarianzmatrix Σ_{xx} .

$$\text{Vol} \sim \prod_{i=1}^u \Gamma_i = \det(\Sigma_{xx}) \dots \text{Minimum} \quad (5-9)$$

Dieses Volumskriterium versagt bei anisotropen Netzen. Es könnten einzelne Achsen relativ groß ausfallen, obwohl das Volumen selbst minimal ist. Dieser Nachteil kann durch Übergang auf ein Varianzkriterium umgangen werden. Dabei wird die Summe der Achsenlängen minimiert:

$$\sum_{i=1}^u \Gamma_i = \text{spur}(\Sigma_{xx}) \dots \text{Minimum} \quad (5-10)$$

Dabei beschreibt der Ausdruck (5-10) den Erwartungswert des Betrages des Vektors der wahren Fehler der Koordinaten bis auf einen konstanten Faktor σ_0^2 in einem Netz mit p Neupunkten.

$$E((\epsilon_x)^2) = \sum_{i=1}^{2p} \sigma_{xi}^2 = \sigma_0^2 \text{ spur}(Q_{xx}) \quad (5-11)$$

In (Grimm-Pitzinger, Hanke, 1985) wird dieses Varianzkriterium als durchschnittliche Standardabweichung der Koordinaten formuliert:

$$\sigma_D = \sigma_0 \sqrt{\text{spur}(\hat{Q}_{xx})} / (2 p) \quad (5-12)$$

Ein weiteres globales Genauigkeitskriterium ergibt sich aus der Forderung:

$$\Gamma_{\max} \dots \text{Minimum} \quad (5-13)$$

Dabei werden möglichst kleine Eigenwerte bzw. Halbachsen gesucht. Die bisher besprochenen globalen Genauigkeitskriterien verwendeten aus der Spektralzerlegung lediglich die Diagonalmatrix. Sollen Kriterien bezüglich der Richtungen der Halbachsen formuliert werden, muß auch die Modalmatrix S betrachtet werden. Ein wesentliches Verfahren, die sogenannte Hauptkomponentenanalyse, untersucht dabei lediglich den zum maximalen Eigenwert gehörigen Eigenvektor. Damit ist eine Möglichkeit gegeben, die Empfindlichkeit bzw. Sensitivität eines Netzes in bestimmte Richtungen zu untersuchen. Dieses speziell bei Rutschungsnetzen angewandte Kriterium soll getrennt in Abschnitt 5.3 formuliert werden. Ein weiteres Genauigkeitskriterium soll nunmehr die Varianz einer beliebigen Funktion der Komponenten des Parametervektors X minimieren.

$$f = f^T x$$

$$\sigma_f^2 = f^T \Sigma_{xx} f \quad (5-14)$$

Mit Hilfe der Rayleigh - Relation ist die Angabe eines oberen und unteren Grenzwertes für die möglichen Varianzen (5-14) möglich.

$$\Gamma_{\min} \leq \frac{f^T \Sigma_{xx} f}{f^T f} \leq \Gamma_{\max} \quad (5-15)$$

Damit ist auf diesem Weg eine Belegung der Form (5-13) angegeben.

5.1.2 Lokale Genauigkeitsangaben

Diese Genauigkeitsmaße stellen die in der geodätischen Praxis zu-
meist ausschließlich verwendeten Qualitätskriterien dar. Es wird
nunmehr nicht mehr die gesamte Kofaktorenmatrix Q_{xxx} verwendet,
sondern lediglich ein den jeweiligen Punkt betreffender Ausschnitt
(siehe Abb. 5.1). Im Extremfall werden dabei nur das Haupt-
diagonalelement von Q_{xxx} verwendet:

$$\begin{aligned} \sigma_{x1} &= \sigma_0 \sqrt{q_{x1x1}} & \text{bzw.} \\ s_{x1} &= s_0 \sqrt{q_{x1x1}} \end{aligned} \quad (5-16)$$

Für Netzpunkte im Lagenetz werden im allgemeinen geometrisch an-
schauliche Genauigkeitsbereiche formuliert. Dazu wird eine (2,2) -
Submatrix Q_{jj} aus der Gesamtkofaktorenmatrix Q_{xxx} herausgegriffen.
Die Helmert'sche Punktfehlerellipse läßt sich dann mittels
Spektralzerlegung der Matrix Q_{jj} angeben:

$$Q_{jj} = S_j D_j S_j^T = \begin{bmatrix} s_1 & s_2 \end{bmatrix} \begin{bmatrix} \Gamma_1 & 0 \\ 0 & \Gamma_2 \end{bmatrix} \begin{bmatrix} s_1^T \\ s_2^T \end{bmatrix} \quad (5-17)$$

$$A_F^2 = \sigma_0^2 \Gamma_1$$

$$B_F^2 = \sigma_0^2 \Gamma_2$$

$$\tan \theta_F = \frac{s_{1y}}{s_{1x}} \quad \begin{array}{l} \text{y- Komponente des Vektors } s_1 \\ \text{x- Komponente des Vektors } s_1 \end{array}$$

Analog zur Überlegung zum Konfidenzhyperellipsoid (5.6) mit vor-
gegebener Sicherheitswahrscheinlichkeit $1 - \alpha$ lautet die Wahr-
scheinlichkeitsbeziehung der Konfidenzellipse:

$$P\{(\mathbf{x}_j - \hat{\mathbf{x}}_j)^T Q_{jj} (\mathbf{x}_j - \hat{\mathbf{x}}_j) \leq \sigma_0^2 \chi^2_{2, 1-\alpha}\} = 1-\alpha \quad (5-18)$$

\mathbf{x}_j zweidimensionaler, zu P_1 gehöriger Vektor der wahren Koordinaten
 $\hat{\mathbf{x}}_j$ zweidimensionaler, zu P_1 gehöriger Vektor der plausibelsten
Koordinaten

Die wahre Lage des Punktes liegt mit einer Wahrscheinlichkeit von $(1 - \alpha)$ innerhalb der Ellipse:

$$\begin{aligned} A_K^2 &= \sigma_0^2 \Gamma_1 \\ B_K^2 &= \sigma_0^2 \Gamma_2 \\ \theta_K &= \theta_F \end{aligned} \quad (5-19)$$

Die Werte der Halbachsen der Konfidenzellipsen stehen zu den Halbachsen der Helmert'schen Punktlagefehlerellipsen in einem konstanten Verhältnis. Helmert'sche Punktlagefehlerellipsen repräsentieren damit ein niedrigeres Signifikanzniveau, abhängig von der Anzahl der Freiheitsgrade des Netzes zwischen 29% und 39%. Der Zusammenhang wird abhängig vom Freiheitsgrad f in der folgenden Tabelle aus (Pelzer, 1980) verdeutlicht:

f	S_H
1	0.293
2	0.333
5	0.366
10	0.379
∞	0.394

Tab. 5.1

Sicherheitswahrscheinlichkeit S_H der Fehlerellipse nach Helmert als Funktion des Freiheitsgrades f .

Eine weitere Tabelle 5.2 zeigt das Achsverhältnis zwischen den Halbachsen der Konfidenzellipsen mit einer vorgegebenen Sicherheitswahrscheinlichkeit S und den entsprechenden Halbachsen der Helmert'schen Punktfehlerellipse:

f	$A_K / A_{K.H}$	
	$S=95\%$	$S=99\%$
10	2.86	3.89
20	2.64	3.42
40	2.54	3.22
60	2.51	3.16
∞	2.45	3.04

Tab. 5.2

Ist die Varianz der Gewichtseinheit nicht bekannt sondern wird aus den Beobachtungen gemäß (2-19) geschätzt, wird σ_0^2 durch den Schätzwert s_0^2 in den Formen (5-17), (5-18) ersetzt. Wie im Kapitel 5.1 angeführt, gilt nun allerdings für die Wahrscheinlichkeitsbeziehung der Konfidenzellipse

$$P\{(\bar{x}_j - \bar{x}_j)^T Q_{jj} (\bar{x}_j - \bar{x}_j) \leq 2 s_0^2 F_{2, f, 1-\alpha}\} = 1-\alpha \quad (5-20)$$

in der das Fraktile $F_{2, f, 1-\alpha}$ der F - Verteilung auftritt. Im Fall von Lagenetzen ist dabei der erste Freiheitsgrad gleich 2 zu setzen und der zweite Freiheitsgrad gleich $f =$ Anzahl der überschüssigen Beobachtungen, welche bei der Schätzung von s_0^2 verwendet wurden. Die Kenngrößen der Konfidenzellipse berechnen sich somit:

$$\begin{aligned} A_K^2 &= 2 s_0^2 \Gamma_1 F_{2, f, 1-\alpha} \\ B_K^2 &= 2 s_0^2 \Gamma_2 F_{2, f, 1-\alpha} \\ \theta_K &= \theta_F \end{aligned} \quad (5-21)$$

Wird σ_0^2 verwendet, könnten die Werte aus den Tabellen für $F = \infty$ entnommen werden. Formuliert man das Analogon zum Volumskriterium (5-9) und zum Varianzkriterium (5-10) für die Matrix Q_{jj} , erhält man den Punktfehler nach Werkmeister:

$$s_{j^W} = s_0 \sqrt{\text{spur}(Q_{jj})} = s_0 \sqrt{\Gamma_1 + \Gamma_2} \quad (5-22)$$

beziehungsweise den Punktfehler nach Helmert:

$$s_{j^H} = s_0 \sqrt{\det(Q_{jj})} = s_0 \sqrt{\Gamma_1 \Gamma_2} \quad (5-23)$$

5.1.3 Funktionale Genauigkeitsangaben

Dabei interessiert die Frage nach der Genauigkeit von Funktionen der Komponenten des Parametervektors x . Insbesondere bei der hier angewandten Ausgleichung nach vermittelnden Beobachtungen stellen ja die Beobachtungen selbst nach der Ausgleichung solche Funktionen der Un-

bekanntes dar. Formal wurde diese Frage bereits in Abschnitt 2 in der Form (2-23) behandelt. Ein Teil des zusammengesetzten Vektors h beinhaltet solche (linearisierte) Funktionen. Anhand der sogenannten relativen Fehler- bzw. Konfidenzellipsen soll die Vorgangsweise für allgemeine Funktionen demonstriert werden. Bei den relativen Fehlerellipsen handelt es sich um Genauigkeitsmaße, welche die Relativgenauigkeit zwischen den Punkten P_1 und P_k beschreiben. Die Funktionen sind in Form von zwei Koordinatendifferenzen anzugeben:

$$dx_{1k} = \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_k - x_1 \\ y_k - y_1 \end{bmatrix} = F_{1k} \hat{x} \quad (5-24)$$

Die Funktionsmatrix F_{1k} lautet:

$$F_{1k} = \begin{bmatrix} 0 & 0 & 0 & \dots & I & \dots & I & \dots & 0 \\ 1 & 2 & 3 & & i & & k & & p \end{bmatrix} \quad (5-25)$$

Die Matrix I (2x2) beschreibt jeweils eine Einheitsmatrix. Das allgemeine Fehlerfortpflanzungsgesetz angewandt auf (5-24) ergibt die entsprechende Kovarianzmatrix

$$\Sigma_{dd^{1k}} = F_{1k} \Sigma_{xx} F_{1k}^T = S_0^2 Q_{dd^{1k}} \quad (5-26)$$

mit der Kofaktorenmatrix

$$Q_{dd^{1k}} = \begin{bmatrix} q_{dx dx} & q_{dx dy} \\ q_{dy dx} & q_{dy dy} \end{bmatrix} = Q_{11} + Q_{kk} - Q_{1k} - Q_{k1} \quad (5-27)$$

Die Matrizen Q_{11} Q_{kk} Q_{1k} Q_{k1} sind die aus der Abb. 5.1 ersichtlichen Submatrizen. Umgeformt ergeben sich die Kenngrößen der relativen Fehlerellipsen wie folgt:

$$\begin{aligned} A_{RF}^2 &= \frac{1}{2} S_0^2 (q_{dx dx} + q_{dy dy} + W_{RF}) \\ B_{RF}^2 &= \frac{1}{2} S_0^2 (q_{dx dx} + q_{dy dy} - W_{RF}) \\ W_{RF} &= (q_{dx dx} - q_{dy dy})^2 - 4 q_{dx dy}^2 \\ \tan \theta_{RF} &= \frac{2 q_{dx dy}}{q_{dx dx} - q_{dy dy}} \end{aligned} \quad (5-28)$$

Analog der Vorgangsweise bei den Helmert'schen Punktfehlerellipsen gelingt der Übergang zur relativen Konfidenzellipse des Differenzvektors wieder durch Multiplikation der Halbachsen mit den entsprechenden Faktoren. Dabei muß wieder auf die Herkunft des Varianzfaktors der Gewichtseinheit geachtet werden. Die relativen Fehlerellipsen des Differenzvektors zeigen die Nachbarschaftsgenauigkeit zwischen den Neupunkten. Sie werden durch die Datumsfestlegung weniger beeinflusst als die Punktfehlerellipsen. Relative Fehlerellipsen können auch zur Abschätzung von Nachbarschaftsbeziehungen bei hierarchischer Datumsvorgabe herangezogen werden. Die relative Fehlerellipse wird auch im Bereich der von Tunnelnetzprognosen zur Abschätzung des Längs- und Querfehlers des Durchschlages verwendet. Für die Verbindungsstrecke des Durchschlagpunktes von beiden Portalseiten gilt ein Erwartungswert 0. Die Genauigkeitsverhältnisse dieser Verbindung der Streckenlänge Null läßt sich durch die angegebene relative Fehlerellipse vollständig und datumsinvariant beschreiben. (Matthias 1984, 1988)

5.2 Zuverlässigkeitsmaße

Ein geodätisches Netz gilt als zuverlässig, wenn Modellfehler bzw. grobe Meßfehler mit hoher Wahrscheinlichkeit aufgedeckt werden können. Andererseits soll der Einfluß nicht aufdeckbarer Fehler auf das Ergebnis klein bleiben. Das Maß von Zuverlässigkeitskriterien ist dabei die Erfüllung dieser beider Forderungen. Dabei werden Beobachtungen nicht zum Erreichen einer bestimmten Genauigkeit ausgeführt, sondern beinhalten vielmehr die Information zur Überprüfung des Ausgleichsmodells auf Modellfehler. Die Zuverlässigkeitskriterien liefern numerische Werte für die Qualität dieser netzinternen gegenseitigen Kontrolle und des Grenzwertes für den maximal möglichen enthaltenen Modellfehler, der nicht mit Hilfe eines Tests aufgedeckt werden kann. Modellbildend wird der Beobachtungsvektor um einen Vektor von Fehlereinflüssen erweitert:

$$l' = l + v_l \quad (5-29)$$

In die Ausgleichung (2-17) wird jetzt mit dem Vektor l' eingegangen, da ja der Fehlervektor nicht greifbar ist.

Für den Parametervektor gilt:

$$\hat{x} = (A^T P A)^{-1} A^T P l' \quad (5-30)$$

und dieser wird in (2-11) eingesetzt:

$$v = A \hat{x} - l = (A (A^T P A)^{-1} A^T P - I) l \quad (5-31)$$

Mit Verwendung der Kofaktorenmatrix der Verbesserungen aus (2-23)

$$Q_{vv} = Q_{ll} - A (A^T P A)^{-1} A^T \quad (5-32)$$

ergibt sich der Vektor der Verbesserungen:

$$v = -Q_{vv} P l \quad (5-33)$$

Analog der Aufspaltung des Beobachtungsvektors (5-29) gilt für eine Änderung des Verbesserungsvektors

$$v' = v + \nabla v \quad (5-34)$$

In (5-34) wirkt natürlich ein grober Fehler ∇l_1 auf alle Verbesserungen zurück. Nach Baarda (1968) bzw. Förstner (1979) wird jedoch die Auswirkung eines Fehlers auf die entsprechende Verbesserung abzuschätzen sein. Dazu ist aus dem Matrizenprodukt $Q_{vv} P$ zu berechnen. Der Einfluß eines groben Fehlers in einer Beobachtung auf die entsprechende Verbesserung läßt sich wie folgt bestimmen:

$$\nabla v_1 = (Q_{vv} P)_{11} \nabla l_1 = -r_1 \nabla l_1 \quad (5-35)$$

Die Größe r_1 wird nach Förstner (1979) als Redundanzanteil r_1 der Beobachtung l_1 bezeichnet. Die Summe aller Redundanzanteile r_1 muß die Gesamtredundanz des Netzes wiedergeben.

$$\sum r_1 = n - u = \text{spur}(Q_{vv} P) \quad (5-36)$$

Die Redundanzanteile bilden die zentrale Grundlage zur Formulierung von Zuverlässigkeitskriterien. Der Wertebereich der r_i liegt zwischen 0 und 1. Anzustreben ist dabei ein Redundanzanteil für jede Beobachtung nahe 1. Sie geben, als Prozentzahl aufgefaßt an, wie stark sich ein grober Fehler in einer Beobachtung in der zugehörigen Verbesserung dokumentiert. Anzustreben ist weiters eine möglichst gleichartige Verteilung der Redundanzanteile. Dabei ergibt sich ein durchschnittlicher Redundanzanteil für jede Beobachtung:

$$\bar{r}_i = \frac{\sum r_i}{n} = \frac{n - u}{n} \quad (5-37)$$

5.2.1 Innere Zuverlässigkeit

Die innere Zuverlässigkeit gibt über die Kontrollierbarkeit der Beobachtungen innerhalb des Netzes Auskunft. Sie wird durch die Mindestgröße ∇_{01} beschrieben. Dies ist der Wert eines groben Fehlers ∇_{i1} , der mit einer vorgegebenen Mindestsicherheit gerade noch aufgedeckt werden kann. Erreicht ein grober Fehler diesen Grenzwert, spricht der Globaltest nach Abschnitt 4.1 an. Der Grenzwert hängt dabei ab von:

- der Genauigkeit der einzelnen Beobachtung (σ_{11})
- von der Netzgeometrie in der Umgebung der Beobachtung (Redundanzanteil r_i)
- und vom Signifikanzniveau $(1-\alpha)$ bzw. der Macht des Tests β_0 .

Die funktionale Beziehung der letzten Einflußgröße ist aus dem Nomogramm in Abschnitt 4.1.1. zu entnehmen. Der dabei auftretende Nicht-zentralitätsparameter μ_0 wird einerseits für den bereits erwähnten Globaltest und andererseits für die folgenden eindimensionalen Tests gleichgesetzt. Die folgende Tabelle für $\sqrt{\mu_0}$ aus Förstner (1979), faßt diese Tabelle für charakteristische α_0 bzw. β_0 zusammen:

	$\alpha_o = 0.01\%$	$\alpha_o = 0.10\%$	$\alpha_o = 1.00\%$	$\alpha_o = 5.00\%$
$\beta_o = 70\%$	4.41	3.82	3.10	2.49
$\beta_o = 80\%$	4.72	<u>4.13</u>	3.42	2.79
$\beta_o = 90\%$	5.18	4.57	3.86	3.24

Tab. 4.3

Der gesuchte Grenzwert ∇_{ol_1} ergibt sich daher zu:

$$\nabla_{ol_1} = \sigma_{11} \frac{\sqrt{\mu_o}}{\sqrt{r_1}} = \text{GWA} \quad (5-38)$$

Ist also der Redundanzanteil r_1 einer Beobachtung klein, kann erst ein relativ großer Fehler erkannt werden.

Wie erwähnt, wird jetzt für jede einzelne Beobachtung ein ein-dimensionaler Test durchgeführt. Die Voraussetzung dabei ist, daß ein grober Fehler in der Beobachtung l_1 hauptsächlich die zugehörige Verbesserung v_1 beeinflusst. Die Testgröße (4.1) nach Baarda lautet:

$$w_1 = \frac{\|v_1\|}{\sigma_{v_1}} = \frac{\|v_1\|}{\sigma_{11} \sqrt{r_1}} \quad (5-39)$$

und folgt einer standardisierten Normalverteilung. Wie erwähnt, wird die Testcharakteristik mit dem gleichen Nichtzentralitätsparameter wie der Globaltest angesetzt. Die entsprechenden Werte sind daher der Tab. 4.3 bei vorgegeben α_o bzw. β_o zu entnehmen. Da für den Globaltest und die eindimensionalen Tests sowohl die Macht der Tests als auch der Nichtzentralitätsparameter gleichbleiben, muß die Variation über die Sicherheitswahrscheinlichkeit gegeben sein. Üblicherweise wird dabei α_o der lokalen Tests vorgegeben. Aus dem Nogramm in Abschnitt 4 kann dann die für den Globaltest gültige Sicherheitswahrscheinlichkeit bei vorgegebenem Freiheitsgrad entnommen werden.

Die Werte aus (5-39) werden mit dem kritischen Wert k

$$k = \sqrt{F_{1-\alpha_o, 1, \infty}} \quad (5-40)$$

verglichen.

Es wird also geprüft, ob der Absolutbetrag der normierten Verbesserung den k -fachen mittleren Fehler σ_{v_i} überschreitet. Der Vergleich wird für jede Beobachtung durchgeführt. Für die übliche Testcharakteristik der Sicherheitswahrscheinlichkeit $\alpha_0 = 0,1\%$, Macht $\beta_0 = 80\%$ ergibt sich für $\sqrt{\mu_0} = 4,13$ und $k = 3,3$. Diese Werte wurden in der Programmentwicklung und für die Testberechnungen verwendet. Die Wahl erfolgte in Übereinstimmung mit vergleichbaren Arbeiten (Benning, 1988; Förstner, 1979; Heckmann, 1985) u.v.a.

Zu beachten ist, daß natürlich mehrere Beobachtungen grobe Fehler enthalten können. In diesem Fall ist nach dem ersten Durchlauf die Beobachtung mit dem größten Wert w_i zu eliminieren und danach eine neue Durchrechnung notwendig. Von diesem Grundsatz kann abgegangen werden, wenn die grob fehlerhaft erkannten Beobachtungen nicht zentral an einem Punkt gehäuft auftreten. In jedem Fall muß aber auf die Beobachtungsprotokolle zurückgegriffen werden und es kann daher keine automatisierte Beseitigung solcher Beobachtungen stattfinden.

5.2.2 Äußere Zuverlässigkeit

Hatte die innere Zuverlässigkeit die Kontrollierbarkeit einer Beobachtung zum Inhalt, soll nunmehr der Einfluß nicht aufgedeckter Modellfehler auf die Komponenten des Parametervektor \hat{x} bzw. daraus abgeleiteter Funktionen untersucht werden. Unter der Annahme des Auftretens nur eines groben Fehlers in GWA (5-38) ist dessen Einfluß auf den Unbekanntenvektor:

$$V_{x_i} = Q_{xx} a_i^T p_i v_{0i} \quad (5-41)$$

a_i^T . . . i -te Zeile von A^T
 p_i . . . zu a_i^T gehöriges Beobachtungsgewicht

In (5-41) wird ein vollbesetzter Vektor V_{x_i} berechnet. Ein grober Fehler einer Beobachtung wirkt also auf alle Komponenten des Parametervektors und ist überdies datumsabhängig. Um ein datumsunabhängiges Maß für die äußere Zuverlässigkeit zu erhalten, wird der durchschnittliche Einfluß eines Beobachtungsfehlers auf die Unbekannten bestimmt:

$$\delta_{01}^2 = \mu_0 = \mathbf{VX}_1^T \Sigma_{XX}^{-1} \mathbf{VX}_1$$

mit (5-41):

$$\delta_{01}^2 = (1-r_1) p_1 (\nabla_{01})^2 = \frac{1-r_1}{r_1} \mu_0 = Ef \quad (5-42)$$

In der Form (5-42) tritt wieder der Nichtzentralitätsparameter μ_0 auf. Der Wert δ_{01} wird als Netzverformung bzw. als Einflußfaktor Ef bezeichnet. Die naheliegende Forderung für ein ideales Netz ist dabei, daß die Werte möglichst klein bleiben. Die Verwendung der Funktionen in Form der ausgeglichenen Beobachtungen zur Untersuchung der äußeren Zuverlässigkeiten ist an sich willkürlich. Der Einfluß δ_{01} schätzt damit die maximale Verfälschung einer Funktion f durch einen Fehler ∇_{01} ab.

$$\delta_{01} \geq \frac{\nabla_{01} f}{\sigma_f} \quad (5-43)$$

Es ist damit ein datumsunabhängiges Maß für die äußere Zuverlässigkeit gefunden. Verschiedentlich wurden alternative Formen der Beschreibung der äußeren Zuverlässigkeit gesucht. So ist in Förstner (1979) ein geometrisch anschauliche Maß zur Charakterisierung der äußeren Zuverlässigkeit in Form der Einflußangabe eines groben Fehlers auf die relative Lage, der durch die Beobachtungen betroffenen Punkte angegeben. In Heckmann (1985) wird ein anderes, ebenfalls auf Förstner zurückgehendes Maß der äußeren Zuverlässigkeit bestimmt. Als punktbezogenes Kriterium für die Beurteilung der äußeren Zuverlässigkeit einer Punktlage wird für jeden Punkt das Produkt aus (5-42) und der großen Halbachse der Fehlerellipse angegeben. Dabei ist natürlich zu berücksichtigen, daß eine Beobachtung sich auf zwei Punkte bezieht. Dieses Maß ist natürlich wieder dem Netzdatum unterworfen (Steinhilber-Förstner, 1985). In der vorliegenden Arbeit werden gemäß einer Empfehlung von Alberda (1980), Bill (1984) u.v.a. folgende Grenzwerte untersucht:

- Redundanzanteile r_1 0,3 - 0,5
- Grenzwerte GWA ∇_{01} (6-8) σ_{11}
- Einflußfaktoren Ef δ_{01} 8-10

Die Bereichsüber- oder unterschreitung wird im Ausdruck der vorliegenden Arbeit durch spezielle Symbolik angemerkt.

5.3 Hauptkomponentenuntersuchung

Dieses Kriterium der Netzqualität gestattet eine globale Aussage über die Genauigkeitsverhältnisse im Netz. Das Verfahren stammt aus der multivariaten Statistik und wird dort häufig angewandt, um charakteristische Eigenschaften höherdimensionaler Kovarianzmatrizen zu untersuchen. Dabei wird gegenüber den bisher besprochenen globalen Genauigkeitskriterien bei der Spektralzerlegung (5-2) auch zumindest ein Eigenvektor berücksichtigt. Insbesondere bei der Untersuchung von Deformationsnetzen wird dieses Verfahren zur Aufdeckung von globalen Schwächezonen eines Netzes angewandt. Wie bereits erwähnt, wird die erste Hauptkomponente bestimmt. Diese entspricht dabei der Funktion der Netzparameter, die von allen denkbaren Funktionen des Parametervektors am schlechtesten bestimmt ist. Die Berechnungsvorschrift für die erste Hauptkomponente lautet:

$$p_1 = s_1 \sigma_0 \sqrt{\Gamma_1} \quad (5-44)$$

Dabei bezeichnet Γ_1 den maximalen Eigenwert und s_1 den zugehörigen normierten Eigenvektor der Spektralzerlegung der Kofaktorenmatrix Q_{xx} .

Soll ein Deformationsnetz untersucht werden, sollten die zu erwartenden Deformationen nicht in Richtung dieser Hauptkomponente weisen. Für ein übliches Verdichtungsnetz im Bereich der Landesvermessung stellt dieses Kriterium keine wesentliche Forderung dar. Lediglich im Bereich von Schwächezonen könnte dieses Kriterium auch für diese Arten und Netztypen interessant werden. Da bei der Berechnung lediglich der größte Eigenwert und der zugehörige normierte Eigenvektor gesucht sind, wäre die Berechnung einer kompletten Spektralanalyse ein etwas hoher Aufwand. Aus dem Bereich der numerischen Mathematik sind iterative Verfahren bekannt, die die Berechnung des maximalen Eigenwertes und des zugehörigen Eigenvektors ermöglichen. Das Iterationsverfahren nach Mises (Engeln-Müllges Reutter, 1987) erfüllt eben diese Forderung.

Die Matrix G sei eine reelle, nicht notwendigerweise symmetrische Matrix mit n linear unabhängigen Eigenvektoren z_1 bis z_n .

$$G_{n,n} \quad e_1, e_1, e_1, \dots, e_n, \in R^n \quad (5-45)$$

Man beginnt mit einem beliebigen reellen Vektor $z^0 \neq 0$ und bildet mit der Matrix F die iterierten Vektoren z^1 nach folgender Vorschrift:

$$z^{i+1} = F z^i \quad z^i = \begin{bmatrix} z_1^i \\ z_2^i \\ \vdots \\ z_n^i \end{bmatrix} \quad (5-46)$$

Der Vektor z^0 lässt sich als Linearkombination der von ihm unabhängigen Eigenvektoren e_i darstellen:

$$z^0 = \sum_{i=1}^n c_i e_i \quad (5-47)$$

wobei mindestens eine der Konstanten c_i von 0 verschieden ist. Für z^i gilt:

$$z^i = c_1 \Gamma_1^i e_1 + c_2 \Gamma_2^i e_2 + \dots + c_n \Gamma_n^i e_n \quad (5-48)$$

Nun werden die Quotienten q_j^i der j -ten Komponente der Vektoren z^{i+1} und z^i gebildet:

$$q_j^i = \frac{z_j^{i+1}}{z_j^i} = \frac{c_1 \Gamma_1^{i+1} x_{1,j} + c_2 \Gamma_2^{i+1} x_{2,j} + \dots + c_n \Gamma_n^{i+1} x_{n,j}}{c_1 \Gamma_1^i x_{1,j} + c_2 \Gamma_2^i x_{2,j} + \dots + c_n \Gamma_n^i x_{n,j}} \quad (5-49)$$

Im weiteren ist eine Fallunterscheidung vorzusehen. Gilt $c_1 \neq 0$ und $x_{1,j} \neq 0$, berechnet sich der Quotient q_j^i zu:

$$q_j^i = \Gamma_1 + 0 \left(\left\| \frac{\Gamma_2}{\Gamma_1} \right\|^i \right) \quad \text{bzw.} \quad \lim_{i \rightarrow \infty} q_j^i = \Gamma_1 = \Gamma_{\max} \quad (5-50)$$

Die obige Voraussetzung ist mindestens für eine der Größen $x_{1,j}$ erfüllt und es strebt damit mindestens einer der Quotienten q_j^i gegen e_1 . Für eine genügend große Iterationsanzahl stellt der Quotient q_j^i eine Näherung für den betragsgrößten Eigenwert Γ_1 dar. Für eine

große Anzahl von Iterationen nähert sich die Richtung von z^4 annähernd der Richtung von e_1 asymptotisch an.

Alternativ dazu ist im Fall das $c_1 = 0$ oder $x_{1,j} = 0$, jedoch $c_2 \neq 0$ und $x_{2,j} \neq 0$ ist. Der Fall $c_1 = 0$ tritt ein, wenn der Ausgangsvektor z^0 keine Komponente in Richtung von e_1 besitzt. Im Falle symmetrischer Matrizen ist $c_1 = 0$, wenn z^0 orthogonal zu e_1 ist.

Bei dem Verfahren sollte daher durch Vorgabe verschiedener Startlösungen die Iteration überprüft werden, um Konvergenzschwierigkeiten im zweiten Fall zu vermeiden. Schwierigkeiten sind auch zu erwarten, wenn der maximale Eigenwert mehrfach auftritt. Da dies aber für die Anwendung im Bereich der Hauptkomponentenanalyse nicht auftreten sollte, kann der Fall ausgeklammert werden.

Die Aussagekraft der Hauptkomponentenanalyse hängt von der Dominanz des größten Eigenwertes ab. Hat er großen Anteil, lassen sich mit diesem Kriterium aussagekräftige Netzschwachstellen nachweisen. Eine Darstellung des Eigenwertspektrums (siehe Abschnitt 8.1) zeigt dies deutlich auf. Die Hauptkomponentenanalyse stellt ein datumsabhängiges Kriterium dar. Laut Niemeier (1985) zeigt jedoch der wesentliche Eigenvektor vorhandene Schwächezonen eines Netzes in den verschiedensten Datumsfestlegungen in weitgehend gleicher Weise auf, sodaß trotzdem von einem universell einsetzbarem Kriterium gesprochen werden kann.

5.4 Lokale Einzelpunktuntersuchung

Bei diesem Verfahren steht, wie schon der Name sagt, die Beurteilung der Netzqualität im Einzelpunkt im Vordergrund. Zum ersten sollen damit Konfigurationsdefekte, wie sie in kombinierten Netzen auftreten können effektiv überprüft werden. So können etwa ganze Punktgruppen innerhalb eines Netzes "schwimmen" oder auf Grund mangelnder Konfigurationsqualität unsicher gelagert sein.

Jeder Punkt wird im Zuge einer Einzelpunktausgleichung einer Analyse unterzogen. Alle Beobachtungen, die den Punkt als Stand oder Zielpunkt betreffen, werden aus dem Gesamtbeobachtungsmaterial herausgefiltert. Sodann wird ein Normalgleichungssystem der Dimension 2 (Koordinatenunbekannte) bzw. 3 (ev. zusätzliche Orientierungsunbekannte) aufgebaut. Maßstabsunbekannte werden in

diesem Zusammenhang nicht berücksichtigt, da häufig nur wenige Streckenbeobachtungen zu einem Punkt verlaufen. Nach Bildung des lokalen Normalgleichungssystems wird die Normalgleichungsmatrix invertiert und damit die lokalen Genauigkeitsmaße lt. Abschnitt 5.1.2 berechnet. Neben den Elementen der Helmert'schen Punktfehlerellipse wird außerdem noch mittels einer Spektralzerlegung die Kondition überprüft und für jeden Punkt ausgewiesen. Damit ist die lokale Bestimmbarkeit des Punktes aus den ihn betreffenden Beobachtungen überprüfbar.

Der Vergleich mit den Werten aus der Gesamtausgleichung zeigt die Bedeutung der restlichen Beobachtungen für die Bestimmbarkeit des jeweiligen Einzelpunktes.

Die berechneten Unbekannten (als Zuschläge an die aus der Gesamtausgleichung gewonnenen Werte) geben Auskunft über eventuelle Punktverschiebungen auf Grund der lokalen, zum Punkt gehörigen Beobachtungen. Der Erwartungswert dieser Zuschläge für Koordinaten und Orientierungsunbekannte ist 0.

Darüberhinaus wird innerhalb einer Einzelpunktausgleichung nach und nach jeweils eine Beobachtung eliminiert. Der Varianzfaktor der Gewichtseinheit sowie die Koordinatenzuschläge werden auch in diesem Fall berechnet. Für jede Beobachtung eines Punktes wird ein getrennter Lauf durchgeführt. Ein Vergleich mit den Werten aller Beobachtungen eines Einzelpunktes zeigt interessante Abweichungen. Insbesondere in Fällen schlechter Zuverlässigkeitskennzahlen treten dabei deutliche Effekte in den Unbekanntenzuschlägen auf und gestatten damit eine geometrische Interpretation von Zuverlässigkeitsmaßen in Wirkung auf die Unbekannten. Deutlich zeigen sich auch grobe Fehler. Für solche Beobachtungen sinkt der Varianzfaktor der Gewichtseinheit unter den Wert, der sich aus Gesamteinzelpunktuntersuchung ergibt. Damit kann, falls der Beobachtungswert auf Grund der Beobachtungsprotokolle nicht eliminiert werden soll, die Wirkung auf die jeweiligen Einzelpunktkoordinaten bzw. Orientierungen untersucht werden. Zu beachten ist dabei jedoch, daß eine Beobachtung natürlich zumindest auf zwei Punkte rückwirkt, sodaß auch die Einzelpunktuntersuchung des zweiten beteiligten Punktes herangezogen werden muß.

Als zusätzlicher Parameter kann mittels der Einzelpunktuntersuchung die Verteilung der lokalen Standardabweichung der Gewichtseinheit überprüft werden. Zu dem Zweck wird gemäß der in Abschnitt 4.4 beschriebenen Varianzkomponentenüberprüfung eine Gruppeneinteilung dergestalt vorgenommen, daß alle Beobachtungen eines Punktes als Gruppe aufgefaßt werden. Mit der in Abschnitt 4.4 beschriebenen Methodik wird nun die Varianzkomponente dieser Beobachtungsgruppe bestimmt. Diese Vorgangsweise wird ebenfalls von Ehlert (1984) in der Diagnoseausgleichung 1980 des DHDN durchgeführt. Es zeigt sich auch hier, daß insbesondere auf Punkten mit gehäuft auftretenden wenig zuverlässigen Beobachtungen dieser Wert ansteigt. Ist ein Punkt nur von wenigen Beobachtungen bestimmt worden, darf die Aussagekraft dieser Kenngröße nicht überschätzt werden, da sich insbesondere auf solchen Punkten häufig besonders kleine lokale Standardabweichungen der Gewichtseinheit l_{m_0} ergeben können.

6. a priori Untersuchung durch Simulationsausgleichung

In diesem Kapitel soll die Fragestellung, welche Qualitätskriterien eines Netzes können vorab beurteilt werden, behandelt werden. Bei der Anlage bzw. Erweiterung geodätischer Netze sind bestimmte Zielvorstellungen zu erfüllen. Diese sollen durch das Netz möglichst gut angenähert bzw. erfüllt werden. Dabei können diese Forderungen einzelne oder eine Kombination aus verschiedenen Qualitätskriterien, wie sie im Abschnitt 5 erwähnt wurden, darstellen. Für Verdichtungsnetze, wie sie in dieser Arbeit untersucht werden sollen, sind dabei in erster Linie Isotropie und Homogenität der Fehlerellipsen anzustreben. Daneben sollen die Zuverlässigkeitskennzahlen die angegebenen Grenzwerte nicht überschreiten. In anderen Fällen, etwa Deformationsnetzen, wird die Zielvorstellung im allgemeinen anders definiert. In diesem Fall soll in Bewegungsrichtung - möglichst nach kurzer Wiederholungszeit (= Deformation) - der Effekt statistisch einwandfrei nachweisbar sein. Die isotrope Fehlerinformation ist in diesem Fall nicht anzustreben. Die Vorgangsweise der a priori - Simulationsausgleichung, gestattet eine Vorabuntersuchung von Qualitätskriterien. Durch Vorgabe der Konfiguration (A-Matrix) und der Festsetzung der Meßgenauigkeit ($P = Q_{11}^{-1}$ -Matrix), mit der die Beobachtungen im Netz durchgeführt werden sollen (abhängig von den geplanten Gerätegenauigkeiten), lassen sich die Funktionswerte der vorgegebenen Zielfunktion (-en) berechnen. Solch einer Berechnung geht die Erkundung der sichtbaren Visuren im Gelände voraus. Danach kann entschieden werden, ob der den Berechnungen zugrunde gelegte Entwurf den Anforderungen genügt, oder ob gewisse Modifikationen notwendig sind. Nach Veränderung der A - Matrix oder der P - Matrix ist eine erneute Durchrechnung und Kontrolle der Auswirkungen auf die Zielfunktion (-en) zu untersuchen. Diese Vorgangsweise stellt ein systematisches Probiervverfahren dar, bei dem einzelne diskrete Werte der vorgegebenen Zielfunktion berechnet werden. Die Erreichung des mathematischen Optimums der Zielfunktion ist lediglich durch Anwendung strenger Optimierungsstrategien möglich. Die Formulierung der Zielfunktionen für geodätische Netzoptimierungen sind aus den Kriterien der Genauigkeit, Zuverlässigkeit und versuchsweise auch der Wirtschaftlichkeit anzusetzen.

Eine Gliederung der Gesamtproblematik der Optimierung geodätischer Netze geht auf Grafarend zurück. Er gliedert die Problemstellung in Designs 0. bis 3. Ordnung:

Design 0. Ordnung:

Die Fragestellung beinhaltet das Datumsproblem bei der Ausgleichung freier Netze. Die Konfiguration des Netzes und das Gewichtsverhältnis der Beobachtungen werden dabei als festgesetzt betrachtet. Für die zu schätzenden Parameter und für deren Genauigkeitsverhältnisse (Q_{xx} Kofaktorenmatrix) ist das günstigste Bezugssystem zu wählen.

Design 1. Ordnung:

Hier geht es um die Suche nach einer optimalen Konfiguration für das zu entwerfende Netz. Die Konfiguration eines Netzes wird ja gemäß (2-9) durch die Design-Matrix A , die den geometrischen Zusammenhang zwischen den Beobachtungen und den Unbekannten in linearer Form vermittelt, dargestellt.

Design 2. Ordnung:

Beim sogenannten Gewichtsproblem der geodätischen Netzoptimierung besteht die Frage nach optimalen Beobachtungsgewichten bzw. Gerätegenauigkeiten. Für eine vorgegebene feste Konfiguration sind die Genauigkeiten bzw. die Wiederholungszahlen der im Netz durchzuführenden Beobachtungen festzulegen, sodaß das Netz den gestellten Forderungen bestmöglichst genügt. Diese Forderungen werden dabei in einer vorgegebenen Kofaktorenmatrix zusammengefaßt. Die Einführung der sogenannten Wiederholungszahlen im Bereich der P - Matrix spiegelt die Tatsache wieder, daß Gewichte bzw. Beobachtungsgenauigkeiten nicht stufenlos verändert werden können. Lediglich eine Mehrfachbeobachtung mit dem selben Gerät und damit gleicher Beobachtungsgrundgenauigkeit sind möglich. Eine große Zahl von Publikationen beschäftigt sich mit dieser Fragestellung unter Auffindung optimaler Wiederholungszahlen. Ein instruktives Beispiel zu dieser sogenannten sequentiellen Gewichtsoptimierung findet sich bei Augath (1985) für ein Streckennetz unter Berücksichtigung verschiedener Zielfunktionen. Dabei wird gezeigt, daß Optimierung bezüglich Genauigkeitskriterien mit einer größeren Streuung von Wiederholungszahlen verbunden ist als dies für Zuverlässigkeitskriterien gilt.

Wie erwähnt, bezieht sich dieses Beispiel auf Streckenbeobachtungen. Die Problematik dieser Vorgangsweise liegt in der Tatsache, daß dabei die Beobachtungen als unkorreliert gelten. Solche unabhängigen Mehrfachbeobachtungen (verschiedene Meßtage) widersprechen sicher eventuell mitzuführenden Wirtschaftlichkeitsüberlegungen. Eine weitere Problematik tritt bei Einbeziehung von Richtungsbeobachtungen auf. Entweder werden die Wiederholungszahlen gleichzeitig für alle Ziele eines Richtungssatzes eingeführt, oder einzelne Richtungen innerhalb eines Satzes können verschiedene Wiederholungszahlen und damit verschiedene Bedeutungen innerhalb des Richtungssatzes gewinnen. Dies widerspricht aber eigentlich der Strategie der Richtungssatzmessung. In Hanke, Grimm - Pitzinger (1987) werden daher lediglich die Wiederholungszahlen 0 und 1 zugelassen. Dies hat zur Konsequenz, daß eine Beobachtung entweder durchzuführen, oder fortzulassen ist. Diese Vorgangsweise stellt meines Erachtens bezüglich der späteren geodätischen Beobachtungsmethodik den ehrlichsten Ansatz dar.

Design 3. Ordnung

Die Aufgabenstellung dabei ist definiert als die optimale Verbesserung bzw. Erweiterung eines bereits bestehenden oder geplanten Netzes durch Einschaltung zusätzlicher Neupunkte. Solche Probleme treten auf, wenn es darum geht, vorhandene Netze zu verdichten oder etwa in Ingenieur-Projekten wenn das bestehende Netz in geeigneter Weise zu erweitern ist oder Punktverlagerungen durchzuführen sind. Dabei bedeutet die strenge Lösung des Designs 3. Ordnung die Konfigurationsoptimierung der Ergänzungsbeobachtungen. Es beantwortet also damit gleichzeitig die Fragestellung des Designs 1. und 2. Ordnung bezogen auf den Netzentwurf für die zusätzlichen Beobachtungen. Eine sehr umfassende Darstellung des Designs 3. Ordnung gibt Schmid u.a. (1985). Eine interessante und anschauliche Vorgangsweise beschreibt Illner (1986) zur gleichzeitigen Konfigurations- und Gewichtsoptimierung, wenn mit den Ergänzungsbeobachtungen Neupunkt-Einschaltungen geplant sind. Unter Berücksichtigung der Tatsache, daß durch die Topographie der Variationsbereich für die Lage der Neupunkte in der Regel stark eingeschränkt wird und somit nur kleine Lageänderungen möglich sind, werden die verschiedenen Möglichkeiten der Punktlage durch die Vorgabe eines

Rasters mit variabler Maschenweite und variabler Anzahl von Knoten simuliert. Für die Entscheidungsfindung der optimalen Punktlage dienen verschiedene skalare Zielfunktionen, deren aktueller Wert in den diskutierten Rasterpunkten miteinander verglichen werden kann.

Schon in der Frühzeit der Anwendung der Optimierung im Bereich der Geodäsie wurde neben ausschließlich skalaren Zielfunktionen ein allgemeineres Konzept, basierend auf der Einführung einer sogenannten Kriteriummatrix als idealisierte Varianz-Kovarianzmatrix, eingeführt. Ziel der Optimierung war dabei, diese Matrix bestmöglich durch das zu entwerfende Netz anzunähern; Grafarend (1972), Baarda (1973). Eine aktuelle Zusammenschau der geodätischen Netzoptimierung ist in Graferend, Sanso (1985) enthalten. Die nachfolgende Tabelle 6.1 soll nochmals die Gliederung des Designs geodätischer Netze zusammenfassen:

Ordnung	festgeh. Parameter	freie Parameter	Zielfunktion
0	A, P	\mathbf{x} , $Q_{\mathbf{x}\mathbf{x}}$	optimale Datumsvorg.
1	P, $Q_{\mathbf{x}\mathbf{x}}$	A	Konfigurationsopti.
2	A, $Q_{\mathbf{x}\mathbf{x}}$	P	Gewichtsoptimierung
3	$Q_{\mathbf{x}\mathbf{x}}$	A, P teilw.	optimale Erweiterung

Tab.6.1

Bei der Formulierung von Qualitätsmaßstäben in Überwachungsnetzen leisten Kriteriummatrizen große Hilfe. Sie sind, wie erwähnt, als ideale Matrizen anzusehen; wobei hier "ideal" optimal angepaßt auf die Situation des geplanten Netzes bedeutet. Der Hauptvorteil der Anwendung der Kriteriummatrizen ist deren Unabhängigkeit vom Beobachtungsplan. Sie sind nicht nur für die absolute Koordinatenfehler verwendbar, sondern auch für alle Arten von abgeleiteten Größen, wie Distanzen, Azimute, Winkel und beliebigen Kombinationen.

In der Praxis wird von einigen Autoren (Niemeier, 1985 u.a.) vor der Einführung automatisierter Verfahren bei der sequentiellen Optimierung gewarnt. Vielmehr sollte eine interaktive Vorgangsweise jeden Schritt der Optimierung nachvollziehbar machen. Im speziellen

können dabei wirtschaftliche Aspekte, welche sich nur sehr schwer in Funktionen formulieren lassen, berücksichtigt werden. So kann etwa das Fortlassen aller Beobachtungen auf einer Station bedeutende wirtschaftliche Auswirkungen haben, wobei die Station als Zielpunkt weiterhin im Netz belassen werden kann, dann jedoch nur zur Signalerrichtung einmal begangen werden muß.

6.1 Planung von Netzverdichtungen

Gemäß dem Design-Konzept nach Grafarend ist dieses Problem einer Optimierungsaufgabe gemäß dem Design 3. Ordnung zuordenbar. Auf Grund der topographischen Verhältnisse im Bereich von in dieser Arbeit im besonderen untersuchten Netzen der Landesvermessung ist die Punktlage von zusätzlichen Punkten zumeist in wenigen Varianten gegeben. Damit ist eine diskrete Untersuchung gerechtfertigt. Bei der a priori-Berechnung können die im Abschnitt 5 erwähnten Genauigkeits- und Zuverlässigkeitskennzahlen berechnet werden und dabei ersetzt die Varianz der Gewichtseinheit a priori die Größe (2-19) und wird in alle Fehlermaßen eingerechnet. Diese Größe stellt also quasi einen Maßstab der Genauigkeitssituation dar. Sollte sich bei der späteren Ausgleichung der Beobachtungen ein anderer Wert für die Varianz der Gewichtseinheit ergeben, ändern sich alle Genauigkeitsgrößen im selben Verhältnis. Für die Zuverlässigkeitskennzahlen (5-38, 5-42), basierend auf den Redundanzanteil r_1 (5-36), lassen sich keine Aussagen machen; wengleich Testrechnungen gezeigt haben, daß die Empfindlichkeit gegenüber kleinen Veränderungen des Varianzniveaus vor und nach der Ausgleichung auf die Zuverlässigkeitskennzahlen geringe Rückwirkungen haben. Auch die Hauptkomponentenuntersuchung (Abschnitt 5.3) und die lokale Einzelpunktuntersuchung (Abschnitt 5.4) läßt sich a priori durchführen. Im Falle der Netzverdichtung bei bereits vorgegebenen und ausgeglichenen geodätischem Netz ist die Stochastik innerhalb des bereits bestehenden Netzteiles bekannt. Diese kann natürlich in der Simulationsausgleichung des zu erweiternden Netzteiles einbezogen werden. Wengleich diese Vorgangsweise im Bereich von Netzen der Landesvermessung derzeit nicht durchgeführt wird, sollte sie zumindest bei der Analyse von Netzerweiterungen Berücksichtigung finden. Die Vorgangsweise wurde bereits im

Abschnitt 3 gegeben. Da zu einer dynamischen Datumsvorgabe die Kovarianzmatrix der Erstausgleichung erforderlich ist, kann hier nur die Datumsvorgabe mittels Ersatzbeobachtungen für die Netzerweiterung angewandt werden. Dabei könnten die Ersatzbeobachtungen die Stochastik der Altpunkte für die Netzerweiterung beschreiben und auf diese Weise realistische Abschätzungen der Punktgenauigkeiten eventuell später folgender Gesamtausgleichung wiedergeben.

6.2 Abschätzung beliebiger Beobachtungskombinationen

Ein weiterer Vorteil der a priori - Analyse von geodätischen Netze bzw. deren Erweiterungen liegt in der Möglichkeit der Untersuchung von Kombinationen beliebiger Beobachtungsarten zu Punktstellung. Insbesondere die Erweiterung um beobachtete Koordinatendifferenzen eröffnet hierbei ein weites Spektrum von Anwendungen. So könnten etwa Koordinatendifferenzen in direkter Weise bei Industriemeßsystemem in Kombination mit kleinräumigen geodätischen hochpräzisen Netzen auftreten.

Andererseits stellen moderne Satellitenbeobachtungsverfahren (GPS) nach derzeitigem Forschungsstand in erster Linie Koordinatendifferenzen als Beobachtungen zur Verfügung. Diese Satellitenbeobachtungen sind jedoch in einem weltweit eindeutig gegebenen Koordinatensystem (WGS 84) durchgeführt worden und müssen in das aktuelle Berechnungssystem transformiert werden. Auf Grund dieser Transformation und den Bahndaten der Satelliten sind diese dreidimensionalen Koordinatendifferenzen korreliert. Diese Korrelation müßte eigentlich in die Berechnung aufgenommen werden. Für die Vorausabschätzung der Kombination und von Satelliten beobachteten Koordinatendifferenzen und klassisch konventionellen Beobachtungsformen der Geodäsie können jedoch unkorrelierte Ersatzbeobachtungen Verwendung finden, wobei die Schwachstelle der vernachlässigten Korrelation durch eine Anhebung des Varianzniveaus der Koordinatendifferenzen ausgeglichen werden sollten. Eine aktuelle Frage dazu ist, inwieweit bei technischen Projekten (Tunnelnetze) der obertägige Netzanteil durch GPS - Beobachtungen ersetzt werden kann und die bestehenden Netze der Landesvermessung hier eine Kontrolle und Überprüfbarkeit der GPS - Beobachtungen ermöglichen.

Zur gemeinschaftlichen Ausgleichung von terrestrischen Beobachtungen und Satellitenbeobachtungen werden ähnliche Vorgangsweisen mit teilweise erweiterten Modellbildungen angewandt. (Niemeier 1987) Dabei werden für die beobachteten Koordinatendifferenzen eine gemeinsame Orientierungsunbekannte als auch eine Maßstabsunbekannte in die Modellbildung aufgenommen. (Fritzensmeier u.a. 1988)

7. Realisierung einer umfassenden Analyseausgleichung

Die in den vorangegangenen Abschnitten beschriebenen Verfahren wurden in einem Pascal-Programm ANAG (ANalyseAusGleichung) in einem System zusammengefaßt. Die Datenorganisation wurde projektbezogen organisiert. Gemäß der Zielvorgabe in Abschnitt 1 wurde der Benutzerinteraktion während des Programmlaufes besonderes Augenmerk geschenkt. Zur Realisierung einer allgemein verständlichen Benutzerschnittstelle wurde ein dialogorientierter Maskenaufbau gewählt. Dabei konnte bezüglich des Maskenkonzeptes auch eine bewährte Lösung in Ebenhoch (1986) zurückgegriffen werden. Die Implementierung der Algorithmen sollte für einen IBM - PC geeignet erfolgen. Auf Grund der hohen Anforderungen bezüglich der Datenstruktur bei Netzausgleichungsprogrammen fiel die Wahl der Programmiersprache auf Pascal. Diese Wahl wird auch von Wirth (1984) bestätigt. Die von Jensen und Wirth (1975) formulierte Programmiersprache Pascal fand in ihrer Anfangszeit in erster Linie im Bereich der Lehre große Verbreitung. Dies mag mit der ausgezeichneten Strukturierbarkeit des Programmablaufes in Verbindung stehen. Auf den damals üblichen Großrechenanlagen wurde der USCD-Pascalstandard erarbeitet. Bei Aufkommen der Personalcomputer wurden auch für diese entsprechende Compiler erstellt. Solch ein besonders leistungsfähiger Pascal-Compiler liegt in Form des Turbo-Pascal 3.X vor. In diesem Dienstprogramm wurden einige Pascal Schwachstellen (Datenmanipulation) durch entsprechend mächtige Befehle erweitert. Ein weiterer Vorteil dieser Programmiersprache ist das Portabilitätsvermögen zwischen verschiedenen Rechnertypen, mit Ausnahme der Zusatzbefehle. Zusätzlich stellt Pascal mächtige Datentypen, insbesondere dynamische Zeigervariablen, zur Verfügung. Letztere konnten im vorliegenden Programm verwendet werden, um eine dynamische Speicherverwaltung im Bereich der Normalgleichungsmatrix bzw. deren Inversen zu realisieren. Die Designmatrix A wurde in indizierter Form organisiert. Dabei werden lediglich die in der einzelnen Verbesserungsgleichung auftretenden Elemente der A - Matrix verwendet. Dabei treten also je Verbesserungsgleichung maximal 3 Koeffizienten auf (2 für Koordinaten und 1 für Orientierung bzw. Maßstab). Bei der Berechnung des Matrizenproduktes $A^T P A$ wird lediglich mit Hilfe der Indices das entsprechende Element bestimmt.

Eine analoge Vorgangsweise wird bei der Berechnung des Matrizenproduktes $AQ_{xxx}A^T$ bzw. $f^T Q_{xxx} f$ verwendet. Für die Inversion wurden eine Reihe von Verfahren getestet und verglichen. Im speziellen wurden dabei das klassische Gauss-Verfahren, Gauss-Jordan-Näherungsverfahren, Eskalator-Methode (Zurmühl Falk, 1986) sowie Orthogonalisierungs-Methoden (Lawson Hanson, 1974) verglichen. Bei einer Reihe von Testbeispielen ergab sich für das Cholesky-Verfahren eine gleichartige, wenn nicht geringfügig höhere Bearbeitungsgeschwindigkeit. Hinsichtlich der Rechenschärfe zeigte sich das Verfahren in manchen Fällen überlegen. Alternativ dazu wurden Versuche mit der Totalinversion der A - Matrix mittels Singulärwertzerlegung durchgeführt. (Forsthuber, Schneid, Stanek; 1987). Auf die Verwendung sogenannter sparse-Matrizentechniken bzw. Anwendung von Hash-Algorithmen wurde hier im Hinblick auf die angestrebte hohe Bearbeitungsgeschwindigkeit zu Lasten erhöhten Speicherbedarfs für die Berechnung der Indizierung der N - Matrix verzichtet.

Die Berechnung der Eigenwerte wird nach dem Jacobi - Iterationsverfahren durchgeführt. Dieses wurde bereits im Abschnitt 5. beschrieben. Die vorliegende Realisierung wurde aus Carusio (1984) entnommen. Für die Berechnung des maximalen Eigenwertes und des zugehörigen Eigenvektors der N-Matrix wurde wie beschrieben das Iterationsverfahren nach Mises (Abschnitt 5.3) übernommen.

Die Kapazitätsgrenzen der derzeit vorliegenden Programmversion lauten:

Anzahl der Beobachtungen :	450
Anzahl der Unbekannten :	220

Überlegungen zur Rechenschärfe:

Die vorliegende Programmversion wird mit dem Compiler TURBO-87 bearbeitet, der die Präsenz eines Mathematik-Coprozessors 80X87 voraussetzt. Damit wird einerseits eine erhöhte Ausführungsgeschwindigkeit erreicht und zum anderen werden besondere Datenformate für Gleitkommazahlen zur Verfügung gestellt. Dem Real-Datenformat (8 Bit) entsprechen im Dezimalsystem etwa 16 gültige Dezimalstellen.

Die Konditionszahl der Matrix N berechnet sich:

$$\text{cond}(N) = \| N \| \| N^{-1} \| \quad (7-1)$$

Die Berechnung der Matrixnormen kann entweder über die vereinfachte Berechnung der Maximumnorm (Gesamtnorm)

$$\| N \| = u * \max_{i,k} \| n_{i,k} \| \quad i,k : 1 \dots u \quad (7-2)$$

oder mit Hilfe der aufwendiger zu bestimmenden Spektralnorm angegeben werden:

$$\| N \| = \Gamma(N)_{\max} \quad (7-3)$$

Im letzteren Fall berechnet sich dann die Kondition der Matrix N nach der Spektralnorm

$$\text{cond}(N) = \| N \| \| N^{-1} \| = \frac{\Gamma(N)_{\max}}{\Gamma(N)_{\min}} \quad (7-4)$$

$$N x + y = 0 \quad (7-5)$$

Für die Matrix N und die Vektoren x und y sollen kleine Störungsanteile zugelassen werden. Die Beeinflussung dieser Störungsanteile auf die jeweiligen anderen Größen läßt sich mit Hilfe der Kondition aus Schwarz (1986) wie folgt abschätzen:

$$\frac{\| \Delta x \|}{\| x \|} \leq \frac{\text{cond}(N)}{1 - \text{cond}(N) \frac{\| \Delta N \|}{\| N \|}} \left[\frac{\| \Delta N \|}{\| N \|} + \frac{\| \Delta y \|}{\| y \|} \right] \quad (7-6)$$

Dabei ist zu beachten, daß bei der Abschätzung eine bei der Berechnung der Matrixnorm N verträgliche Vektornorm für x bzw. y anzuwenden ist. Zur Spektralnorm (7-3) stellt die euklidische Vektornorm eine verträgliche Form dar.

$$\| x \|_2 = \sqrt{\sum_{i=1}^u x_i^2} \quad (7-7)$$

Eine Abschätzung nach Schwarz (1986) besagt:

Wird ein lineares Gleichungssystem (7-5) mit d-stelliger dezimaler Gleitkommarechnung gelöst und beträgt die Konditionszahl $\text{cond}(N)$ ungefähr 10^a , so sind auf Grund der allgemeinen, unvermeidlichen Eingangsfehler in der berechneten Lösung x bezogen auf die betragsgrößte Komponente nur auf $(d - a - 1)$ Dezimalstellen sicher.

Diese Abschätzung liefert im allgemeinen zu pessimistische Aussagen. Wesentlich jedoch ist dabei, daß sich die Abschätzung nur auf die betragsgrößte Komponente des Lösungsvektors bezieht. Dies impliziert die Forderung, daß die Komponenten des Lösungsvektors alle von selber Größenordnung sein sollten. Im vorliegenden Programmsystem wird dies durch Skalierung des gekürzten Beobachtungsvektors l erreicht. Die Beobachtungen werden in mm bzw. cm behandelt, sodaß auch der Lösungsvektor x sich ebenfalls in diesen Dimensionen ergibt. Tritt also in der vorliegenden Berechnungsform eine Kondition von etwa $1 \cdot 10^6$ auf, ergibt sich unter der Annahme der 16 Dezimalstellen eine dezimale Stellenanzahl für das betragsgrößte Element des Unbekanntenvektors von 9. Dies erscheint auf Grund der angeführten Skalierung sicherlich ausreichend.

Alle Informationen eines Projektes werden in einer Bearbeitungsdatei gesammelt abgelegt. Diese Informationen werden dabei als Alpha - ASCII - Zeichenfolgen abgelegt und dienen dem geübten Benutzer direkt als Informationsquelle. Dies hat neben der direkten Lesbarkeit zum Vorteil, daß die Übernahme von Informationen durch andere Programme leicht realisierbar ist. Auf diese Weise wurden Schnittstellen vom Ausgleichungsprogramm NETZ-3D realisiert. Dieses klassische Netz-Ausgleichungsprogramm nach Elmiger (1972) wurde im Zuge einer Diplomarbeit (Zalka, 1988) am Institut für Landesvermessung und Ingenieurgeodäsie ebenfalls in die Programmiersprache Pascal übertragen. Die nunmehr realisierte Schnittstelle gestattet die Übernahme von NETZ-3D - Ergebnisfiles in das Programm ANAG. Auf diese Weise konnte innerhalb des Programms ANAG auf sämtliche Reduktionsberechnungen der Beobachtungswerte verzichtet werden. Sollen Rohbeobachtungsdaten für ANAG erfaßt werden, geschieht dies nicht

direkt im Erfassungsteil dieses Programmes, sondern im ebenfalls dialogorientierten neu erstellten Erfassungsprogrammteil des NETZ-3D. Ein Überleitungsprogramm gestattet dann die Erstellung des entsprechenden Bearbeitungsfiles für ANAG. Auf diese Weise wurde das Programmbeispiel EP-Pottendorf in Abschnitt 8.2 erstellt. In analoger Weise wurden Datenschnittstellen zu anderen Netzausgleichungsprogrammen (etwa für EP-Brand-Nagelberg, Abschnitt 8.4) realisiert. Neben der Erstellung von Ergebnisausdrucken sollte eine graphische Aufbereitung bezüglich der Genauigkeitssituation von geodätischen Netzen möglich sein. Dazu wurde von Gold (1987) ein Plotterprogramm entwickelt, das die Darstellung von Punktlage, Beobachtungsplan und Fehlerinformationen in anschaulicher Weise ermöglicht. Die nachfolgende Abb. 7.1 zeigt die Zusammenhänge in graphischer Form.

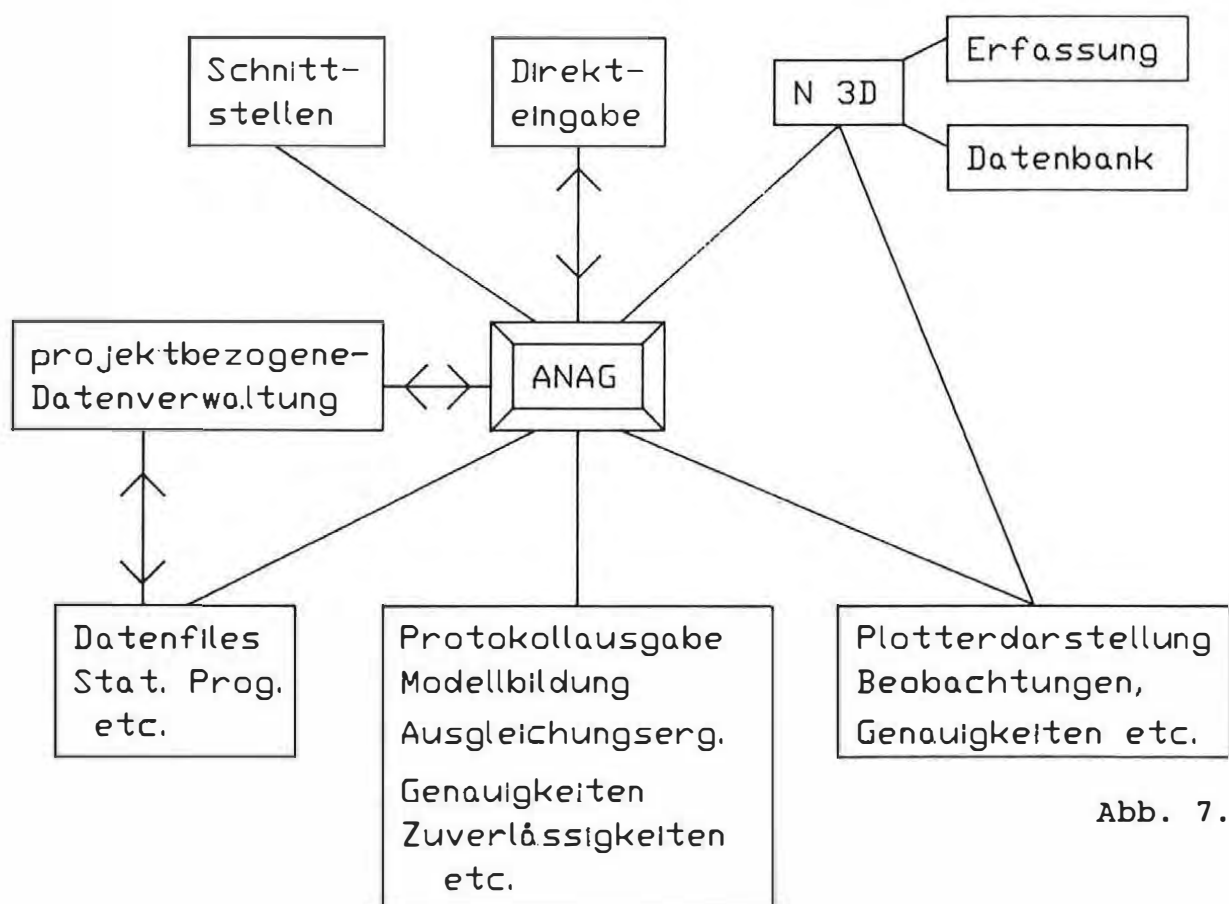


Abb. 7.1

7.1 Programmablauf

In diesem Abschnitt soll in groben Zügen die Abfolge der Bearbeitungsmöglichkeiten aufgezeigt werden.

Mit dem Programmstart wird nach der Berechtigungsüberprüfung entweder eine bestehende Bearbeitungsdatei eingelesen oder eine neue Bearbeitungsdatei mit einzugebenden Namen und häufig verwendeten Default - Werten angelegt.

Der folgende Menüblock beinhaltet folgende Möglichkeiten:

- Steuerdaten
- mittlere Meßfehler
- Punkte
- Beobachtungen
- Gewichte anpassen

- weiter

Steuerdaten:

Hierzu gehören globale, das ganze Operat betreffende Informationen. Zum ersten wird der Name des Projektes angegeben bzw. modifiziert. Wird dabei der Name einer bestehenden Bearbeitungsdatei modifiziert, werden die nachfolgenden Berechnungen unter dem neuen Namen abgelegt. Auf diese Weise bleiben die Originaldaten unter dem alten Projektnamen erhalten. Ein weiterer Punkt der Steuerdaten bezeichnet die Art der Gewichtung innerhalb des Projektes. Dabei steht die Gewichtung gemäß Bundesamt und die Gewichtung gemäß der Vorgabe im Programmpunkt "mittlere Meßfehler" zur Wahl. Weiters ist innerhalb der Steuerdaten anzugeben, ob für die Streckenbeobachtungen eine Maßstabsunbekannte angesetzt werden soll oder nicht. Letztendlich kann innerhalb der Steuerdaten ein Kommentar zum Projekt eingegeben werden.

Der nächste Menüpunkt umfaßt die Angaben der mittleren Meßfehler zur Beschreibung der stochastischen Modellbildung:

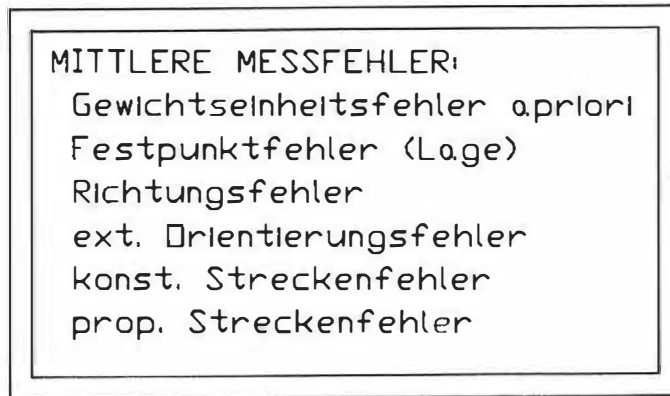


Abb. 7.2

Die Genauigkeitsmaße werden in mm bzw. ∞ angegeben. Relative Genauigkeitsmaße werden in ppm angegeben. Wurde die Gewichtung gemäß der Vorgabe seitens des Bundesamtes für Eich- und Vermessungswesen gewählt, sind lediglich die ersten beiden und die letzten beiden Genauigkeitsmaße wählbar.

Abb. 7.3 zeigt das Bildschirmmenü zur Eingabe und Manipulation von Punkten. Zu beachten ist dabei, daß im Falle "Festpunkte bzw. Neupunkte ändern", diese mit einem Befehl jeweils zum Neupunkt bzw. Festpunkt umgewandelt werden können. Diese Manipulationsform ist insbesondere bei der Analyse von Netzen effektiv anzuwenden.

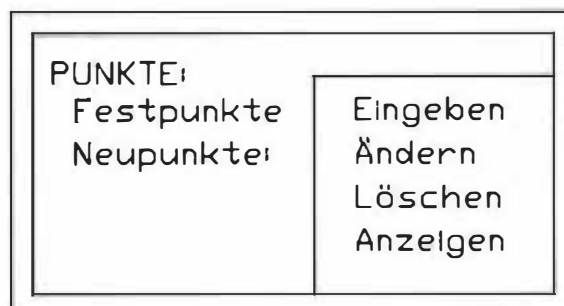


Abb.7.3

Abb. 7.4 zeigt das Bildschirmmenü zur Erfassung bzw. Manipulation von Beobachtungen. Zu beachten ist die Möglichkeit der Manipulation

einer gesamten Station. Dies ist insbesondere für Richtungsbeobachtungen eine häufig notwendige Option. Für jede Beobachtung kann ein individuelles Genauigkeitsmaß eingegeben werden, wobei der im Menüpunkt "mittlere Meßfehler" angegebene Wert als Vorschlagswert übernommen werden kann.

BEOBACHTUNGEN:	
Richtungen	Eingeben
Strecken	Ändern
x - Differenzen	Löschen
y - Differenzen	Station löschen
	Anzeigen

Abb. 7.4

Der Programmpunkt "Gewichte anpassen" ermöglicht die globale Veränderung der Genauigkeiten aller Beobachtungen gemäß der Vorgabe im Programmpunkt "mittlere Meßfehler". Dies ist insbesondere für die Varianzkomponentenüberprüfungen gem. Abschnitt 4.4 von Bedeutung.

Der letzte Punkt "weiter" startet die Überprüfung des Gesamtproblememes. Dabei wird vor der eigentlichen Ausgleichung eine logische Überprüfung vorgenommen. So werden etwa unbestimmte Punkte auf Grund zu weniger Beobachtungen oder voraussichtlich zu großes Normalgleichungssystem etc. mit Fehlermeldungen ausgewiesen und der Programmablauf zur Fehlerkorrektur abgebrochen. Treten keine Fehler auf, wird das Ausgleichungsproblem aufgelöst. Tritt eine Normalgleichungsmatrix mit Rangdefekt auf, wird der Benutzer aufgefordert, das voraussichtliche Rangdefizit anzugeben. Im Falle einer freien Netzausgleichung ist $d = 3$ oder 4 zu wählen. Im anderen Fall liegt ein Konfigurationsdefekt vor. In diesem Fall kann die "lokale Einzelpunktüberprüfung" gestartet werden, mit deren Hilfe der bzw. die unbestimmten Neupunkte lokalisiert werden können.

Sollte die Inversion möglich gewesen sein bzw. mit Hinzunahme der entsprechenden Anzahl von Bedingungsgleichungen als freies Netz aufgelöst worden sein, ergeben sich nach der Berechnung folgende Möglichkeiten:

ERGEBNIS:

Bildschirmanzeiige
Gesamtausdruck
Ausgabe der Kofaktorenmatrix
Textausgabe der Kovarianzmatrix
relative Fehlerellipse
Hauptkomponente berechnen
Kondition nach Spektralnorn
Iterative Ausgleichung
Bildschirm - Grafikausgabe

Abb. 7.5

Der "Gesamtausdruck" beinhaltet die Steuerdaten, Festpunktkoordinaten, Näherungswerte und ausgeglichene Werte der Neupunktkoordinaten sowie Genauigkeits- und Zuverlässigkeitskennzahlen.

Die "Textausgabe" der Kovarianzmatrix Σ_{xx} kann für Folgeprogramme, etwa dem Matrizenpaket GAUSS, aufgerufen werden.

Zur "Hauptkomponentenuntersuchung" kann die Hauptkomponente gem. Abschnitt 5.3 berechnet werden.

Sollen alle Eigenwerte zur Berechnung der "Kondition" des Normalgleichungssystems mit Hilfe der Spektralnorn berechnet werden, ist der entsprechende Menüpunkt aufzurufen. Die iterative Berechnung mittels Jacobi-Verfahren kann zum beliebigen Zeitpunkt abgebrochen werden.

Die "iterative Ausgleichung" ermöglicht die Verwendung der ausgeglichenen Koordinaten als neue Näherungswerte und startet einen erneuten Ausgleichsdurchgang. Diese Vorgehensweise ist sinnvoll, wenn vermutet wird, daß die zur Linearisierung erforderlichen Näherungskordinaten nicht ausreichend genau waren. Ein zweiter Menüpunkt innerhalb der Iterationsausgleichung stellt die Prüfung der Festpunkte dar. In diesem Fall werden alle Festpunkte zu Neupunkten umgewandelt und neue Festpunkte mit eben diesen Koordinaten geschaffen. Zwischen diesen Festpunkten und den geschaffenen Neupunkten werden beobachtete Koordinatendifferenzen in X und Y eingesetzt. Die Genauigkeiten werden dabei aus der in Abschnitt

"mittlere Meßfehler" angegebenen Koordinatendifferenzfehler übernommen. Abschließend wird eine neuerliche Ausgleichung gestartet. Dieser Programmpunkt ist nur bei einer Ausgleichung mit vorhandenen Beobachtungen möglich. Der Abschnitt lokale Berechnung der Neupunkte startet die in Abschnitt 5.4 angegebene Strategie. Es wird dabei für jeden Punkt getrennt eine Einzelpunktuntersuchung inklusive Fehlerrechnung durchgeführt. Liegen reale Beobachtungswerte vor, wird für jede betroffene Beobachtung die Konsequenz des Fortlassens auf die unbekannt Parameter und die Varianz der Gewichtseinheit ausgewiesen.

Der Menüpunkt Bildschirm Druckergraphik bietet bei Verwendung einer Hercules-Graphikkarte die Möglichkeit einer direkten Darstellung am Bildschirm bzw. am Graphikdrucker (diese Option ist lediglich für die Programmversion bis maximal 95 Unbekannte verfügbar).

7.2 Stufenweiser Analyseablauf

In diesem Abschnitt soll eine Vorgangsweise, die sich nach umfangreichen Testberechnungen an etwa 15 geodätischen Lagenetzen herausgebildet hat, stufenweise beschrieben werden. Als Ausgangsstufe wird dazu die erfaßte oder übernommene Bearbeitungsdatei des gesamten Netzes vorausgesetzt.

Stufe 1: Näherungskoordinatenüberprüfung

In einem ersten Programmdurchlauf wird die geometrische Bestimmbarkeit des Netzes überprüft. Sollten Konfigurationsdefekte auftreten, müssen diese durch zusätzliche Beobachtungen behoben werden oder das Netz um entsprechende Netzteile verringert werden. An die Qualität der Näherungskoordinaten werden üblicherweise nicht allzu hohe Forderungen gestellt. Um Rückwirkungen zu vermeiden, sollte eine Iterationsausgleichung (ausgeglichenen Koordinatenwerte als neue Näherungskoordinaten) durchgeführt werden. Als Abbruchkriterium kann dabei die Veränderung der Varianz der Gewichtseinheit Anwendung finden.

Stufe 2: Modellüberprüfung

In dieser Stufe wird die Modellbildung bezüglich grober Meßfehler bzw. unrichtiger stochastischer Modellvorgabe überprüft. Alle Beobachtungen von und zu Festpunkten werden dazu eliminiert. In Ausnahmefällen kann von dieser Vorgangsweise abgesehen werden, wenn Festpunkte versteifend auf das Gesamtnetz wirksam werden. Diese Vorgangsweise muß von Fall zu Fall entschieden werden. Es liegt nunmehr eine Ausgleichung mit freier Datumsvorgabe vor. Da die Maßstabsbestimmung auf Grund der beobachteten Strecken in den meisten Fällen genauer ist als die Maßstabsinformation, welche aus den angenommenen Festpunkten abgeleitet werden kann, sollte mit 3 Bedingungen gerechnet werden. Mit Hilfe des Globaltestes (4-3) und der Überprüfung der normierten Verbesserungen (data snooping) wird nach groben Beobachtungsfehlern gesucht. Wurden mehrere grobe Fehler erkannt, wird der mit der größten Kenngröße versehene eliminiert und eine erneute Ausgleichung durchgeführt. Gleichzeitig wird eine Varianzkomponentenuntersuchung in jedem Schritt durchgeführt. Gruppenweise, nach Beobachtungsformen gegliedert, wird die entsprechende Varianz des Gewichtseinheitsanteil ausgegeben. Eine Annäherung kann durch globale und im Einzelfall auch durch individuelle Anpassung der Meßgenauigkeiten unter Berücksichtigung der Gerätegenauigkeiten erreicht werden. Die individuelle Genauigkeitsanpassung jener Beobachtungen sollte immer erst nach der Sichtung des Beobachtungsprotokolles bzw. der Meßumstände durchgeführt werden.

In vielen Fällen kann eine geeignete Netzpartitionierung die Transparenz der Untersuchung in Stufe 2 deutlich erhöhen. Folgende Kriterien können für sich oder in Kombination für die Netzpartitionierung herangezogen werden:

- Geometrische Zusammenhänge (Netzteile, die durch Beobachtungen besonders gut verknüpft sind).
- Genauigkeitsmäßige Zusammengehörigkeit (Netzteile, die mit bestimmten Geräten oder Gerätekombinationen beobachtet wurden).

- Epochenweise Zusammengehörigkeit (Netzteile, die jeweils zu einer Beobachtungsepoche oder von einem Meßtrupp beobachtet wurden).

Ein weiterer Vorteil der Netzpartitionierung ist die Anwendung des data snooping auf kleinere Netzformen; dies wirkt sich insbesondere auf den Globaltest positiv aus.

Als Ergebnis erhält man aus diesem Schritt die geprüften Beobachtungselemente (vermindert um die Beobachtungen von und zu Festpunkten !) und die überprüften Varianzkomponenten der einzelnen Beobachtungsgruppen. Mit Hilfe der Zuverlässigkeitskennzahlen können Aussagen über nichtkontrollierbare Beobachtungen bzw. geometrische Netzschwachstellen gefunden werden. Wurden Festpunkte in Neupunkte umgewandelt und in die Analyse nach Stufe 2 einbezogen, kann diese Information in der Stufe 3 der Festpunktüberprüfung bevorzugt behandelt werden. Die Genauigkeitsaussagen der in der Stufe 2 durchgeführten freien Netzausgleichungen stellen das Minimum ohne Festpunktzwang dar.

Stufe 3: Überprüfung des Festpunktanschlusses

In dieser Analysestufe wird der Festpunktanschluß überprüft. Zielvorstellung ist dabei eine möglichst widerspruchsfreie Datumsvorgabe durch eine Anzahl von Festpunkten. Zu diesem Zweck wird eine Gruppe von Ersatzbeobachtungen dem Beobachtungsmaterial aus Stufe 1 zugeschlagen. Die Gruppe der Ersatzbeobachtungen umfaßt für jeden Festpunkt zwei Koordinatenzuschläge. Die Genauigkeiten dieser Koordinatenzuschläge muß global vorab angegeben werden. Dabei können Genauigkeitsinformationen aus eventuellen Vorausausgleichungen oder Vergleichsoperaten Anwendung finden. Für die Ersatzbeobachtungen und die in Stufe 2 nicht geprüften Festpunktbeobachtungen werden nun dieselben Tests wie in Stufe 2 durchgeführt. Grobe Beobachtungsfehler werden aufgedeckt und die Varianzkomponentenuntersuchung durchgeführt, wobei die realen Beobachtungen ja bereits geprüft sind. Verfälschungen im Bereich der Varianzkomponenten bzw. nunmehr auftretende grobe Beobachtungsfehler sind also dem Festpunktzwang zuzuordnen. Durch geeignete Manipulationen sind diese Effekte zu bereinigen. Zeigen sich die Ersatzbeobachtungen eines zu

überprüfenden "Festpunktes" als ausreichend kontrollierbar, kann dieser wieder zu einem Festpunkt umgewandelt werden. Dabei werden die entsprechenden Ersatzbeobachtungen gestrichen und der Punkt wieder zum Festpunkt umgewandelt. Ergeben sich für die Ersatzbeobachtungen eines "Festpunktes" grobe Fehler, sollte dieser Festpunkt vorübergehend zum Neupunkt umgewandelt werden. Sinkt in einem weiteren Berechnungsdurchlauf die Varianz der Gewichtseinheit ab, ist dieser Festpunkt als Neupunkt zu führen, sofern er für das Netz von ausgeprägter geometrischer Bedeutung ist. Alternativ dazu kann dieser Punkt auch aus der Ausgleichung eliminiert werden.

"Festpunkte" mit wenig zuverlässigen Ersatzbeobachtungen sollten in einem Durchlauf versuchsweise gestrichen werden. Ist keine wesentliche Rückwirkung auf die Varianz der Gewichtseinheit festzustellen, könnte dieser Festpunkt letztendlich wieder aufgenommen werden. Die Frage, ob solche Festpunkte nicht überhaupt gestrichen werden sollten, hängt von der Zielvorgabe des jeweiligen Netztypes ab. Insbesondere der Übergriff über mehrere Hierachiestufen kann, wie dies im Beispiel EP - Pottendorf im Abschnitt 8.2 gezeigt wird, Probleme mit sich bringen.

Als Ergebnis der Stufe 3 liegt ein geprüftes Modell für Beobachtungen und Ersatzbeobachtungen vor. Eine Gruppe von möglichst widerspruchsfreien Anschlußkoordinaten wurde ausgewählt. In der damit gewonnenen "weichen" Lagerung des Netzes können auch für die Anschlußpunktskoordinaten Zuverlässigkeiten abgeschätzt werden. Die Genauigkeitsmaße stellen repräsentative Werte unter Berücksichtigung der Ersatzbeobachtungen dar.

Stufe 4: Endgültige hierachische Ausgleichung

Verdichtungsnetze der Landesvermessung werden weiterhin hierachisch eingeschaltet. In der letzten Stufe wird daher die in Stufe 3 gewonnene Festpunktauswahl zur Modellbildung übernommen. Die Beobachtungen aus Stufe 1 bzw. kontrolliert und überprüft nach Stufe 2 werden mit dem in Stufe 2 und 3 verifizierten stochastischen Modell einer endgültigen Ausgleichung mit fehlerfreien Festpunkten als Datums-
punkte unterworfen.

Auf Grund der fehlerfreien Festpunkte sind jedoch zu optimistische Genauigkeitsaussagen zu erwarten. Diese Genauigkeitssituation im Verdichtungsnetz ist nunmehr die Basis zum Vergleich bzw. der Überprüfung der Genauigkeitsmaße für Verdichtungsnetze im Bereich der Landesvermessung.

Bei den Testberechnungen und deren Interpretation hat sich gezeigt, daß dieser starre Aufbau in mehreren Stufen nicht in jedem Fall durchzuhalten ist. Insbesondere die Rückwirkung zwischen Stufe 2 und 3 erforderte zeitweise Modifikationen dieses Ablaufes. Darüber hinaus wurden auch bereits in Stufe 2 bzw. Stufe 3 Qualitätskriterien geodätischer Netze zur Interpretation zu diesem frühen Zeitpunkt herangezogen. Für Netzsonderformen wurde auch eine Hauptkomponentenuntersuchung durchgeführt. Das Hilfsmittel lokale Einzel-punktuntersuchung (Abschnitt 5.4) erlaubte eine punktweise Untersuchung von "groben" Fehlern, bezogen auf die jeweils betroffenen Netzpunkte. Die Interpretation von Verdichtungsnetzen kann nicht in ein automatisiertes Analyseverfahren eingebaut werden. Vielmehr eine interaktive, unter Berücksichtigung sämtlicher vorhandener Informationen durchgeführte schrittweise Analyse, stellt eine Lösungsmöglichkeit dar.

8. Testberechnungen

In diesem Abschnitt sollen 4 Testbeispiele gezeigt werden.

Das Netz T2 stellt eine synthetische Aufgabenstellung dar und wurde zu Demonstrationszwecken gewählt.

Das Operat Pottendorf wurde anlässlich der Meßübungen WS 1986 am Institut für Landesvermessung und Ingenieurgeodäsie der T.U. Wien beobachtet und gemäß Dienstvorschrift Nr. 9 ausgewertet. Die übungs-mäßige Beobachtungsweise mit einer großen Anzahl von redundanten Messungen ermöglichte nunmehr umfangreiche Testbearbeitungen und Interpretationen mittels Analyseausgleichung.

Das Netzoperat B - 79 Sieggraben wurde vom BA f.E u.Vw zur Verfügung gestellt und zeigt besonders die Notwendigkeit der Einbeziehung von Zuverlässigkeitskriterien bei der Qualitätsbestimmung von Verdichtungsnetzen.

Das Operat BRAND - NAGELBERG wurde im Zuge der Diplomarbeit Döller (1988) beobachtet. Es soll als größeres Netzbeispiel zur Untersuchung verwendet werden.

Der vollständige Computerausdruck wird nur für das synthetische Bsp. T2 wiedergegeben. Für die weiteren Beispiele werden jeweils nur die für die Erklärungen und Bemerkungen relevanten Auszüge eingearbeitet. Der thematische Bezug wird durch das Symbol < I mit nachfolgender Kennzahl hergestellt. Diese wird je Berechnungsablauf eindeutig angegeben.

8.1 Schematisches Netz T2

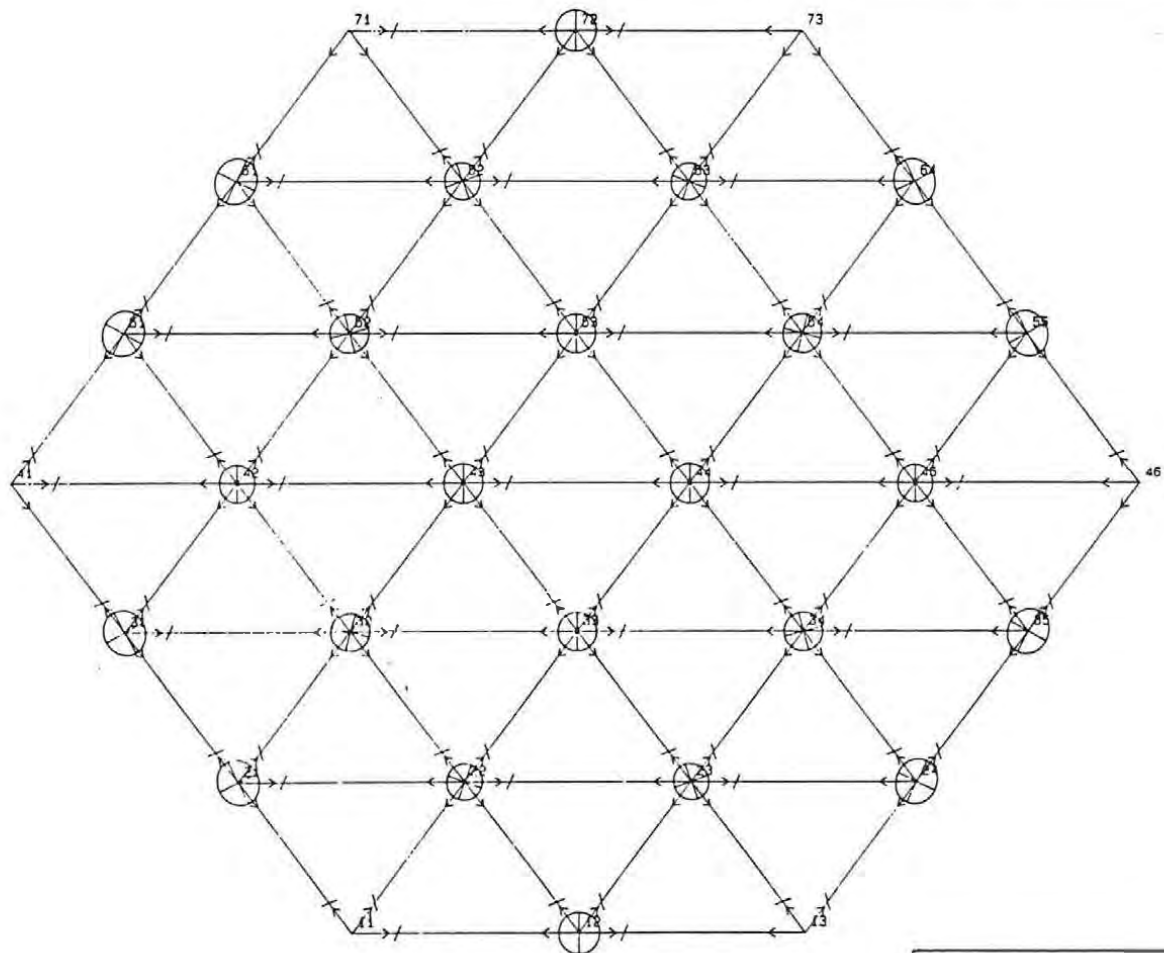
An Hand dieses Netzes sollen neben einer allgemeinen Erklärung die Auswirkungen verschiedener Datumsvorgaben aufgezeigt werden.

Die Variante T2 (Abb. 8.1) zeigt eine hierarchische Datumsvorgabe mit 6 Festpunkten. Der angeschlossene Ergebnisausdruck umfaßt folgende Abschnitte:

1: globale Informationen zur Ausgleichung

T2
Hierarch. Datumswahl

Punktverbindungen:
Richtung >
Distanz /
Zenitdistanz Z



FEHLER/VEKTOREN : 	Dissertation Heinz STANEK

Abb. 8.1

PROJEKT : T2 88/09/01 15:55:45

Kommentar : Testbeispiel: Hierarchischer Netzaufbau - synthetisches Regelnetz

Punkte : 30 (6 Festpunkte, 24 Neupunkte)
 Beobachtungen : 213 (142 Richtungen, 71 Strecken)
 Unbekannte : 78 (48 Koordinaten, 30 Orientierungen)
 Redundanz : 135
 Kondition : 6
 Bedingungen : 0

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 1.0
 Festpunkt-Lagefehler ± 0.0 mm
 Richtungsfehler ± 5.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 4.0 mm
 Entfernungsabhängiger Streckenfehler ± 0.0 ppm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 1.1
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 1.2 cc
 Streckenfehler ± 0.9 mm

FESTPUNKTE

Nummer	Bezeichnung	Y	X
11	11	1100.000	200.005
13	13	2299.998	200.014
21	21	200.001	1399.995
46	46	3200.002	1400.005
71	71	1099.996	1000.009
73	73	2300.002	2600.023

NEUPUNKTE (Näherungskordinaten)

Nummer	Bezeichnung	Y	X
12	12	1699.994	200.013
21	21	799.996	600.005
22	22	1399.995	600.011
23	23	1999.996	600.016
24	24	2599.999	600.015
31	31	499.995	1000.001
32	32	1099.995	1000.009
33	33	1699.997	1000.014
34	34	2299.998	1000.019
35	35	2900.000	1000.012
42	42	799.997	1400.004
43	43	1399.998	1400.012
44	44	2000.000	1400.017
45	45	2600.002	1400.018
51	51	499.998	1799.998
52	52	1099.999	1800.008
53	53	1700.001	1800.015
54	54	2300.005	1800.019
55	55	2900.005	1800.016
61	61	799.999	2200.001
62	62	1399.999	2200.010
63	63	2000.004	2200.020
64	64	2600.010	2200.019
72	72	1700.005	2600.016

NEUPUNKTE (Ausgleichsergebnisse)

Nummer	Bezeichnung	Y	X
12	12	1699.994	200.013
21	21 Or = 399.9999	799.995	600.006
22	22 Or = 399.9993	1399.995	600.011
23	23 Or = 399.9993	1999.996	600.016
24	24 Or = 399.9995	2599.999	600.015
31	31 Or = 0.0004	499.995	1000.001
32	32 Or = 399.9995	1099.995	1000.009
33	33 Or = 399.9993	1699.997	1000.014
34	34 Or = 399.9994	2299.998	1000.019
35	35 Or = 399.9998	2900.000	1000.012
42	42 Or = 0.0003	799.997	1400.004
43	43 Or = 399.9992	1399.998	1400.012
44	44 Or = 399.9994	2000.000	1400.017
45	45 Or = 399.9996	2600.002	1400.018
51	51 Or = 0.0007	499.998	1799.998
52	52 Or = 0.0007	1099.999	1800.008
53	53 Or = 0.0007	1700.001	1800.015
54	54 Or = 0.0007	2300.005	1800.019
55	55 Or = 0.0007	2900.005	1800.016
61	61 Or = 0.0007	799.999	2200.001
62	62 Or = 0.0007	1399.999	2200.010
63	63 Or = 0.0007	2000.004	2200.020
64	64 Or = 0.0007	2600.010	2200.019
72	72 Or = 0.0007	1700.005	2600.016

81

◀ 1

◀ 2

◀ 3

◀ 4

GENAUIGKEIT DER UNBEKANNTEN

Fehler der Gewichtseinheit a posteriori: +/- 1.06
 Fehler der Gewichtseinheit a priori : +/- 1.00
 Die Prüfergröße T ergibt sich zu : 1.12
 Sicherheitsgrenzen der FISHER - Verteilung : 1.23 (S = 95%)
 Globaltest:
 Sicherheitsgrenzen der FISHER - Verteilung : 1.02 (S* = 56%)

◀6

Neupunkte : mX, mY, a, b in mm; mOr in cc; theta in gon

Nummer	mX	mY	mOr	a	b	theta
12	2.2	2.2	2.9	2.2	2.2	100.0
21	2.4	2.3	3.1	2.5	2.2	171.0
22	1.9	1.9	2.4	2.0	1.9	20.6
23	1.9	1.9	2.4	2.0	1.9	177.4
24	2.4	2.3	3.1	2.5	2.2	29.0
31	2.4	2.3	3.0	2.5	2.2	168.2
32	2.1	2.1	2.4	2.1	2.1	16.7
33	2.1	2.1	2.4	2.1	2.1	200.0
34	2.1	2.1	2.4	2.1	2.1	183.3
35	2.4	2.3	3.0	2.5	2.2	51.8
42	2.1	1.9	2.4	2.1	1.9	200.0
43	2.2	2.1	2.4	2.2	2.1	200.0
44	2.2	2.1	2.4	2.2	2.1	200.0
45	2.1	1.9	2.4	2.1	1.9	0.0
51	2.4	2.3	3.0	2.5	2.2	51.8
52	2.1	2.1	2.4	2.1	2.1	183.3
53	2.1	2.1	2.4	2.1	2.1	200.0
54	2.1	2.1	2.4	2.1	2.1	16.7
55	2.4	2.3	3.0	2.5	2.2	168.2
61	2.4	2.3	3.1	2.5	2.2	29.0
62	1.9	1.9	2.4	2.0	1.9	177.4
63	1.9	1.9	2.4	2.0	1.9	20.6
64	2.4	2.3	3.1	2.5	2.2	171.0
72	2.2	2.2	2.9	2.2	2.2	100.0

Festpunkte : mittlere Fehler der Orientierungen in cc

Nummer	mOr
11	3.5
13	3.5
41	3.5
46	3.5
71	3.5
73	3.5

52	52	Or =	399.9992	1977.999	1800.008
53	53	Or =	399.9993	1700.001	1800.015
54	54	Or =	399.9998	2300.005	1800.019
55	55	Or =	0.0007	2900.005	1800.016
61	61	Or =	399.9997	799.999	2200.001
62	62	Or =	399.9991	1099.999	2200.010
63	63	Or =	399.9994	2000.004	2200.020
64	64	Or =	0.0003	2600.010	2200.019
72	72	Or =	399.9991	1700.005	2600.016

RELATIVE FEHLERELLIPSEN

Angaben in mm und gon.

←7

RFE 42-34 :

a = 2.8	mdX = 2.8	Längsfehler = 2.6
b = 2.6	mdY = 2.6	Querfehler = 2.6
theta = 1.6		

RFE 42-54 :

a = 2.8	mdX = 2.8	Längsfehler = 2.6
b = 2.6	mdY = 2.6	Querfehler = 2.6
theta = 198.4		

RFE 34-54 :

a = 2.6	mdX = 2.4	Längsfehler = 2.4
b = 2.4	mdY = 2.6	Querfehler = 2.6
theta = 100.0		

2: globale Formulierung des stochastischen Modellbildung. Dieses könnte für einzelne Beobachtungen abgändert oder durch die Gewichtsvorgabe nach B.A.f.E.u.V.W. ersetzt worden sein. Die tatsächlich angewendeten Beobachtungsfehler werden, je Beobachtung im Abschnitt der Zuverlässigkeitsberechnung ausgewiesen.

3: Ausgabe von s_0^2 nach der Ausgleichung und die Varianzkomponenten gegliedert nach Beobachtungsgruppen. Dieser Abschnitt entfällt bei Simulationsausgleichungen.

4: Festpunktsinformationen. In der zweiten Zeile wird, falls der Festpunkt als Standpunkt eines Richtungssatzes diene, die ausgeglichene Orientierungsunbekannte des Richtungssatzes ausgewiesen.

5: Neupunkte: Näherungskordinaten und Ausgleichungsergebnisse und wieder in der zweiten Zeile eine ev. vorhandene und ausgeglichene Orientierungsunbekannte.

6: Genauigkeitsinformationen beinhaltet zum ersten die Formulierung des Globaltests (entfällt bei Simulationsausgleichungen). Dabei wird der Wert der F - Verteilung einerseits für $\alpha = 5\%$ und andererseits für α_c gemäß der aktuellen Anzahl von Überbestimmungen ($n - u$) ausgewiesen. Danach folgen die Genauigkeitsinformationen der Unbekannten inkl. Orientierung und ev. Maßstabsunbekannten.

7: Die Werte der relativen Fehlerellipsen und die Fehlermaße der Koordinatendifferenzen, der Verbindungsstrecke und der Querabweichung werden ausgewiesen.

Der Übergang von Helmert'schen Punktfehlerellipsen bzw. relativen Fehlerellipsen auf statistisch aussagekräftige Konfidenzellipsen wird durch Multiplikation der Werte der Halbachsen mit den Werten aus Abschnitt 5 erreicht.

Die Variante T2F (Abb. 8.2) zeigt eine Datumsvorgabe ohne Festpunkte, also eine freie Netzausgleichung.

1: Da Streckenbeobachtungen vorliegen, kann der Netzmaßstab bei der freien Netzausgleichung aus diesen abgeleitet werden und es ist ein Rangdefizit $d = 3$ zu erwarten. Sollten keine Streckenbeobachtungen vorliegen, oder eine Maßstabsunbekannte mitgeführt werden, beträgt das Rangdefizit $d = 4$.

2: Die Varianzkomponenten zeigen gute Übereinstimmung mit den Vergleichswerten der Variante T2. Da die "Beobachtungen" für dieses Netz mittels Zufallszahlengenerator gemäß einer Normalverteilung mit den Eingangsgrößen der angenommenen Meßgenauigkeit generiert wurden, ist diese Ergebnis zu erwarten.

3: Die Zuverlässigkeiten der Beobachtungen werden nach Angabe der Testcharakteristik nach Beobachtungsarten gegliedert, gruppenweise angegeben. Ausgewiesen werden je Beobachtung l_1 die Genauigkeit vor und nach der Ausgleichung, der Redundanzanteil r_1 in ‰, der Grenzwert GWA für die Aufdeckung eines groben Fehlers in der Beobachtung l_1 (innere Zuverlässigkeit) und der Einflußfaktor E_f zur Charakterisierung der äußeren Zuverlässigkeit.

Die zweite Zeile beinhaltet eine Gegenüberstellung des Beobachtungswertes und des ausgeglichenen Meßwertes, die Verbesserung v_1 und die normierte Verbesserung w_1 . Die Angabe der Werte der zweiten Zeile entfallen bei Simulationsausgleichungen.

Als Interpretationshilfe werden die angeführten Werte gemäß der in Abschnitt 5.2 angegebenen Schrankenwerte mittels angemerakter Symbole klassifiziert. Das entsprechende Symbol steht dabei immer auf der rechten Seite des betroffenen Wertes:

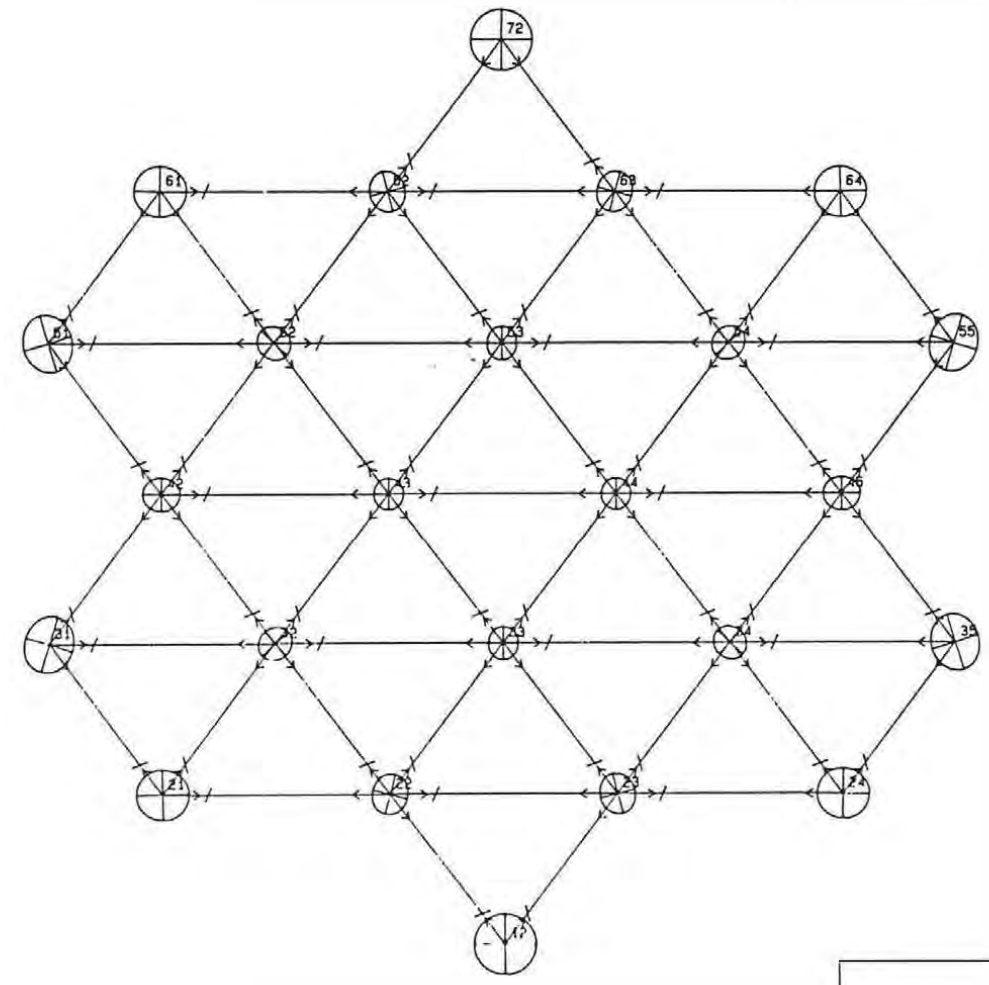
Wert	Schranke	Symbol	Konsequenz
r_1	< 0.3	<<	schlecht kontrollierbare Beo. zusätzliche Beo. vorsehen
r_1	> 0.6	>>	sehr gut kontrollierbare Beo. Beobachtung ev. weglassen
GWA	> 6 m_1	//	
GWA	> 8 m_1	@@	
Ef	> 8	!!	
Ef	> 10	##	

Führt der Vergleich der normierten Verbesserung w_1 mit dem kritischen Wert k zur Ablehnung des lokalen Tests ($w_1 > k$) wird dies ebenfalls durch ein Symbol kenntlich gemacht:

w_1	>	3.3	<>
-------	---	-----	----

T2F
Freie Ausgleichung

Punktverbindungen:
Richtung >
Distanz /
Zenitdistanz ... Z



0 250 500 750 1000 1250 m	
FEHLER/VEKTOREN :	Dissertation
0 5 10 15 mm	Heinz STANEK

Abb. 8.2

PROJEKT : T2F 88/09/01 16:05:56

Kommentar : Testbeispiel : ohne Festpunkte

Punkte : 24 (0 Festpunkt, 24 Neupunkte)
 Beobachtungen : 159 (106 Richtungen, 53 Strecken)
 Unbekannte : 72 (48 Koordinaten, 24 Orientierungen)
 Redundanz : 90
 Bedingungen : 3

<1

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 1.0
 Festpunkt-Lagefehler ± 0.0 mm
 Richtungsfehler ± 5.0 cm
 Orientierungsfehler ± 50.0 cm
 Konstanter Streckenfehler ± 4.0 mm
 Entfernungsabhängiger Streckenfehler ± 0.0 ppm

88

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 1.0
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 1.1 cm
 Streckenfehler ± 0.9 mm

<2

FESTPUNKTE

Nummer	Bezeichnung	Y	X

NEUPUNKTE (Näherungs koordinaten)

Nummer	Bezeichnung	Y	X
12	12	1649.294	200.017
21	21	299.956	60.065
22	22	179.999	100.011
23	23	129.966	200.011

ZUVERLÄSSIGKEIT DER BEOBSACHTUNGEN

m0/Jp bzw m0*Jq = mittl. Fehler der Beobachtung vor bzw nach dem Ausgleich.
 r = 1-p*q = Redundanzanteil der Beobachtung an der Gesamtredundanz G.
 GWA = Grenzwert für die Aufdeckung grober Beob.fehler bei a=0.1% und B=90%.
 Ef. = Einflußfaktor (Maß für die äußere Zuverlässigkeit)
 Fehlerwerte und Verbesserungen für Richtungen in cc. für Strecken in mm.
 Der kritische Wert k für die normierten Verbesserungen ist 3.3.

RICHTUNGEN

<3

Anteil v/Pv: 69.38
 m0 der Gruppe Richtungen nach der Ausgleichung +/- 1.10 cc

Station	Ziele	m0/Jp Messwert	m0*Jq ausgeglt	r (%) Verb.	GWA w	Ef.
12	22	5.1	4.2	33.2	36.5 //	5.9
		359.0340	359.0339	-0.81	0.18	
	23	5.1	4.2	33.2	36.5 //	5.9
		40.9670	40.9671	0.81	0.19	
21	22	5.1	3.8	45.3	21.2 //	4.5
		100.0012	100.0004	-7.81	2.68	
	31	5.1	4.0	38.9	33.7 //	5.2
		359.0342	359.0339	-3.43	0.66	
	32	5.1	3.5	53.1	28.8	3.9
		40.9661	40.9672	11.24	3.23	
22	12	5.1	4.0	39.4	33.5 //	5.1
		159.0339	159.0342	2.84	0.73	
	21	5.1	3.6	48.7	30.1	4.2
		299.9992	299.9997	5.45	1.50	
	23	5.1	3.2	60.2	27.1	3.4
		100.0008	100.0002	-6.17	1.92	
	32	5.1	3.3	57.4	27.8	3.6
		359.0339	359.0339	0.16	0.05	
	33	5.1	3.3	59.0	27.4	3.4
		40.9675	40.9673	-2.28	0.70	
23	12	5.1	4.0	39.4	33.5 //	5.1
		240.9670	240.9669	-0.87	0.22	
	22	5.1	3.2	60.2	27.1	3.4
		299.9992	299.9998	5.50	1.71	
	24	5.1	3.6	48.7	30.1	4.2
		100.0008	100.0008	-0.35	0.10	
	33	5.1	3.3	59.0	27.4	3.4
		359.0339	359.0338	-0.85	0.26	
	34	5.1	3.3	57.4	27.8	3.6
		40.9675	40.9672	-3.43	1.03	
24	23	5.1	3.8	45.3	21.2 //	4.5
		299.9996	300.0000	4.04	1.07	
	34	5.1	3.5	53.1	28.8	3.9
		359.0333	359.0333	0.21	0.06	
	35	5.1	4.0	38.9	33.7 //	5.2
		40.9669	40.9665	-4.25	1.07	
21	21	5.1	3.8	45.3	21.7 //	4.7
		159.0328	159.0336	8.05	2.11	
	32	5.1	3.4	56.5	28.0	3.6
		100.0006	99.9998	-7.74	2.51	
	42	5.1	3.8	44.9	21.4 //	4.6

	64	299.9997	299.9957	-0.299	0.09		
		5.1	3.8	31.9	31.7	4.1	
		359.0339	359.0335	-0.45	1.17		
	61	5.1	4.0	38.9	33.7	5.2	
		240.9659	240.9667	7.64	1.92		
		5.1	3.5	53.1	28.8		
		159.0329	159.0332	3.30	0.95		
		5.1	3.8	45.3	31.2	4.5	
	62	100.0006	99.9995	-10.94	2.91		
		5.1	3.3	57.4	27.8	3.6	
		240.9670	240.9673	2.64	0.80		
		5.1	3.3	59.0	27.4	3.4	
		159.0339	159.0338	-0.65	0.20		
		5.1	3.6	48.7	30.1	4.2	
		299.9992	299.9994	4.44	1.22		
		5.1	3.2	50.2	27.1	3.4	
		100.0008	100.0000	-7.86	2.45		
		5.1	4.0	39.4	33.5	5.1	
		40.9675	40.9676	1.42	0.36		
	63	5.1	3.3	55.0	27.4	3.4	
		240.9670	240.9671	1.32	0.41		
		5.1	3.3	57.4	27.8	3.6	
		159.0339	159.0339	-0.09	0.03		
		5.1	3.2	60.2	27.1	3.4	
		299.9992	299.9996	4.12	1.28		
		5.1	3.6	48.7	30.1	4.2	
		100.0008	100.0005	-3.04	0.84		
		5.1	4.0	39.4	33.5		
		359.0339	359.0337	-0.28	0.58		
	64	5.1	3.5	53.1	28.8		
		240.9660	240.9668	8.45	2.43		
		5.1	4.0	38.9	33.7	5.2	
		159.0340	159.0338	-2.07	0.52		
		5.1	3.8	45.3	31.2		
		300.0004	299.9998	-6.38	1.70		
	72	62	5.1	4.2	30.2	36.5	
		240.9672	240.9674	1.84	0.44		
	63	5.1	4.2	30.2	36.5		
		159.0340	159.0338	-1.84	0.44		

		Mittel	5.1	3.4	54.0	28.9	3.9

STRECKEN

Anteil vFv: 23.88
 m0 der Gruppe Strecken nach der Ausgleichung +/- 0.85 mm

Station	Ziele	m0/Jp Messwert	m0*Jq ausgeg1	n(%) Verb.	GWA w	Ef.	
	12	22	4.1	3.1	43.7	25.4	4.1
			499.995	499.995	0.43	0.14	
		23	4.1	3.1	43.7	25.4	4.1
			500.004	500.002	-1.71	0.56	
	21	22	4.1	2.9	48.7	24.1	4.1
			600.003	599.997	-6.48	2.22	
		31	4.1	3.0	46.2	26.7	4.5
			499.994	499.997	1.99	0.67	
		32	4.1	3.1	43.7	25.4	4.1

	22	23	500.004	500.002	-1.56	0.64	
			4.1	2.4	66.2	30.7	3.6
			600.003	599.999	-4.02	1.70	
		32	4.1	2.3	66.8	20.6	2.9
			499.995	499.997	1.69	0.73	
		33	4.1	2.2	69.8	20.1	2.7
			500.004	500.003	-1.02	0.46	
	23	24	4.1	2.9	48.7	24.1	4.2
			600.004	600.003	-0.70	0.24	
		33	4.1	2.2	69.8	20.1	2.7
			499.995	499.996	1.18	0.53	
		34	4.1	2.3	66.8	20.6	2.9
			500.004	500.002	-1.95	0.83	
	24	34	4.1	2.4	64.3	21.0	3.1
			500.004	500.004	-0.24	0.10	
		35	4.1	3.0	46.2	24.7	4.5
			500.001	499.999	-2.11	0.71	
	31	32	4.1	2.4	64.7	20.9	3.1
			600.003	600.001	-1.70	0.70	
		42	4.1	3.0	46.3	24.7	4.4
			500.004	500.007	3.09	1.04	
	32	33	4.1	2.2	71.5	19.9	2.6
			600.003	600.001	-1.63	0.75	
		42	4.1	2.3	68.3	20.3	2.8
			499.995	499.995	-0.45	0.20	
		43	4.1	2.2	70.8	20.0	2.7
			500.004	500.004	0.08	0.04	
	33	34	4.1	2.2	71.5	19.9	2.6
			600.003	600.000	-2.57	1.18	
		43	4.1	2.2	71.4	19.9	2.6
			499.995	499.996	1.24	0.57	
		44	4.1	2.2	71.4	19.9	2.6
			500.004	500.003	-1.08	0.49	
	34	35	4.1	2.4	64.7	20.9	3.1
			600.003	600.002	-1.11	0.46	
		44	4.1	2.2	70.8	20.0	2.7
			499.995	499.997	1.56	0.71	
		45	4.1	2.3	68.3	20.3	2.8
			500.004	500.001	-2.82	1.23	
	35	45	4.1	3.0	46.3	24.7	4.4
			500.002	500.002	0.44	0.15	
	42	43	4.1	2.2	71.5	19.9	2.6
			600.003	600.002	-1.46	0.67	
		51	4.1	3.0	46.3	24.7	4.4
			499.995	499.993	-1.55	0.52	
		52	4.1	2.3	68.3	20.3	2.8
			500.004	500.005	1.13	0.49	
	43	44	4.1	2.1	72.6	19.7	2.5
			600.003	600.002	-1.08	0.50	
		52	4.1	2.2	70.8	20.0	2.7
			499.995	499.996	0.96	0.44	
		53	4.1	2.2	71.4	19.9	2.6
			500.004	500.004	-0.13	0.06	
	44	45	4.1	2.2	71.5	19.9	2.6
			600.003	600.002	-1.38	0.63	
		53	4.1	2.2	71.4	19.9	2.6
			499.995	499.997	2.29	1.05	
		54	4.1	2.2	70.8	20.0	2.7
			500.004	500.003	-0.82	0.37	
	45	54	4.1	2.3	68.3	20.3	2.8
			499.995	499.998	3.09	1.25	
		55	4.1	3.0	46.3	24.7	4.4
			500.004	499.999	-4.81	1.61	
	51	52	4.1	2.4	64.7	20.9	3.1
			600.004	600.000	-3.64	1.51	
		61	4.1	3.0	46.3	24.7	4.4

		499.993	500.002	8.70	2.91	
52	53	4.1	2.2	71.3 >>	19.9	2.6
		600.003	600.002	-0.69	0.32	
	61	4.1	2.4	64.3 >>	21.0	3.1
		499.995	499.995	0.44	0.18	
	62	4.1	2.3	66.8 >>	20.6	2.9
		500.004	500.003	-0.65	0.28	
53	54	4.1	2.2	71.3 >>	19.9	2.6
		600.003	600.004	0.60	0.27	
	62	4.1	2.2	69.8 >>	20.1	2.7
		499.995	499.997	2.11	0.94	
	63	4.1	2.2	69.8 >>	20.1	2.7
		500.004	500.005	1.01	0.45	
54	55	4.1	2.4	64.7 >>	20.9	3.1
		600.003	600.000	-2.65	1.10	
	63	4.1	2.3	66.8 >>	20.6	2.9
		499.995	500.000	4.57	1.95	
	64	4.1	2.4	64.3 >>	21.0	3.1
		500.004	500.002	-1.61	0.66	
55	64	4.1	3.0	46.2	24.7 //	4.5
		499.997	499.999	1.74	0.58	
61	62	4.1	2.9	48.7	24.1	4.2
		599.996	600.004	8.24	2.83	
62	63	4.1	2.4	66.2 >>	20.7	3.0
		600.003	600.005	2.01	0.85	
	72	4.1	3.1	43.7	25.4 //	4.7
		500.004	500.007	2.72	0.89	
63	64	4.1	2.9	48.7	24.1	4.2
		600.003	600.005	1.82	0.62	
	72	4.1	3.1	43.7	25.4 //	4.7
		499.995	499.997	1.78	0.58	

	:Mittel	4.1	2.5	61.8 >>	21.7	3.3

Als nächste Variante werden in T2FF einige Beobachtungen mit kleinen "groben" Fehlern beaufschlagt um die Anwendung des data snooping zu zeigen:

von 53	Originalwert	"grober" F.	Fehlerh. W.
nach 54	600.003	+ 0.015	600.018
nach 63	500.004	+ 0.020	500.024

Tab. 8.1

1: Die Beobachtung mit dem größten Wert für w_1 ist der verfälschten Streckenbeobachtung von 53 nach 63 zugeordnet. Diese sollte erst nach Überprüfung der Beobachtungsprotkollie eliminiert werden.

2: Die lokale Einzelpunktuntersuchung zeigt insbesondere die lokale Bestimmbarkeit des Einzelpunktes auf Grund des ausgewiesenen Konditionswertes (gemäß Spektraln σ m).

3: Je Neupunkt wird zeilenweise, für jede Beobachtung der Meßwert, der ausgeglichene Wert, die Verbesserung aus der Gesamtausgleichung v_1 sowie der Zuschlag an die Unbekannten $x, y, (ev. o)$ aufgrund der Gesamtausgleichung, bei Fortlassen der entsprechenden Beobachtung l_1 in der Einzelpunktausgleichung ausgewiesen.

In der letzten Spalte wird der entsprechende Varianzfaktor der Gewichtseinheit angegeben.

Die letzte Zeile umfaßt die Ergebnisse einer Einzelpunktausgleichung mit allen den Punkt betreffenden Beobachtungen.

Zum Abschluß wird für jeden Punkt die lokale Standardabweichung l_{m0} gemäß Abschnitt 5.4 berechnet deren Darstellung in Form eines Histogramms für dieses Beispiel in Abb. 8.3 wiedergegeben ist.

4: Das beobachtungweise Aufrollen der Berechnungen zeigt für die beaufschlagten Beobachtungen eine Verringerung der Varianz der Gewichtseinheit bei Fortlassen dieser Beobachtungen gegenüber dem Vergleichswert mit Berücksichtigung aller Beobachtungen.

ZUVERLÄSSIGKEIT DER BEOBSCHTUNGEN

m0/Jp bzw m0*Jq = mittl. Fehler der Beobachtung vor bzw nach dem Ausgleich.
 r = 1-p*q = Redundanzanteil der Beobachtung an der Gesamtredundanz In.
 GWA = Grenzwert für die Aufdeckung grober Beob.fehler bei α=0.1% und β=80%.
 Ef. = Einflußfaktor (Maß für die äußere Zuverlässigkeit)
 Fehlerwerte und Verbesserungen für Richtungen in cc, für Strecken in mm.
 Der kritische Wert K für die normierten Verbesserungen ist 3.3.

RICHTUNGEN

Anteil vtfPv: 70.90
 mo der Gruppe Richtungen nach der Ausgleichung +/- 1.11 cc

Station	Ziele	m0/Jp Messwert	m0*Jq ausgegl	r(%) Verb.	GWA w	Ef.
12	22	5.6	4.6	33.2	40.2 //	5.9
		359.0340	359.0339	-0.90	0.20	
	23	5.6	4.6	33.2	40.2 //	5.9
21	22	40.9670	40.9671	0.90	0.20	
		5.6	4.1	45.3	34.4 //	4.5
	31	100.0012	100.0004	-7.72	1.86	
22	31	5.6	4.4	38.9	37.2 //	5.2
		359.0342	359.0338	-3.50	0.80	
	32	5.6	3.8	53.1	31.8	3.9
		40.9661	40.9672	11.22	2.92	
	12	5.6	4.4	39.4	36.9 //	5.1
		159.0339	159.0342	2.85	0.65	
23	21	5.6	4.0	48.7	33.2	4.2
		299.9992	299.9997	5.43	1.35	
	23	5.6	3.5	60.2 >>	29.9	3.4
		100.0008	100.0002	-6.09	1.72	
	32	5.6	3.7	57.4	30.6	3.6
		359.0339	359.0339	-0.04	0.01	
24	33	5.6	3.6	59.0	30.2	3.4
		40.9675	40.9673	-2.16	0.60	
	12	5.6	4.4	39.4	36.9 //	5.1
		240.9670	240.9669	-0.90	0.21	
	22	5.6	3.5	60.2 >>	29.9	3.4
		299.9992	299.9997	5.37	1.52	
24	24	5.6	4.0	48.7	33.2	4.2
		100.0008	100.0008	-0.22	0.06	
	33	5.6	3.6	59.0	30.2	3.4
		359.0339	359.0338	-1.09	0.30	
	34	5.6	3.7	57.4	30.6	3.6
		40.9675	40.9672	-3.15	0.86	
31	23	5.6	4.1	45.3	34.4 //	4.5
		299.9996	300.0000	3.81	0.92	
	34	5.6	3.8	53.1	31.8	3.9
		359.0333	359.0333	0.16	0.04	
	35	5.6	4.4	38.9	37.2 //	5.2
		40.9669	40.9665	-3.97	0.90	
32	21	5.6	4.2	43.9	35.0 //	4.7
		159.0328	159.0336	8.04	1.91	
	32	5.6	3.7	56.5	30.8	3.6
		100.0006	99.9998	-7.62	2.06	
42	5.6	4.2	44.9	34.6 //	4.6	

32	21	40.9675	40.9675	-0.42	0.10	
		5.6	3.9	52.2	32.1	4.0
	240.9670	240.9667	-2.65	0.68		
22	22	5.6	3.6	58.1	30.4	3.5
		159.0339	159.0341	1.94	0.53	
	31	5.6	3.7	55.8	31.0	3.7
33	299.9992	299.9996	3.96	1.06		
		5.6	3.4	63.7 >>	29.0	3.1
	100.0008	100.0003	-5.35	1.58		
42	42	5.6	3.7	57.5	30.6	3.6
		359.0339	359.0341	1.69	0.46	
	43	5.6	3.5	60.0 >>	29.9	3.4
33	22	40.9675	40.9675	0.40	0.11	
		5.6	3.6	59.6	30.0	3.4
	240.9670	240.9671	0.57	0.16		
23	23	5.6	3.6	59.6	30.0	3.4
		159.0339	159.0340	1.18	0.33	
	32	5.6	3.4	64.2 >>	28.9	3.1
34	299.9992	299.9998	6.40	1.90		
		5.6	3.4	64.2 >>	28.9	3.1
	100.0008	100.0002	-5.96	1.77		
43	43	5.6	3.5	60.7 >>	29.7	3.3
		359.0339	359.0338	-1.11	0.32	
	44	5.6	3.5	60.7 >>	29.7	3.3
34	23	40.9675	40.9674	-1.08	0.31	
		5.6	3.6	58.1	30.4	3.5
	240.9670	240.9669	-0.78	0.21		
24	24	5.6	3.9	52.2	32.1	4.0
		159.0339	159.0339	-0.49	0.13	
	33	5.6	3.4	63.7 >>	29.0	3.1
35	299.9992	299.9997	5.14	1.52		
		5.6	3.7	55.8	31.0	3.7
	100.0008	100.0008	0.49	0.13		
44	44	5.6	3.5	60.0 >>	29.9	3.4
		359.0339	359.0336	-2.82	0.79	
	45	5.6	3.7	57.5	30.6	3.6
35	24	40.9675	40.9673	-1.55	0.42	
		5.6	4.2	43.9	35.0 //	4.7
	240.9660	240.9665	5.06	1.29		
34	34	5.6	3.7	56.5	30.8	3.6
		300.0006	300.0003	-2.83	0.76	
	45	5.6	4.2	44.9	34.6 //	4.6
42	31	359.0340	359.0338	-2.23	0.54	
		5.6	4.3	42.3	35.7 //	4.8
	240.9670	240.9673	2.63	0.62		
32	32	5.6	3.6	58.5	30.3	3.5
		159.0339	159.0341	2.16	0.60	
	43	5.6	3.4	63.2 >>	29.2	3.2
51	100.0008	100.0001	-7.20	2.11		
		5.6	4.3	42.3	35.7 //	4.8
	359.0339	359.0342	2.90	0.68		
52	40.9675	40.9675	-0.50	0.14		
		5.6	3.6	58.5	30.3	3.5
	240.9670	240.9673	2.54	0.72		
33	32	5.6	3.5	60.5 >>	29.8	3.3
		240.9670	240.9673	2.54	0.72	
	33	5.6	3.5	60.9 >>	29.7	3.3
42	159.0339	159.0339	0.29	0.08		
		5.6	3.4	64.2 >>	28.9	3.1
	299.9992	299.9997	5.48	1.63		
44	44	5.6	3.3	64.8 >>	28.8	3.0
		100.0008	100.0002	-6.13	1.84	
	52	5.6	3.5	60.5 >>	29.8	3.3
53	359.0339	359.0339	-0.00	0.00		
		5.6	3.5	60.9 >>	29.7	3.3
	40.9675	40.9673	-2.18	0.62		
44	33	5.6	3.5	60.9 >>	29.7	3.3

22	23	500.004	500.002	-1.57	0.59	
		4.5	2.6	66.2 >>	22.8	3.0
	32	600.003	599.999	-3.86	1.48	
		4.5	2.6	66.8 >>	22.7	2.9
		499.995	499.997	1.85	0.72	
	33	4.5	2.5	69.8 >>	22.2	2.7
		500.004	500.003	-0.99	0.40	
23	24	4.5	3.2	48.7	26.6	4.2
		600.004	600.003	-0.57	0.18	
	33	4.5	2.5	69.8 >>	22.2	2.7
		499.995	499.996	1.26	0.51	
	34	4.5	2.6	66.8 >>	22.7	2.9
		500.004	500.002	-1.72	0.67	
24	34	4.5	2.7	64.3 >>	23.1	3.1
		500.004	500.004	-0.20	0.07	
	35	4.5	3.3	46.2	27.3 //	4.5
		500.001	499.999	-1.97	0.60	
31	32	4.5	2.7	64.7	23.1	3.1
		600.003	600.001	-1.61	0.60	
	42	4.5	3.3	46.3	27.3 //	4.4
		500.004	500.007	3.11	0.95	
32	33	4.5	2.4	71.3 >>	22.0	2.6
		600.003	600.002	-1.31	0.54	
	42	4.5	2.5	68.3 >>	22.4	2.8
		499.995	499.995	-0.22	0.09	
	43	4.5	2.4	70.8 >>	22.0	2.7
		500.004	500.004	-0.13	0.06	
33	34	4.5	2.4	71.3 >>	22.0	2.6
		600.003	600.001	-2.07	0.86	
	43	4.5	2.4	71.4 >>	21.9	2.6
		499.995	499.997	1.66	0.69	
	44	4.5	2.4	71.4 >>	21.9	2.6
		500.004	500.003	-0.94	0.39	
34	35	4.5	2.7	64.7 >>	23.1	3.1
		600.003	600.002	-0.79	0.30	
	44	4.5	2.4	70.8 >>	22.0	2.7
		499.995	499.997	1.63	0.67	
	45	4.5	2.5	68.3 >>	22.4	2.8
		500.004	500.002	-1.92	0.76	
35	45	4.5	3.3	46.3	27.3 //	4.4
		500.002	500.002	0.34	0.10	
42	43	4.5	2.4	71.5 >>	21.9	2.6
		600.003	600.001	-1.52	0.63	
	51	4.5	3.3	46.3	27.3 //	4.4
		499.995	499.994	-1.29	0.39	
	52	4.5	2.5	68.3 >>	22.4	2.8
		500.004	500.005	1.13	0.45	
43	44	4.5	2.4	72.6 >>	21.8	2.5
		600.003	600.003	-0.05	0.02	
	52	4.5	2.4	70.8 >>	22.0	2.7
		499.995	499.996	1.16	0.48	
	53	4.5	2.4	71.4 >>	21.9	2.6
		500.004	500.002	-1.84	0.77	
44	45	4.5	2.4	71.5 >>	21.9	2.6
		600.003	600.003	0.13	0.05	
	53	4.5	2.4	71.4 >>	21.9	2.6
		499.995	499.998	3.29	1.37	
	54	4.5	2.4	70.8 >>	22.0	2.7
		500.004	500.006	2.25	0.93	
45	54	4.5	2.5	68.3 >>	22.4	2.8
		499.995	499.998	2.68	1.06	
	55	4.5	3.3	46.3	27.3 //	4.4
		500.004	500.000	-3.95	1.20	
51	52	4.5	2.7	64.7 >>	23.1	3.1
		600.004	600.000	-3.73	1.40	
	61	4.5	3.3	46.2	27.3 //	4.5

52	53	499.993	500.002	9.02	2.74	
		4.5	2.4	71.3 >>	22.0	2.6
		600.003	600.001	-2.27	0.94	
	61	4.5	2.7	64.3 >>	23.1	3.1
		499.995	499.996	0.64	0.24	
	62	4.5	2.6	66.8 >>	22.7	2.9
		500.004	500.005	0.66	0.25	
53	54	4.5	2.4	71.3 >>	22.0	2.6
		600.018	600.010	-8.14	3.39	<>
	62	4.5	2.5	69.8 >>	22.2	2.7
		499.995	499.997	2.34	0.95	
	63	4.5	2.5	69.8 >>	22.2	2.7
		500.024	500.013	-11.49	4.66	<>
54	55	4.5	2.7	64.7 >>	23.1	3.1
		600.003	600.000	-3.13	1.17	
	63	4.5	2.6	66.8 >>	22.7	2.9
		499.995	500.002	7.25	2.80	
	64	4.5	2.7	64.3 >>	23.1	3.1
		500.004	500.003	-1.01	0.38	
55	64	4.5	3.3	46.2	27.3 //	4.5
		499.997	499.999	2.31	0.70	
61	62	4.5	3.2	48.7	26.6	4.2
		599.996	600.005	8.82	2.74	
62	63	4.5	2.6	66.2 >>	22.8	3.0
		600.003	600.008	5.46	2.09	
	72	4.5	3.4	43.7	28.1 //	4.7
		500.004	500.010	5.57	1.65	
63	64	4.5	3.2	48.7	26.6	4.2
		600.003	600.006	2.78	0.86	
	72	4.5	3.4	43.7	28.1 //	4.7
		499.995	499.997	2.09	0.62	

Mittel	4.5	2.7	61.8 >>	23.9	3.3
--------	-----	-----	---------	------	-----

<1

LEONARDO FERREZ JUNIOR DER INELUPUNTERE

(mX, mY, a, b in mm; mDr in cm; theta in gon)

Number	mX	mY	mDr	a	b	theta	load
12	1.1	1.3	2.0	1.2	1.1	100.0	6.1
21	2.2	2.4	3.2	2.5	3.1	134.6	4.1
22	1.4	1.4	2.1	1.5	1.4	149.0	2.5
23	1.0	1.0	1.5	1.0	1.0	51.0	2.3
24	1.3	1.4	2.2	1.4	1.3	65.4	4.1
31	2.7	2.1	3.9	2.1	2.1	200.0	4.8
32	1.3	1.2	1.5	1.2	1.2	200.0	2.3
33	0.9	0.9	1.4	0.9	0.9	200.0	2.3
34	0.8	0.7	1.1	0.8	0.7	200.0	2.3
35	1.5	1.2	2.3	1.2	1.2	0.0	4.8
42	1.1	1.0	1.6	1.1	1.0	200.0	2.5
43	1.0	1.0	1.5	1.0	1.0	200.0	2.3
44	1.0	1.0	1.5	1.0	1.0	200.0	2.3
45	1.0	1.0	1.5	1.0	1.0	0.0	2.5
51	3.2	2.5	4.7	3.2	2.5	200.0	4.8
52	1.0	1.0	1.5	1.0	1.0	200.0	2.3
53	1.7	1.7	2.5	1.7	1.7	200.0	2.3
54	1.4	1.2	2.1	1.4	1.3	200.0	2.3
55	2.1	2.9	4.0	2.1	2.1	0.0	4.8
61	3.6	2.9	6.0	4.0	3.5	65.4	4.1
62	2.0	2.1	3.1	2.1	1.9	51.0	2.5
63	2.1	2.0	3.1	2.2	2.0	149.0	2.5
64	1.8	2.0	3.0	2.0	1.8	134.6	4.1
72	2.8	2.2	5.6	3.2	2.8	100.0	6.1

2

m	=	0.61
m	=	0.61
m	=	0.75
m	=	0.57
m	=	0.56
m	=	0.63
m	=	0.45
1m	=	0.51
m	=	1.96
m	=	1.80
m	=	1.97
m	=	1.70
m	=	1.73
m	=	1.73
m	=	1.76
m	=	1.73
m	=	1.73
m	=	1.58
1m	=	1.84
m	=	0.98
m	=	0.91
m	=	0.99
m	=	1.01
m	=	0.99
m	=	0.98
m	=	0.99
m	=	0.98
m	=	0.99
m	=	0.95
m	=	0.98
m	=	0.98
m	=	0.92
m	=	0.96
m	=	0.98
m	=	0.98
m	=	0.94
1m	=	1.14
m	=	0.69
m	=	0.62
m	=	0.69
m	=	0.70
m	=	0.69
m	=	0.69
m	=	0.70
m	=	0.66
m	=	0.69
m	=	0.69
m	=	0.68
m	=	0.66
m	=	0.69
m	=	0.68
m	=	0.68
m	=	0.66
1m	=	0.80
m	=	0.71
m	=	0.84
m	=	0.71
m	=	0.91
m	=	0.71
m	=	0.71
m	=	0.71
m	=	0.71
m	=	0.72
m	=	0.65
1m	=	0.75
m	=	1.22
m	=	1.56
m	=	1.32
m	=	1.71

R	12	22	359.0340	0.9	0.000	0.000	0.0012
R	12	23	40.9670	-0.9	0.000	0.000	-0.0012
R	22	12	159.0339	-2.9	-0.001	-0.002	0.0017
R	23	12	240.9670	0.9	-0.000	0.001	-0.0005
S	12	22	499.9950	-0.5	-0.000	0.000	-0.0002
S	12	23	500.0040	1.7	0.001	0.001	-0.0008
			Pkt:	12	0.000	0.000	-0.0000
R	21	22	100.0012	7.7	-0.002	-0.002	0.0055
R	21	31	359.0342	3.5	0.001	0.000	0.0028
R	21	32	40.9661	-11.2	0.001	-0.001	-0.0047
R	22	21	299.9992	-5.4	-0.001	-0.000	0.0005
R	31	21	159.0328	-8.0	-0.001	-0.002	0.0009
R	32	21	240.9670	2.6	-0.001	0.001	-0.0007
S	21	22	600.0030	6.4	-0.000	0.003	-0.0019
S	21	31	499.9950	-2.1	-0.001	0.001	-0.0006
S	21	32	500.0040	1.6	0.000	0.000	-0.0001
			Pkt:	21	0.000	-0.000	0.0000
R	12	22	359.0340	0.9	-0.000	-0.000	0.0000
R	21	22	100.0012	7.7	0.001	-0.000	0.0001
R	22	12	159.0339	-2.9	0.000	0.000	-0.0010
R	22	21	299.9992	-5.4	-0.001	0.000	-0.0018
R	22	23	100.0008	6.1	-0.001	-0.000	0.0016
R	22	32	359.0339	0.0	0.000	0.000	0.0000
R	22	33	40.9675	2.2	-0.000	0.000	0.0005
R	23	22	299.9992	-5.4	0.001	-0.000	0.0001
R	32	22	159.0339	-1.9	-0.000	-0.000	0.0000
R	33	22	240.9670	-0.6	0.000	-0.000	0.0000
S	12	22	499.9950	-0.5	0.000	-0.000	0.0000
S	21	22	600.0030	6.4	0.000	-0.001	0.0002
S	22	23	600.0030	3.9	-0.000	0.001	-0.0001
S	22	32	499.9950	-1.8	-0.000	0.000	-0.0001
S	22	33	500.0040	1.0	0.000	0.000	0.0000
			Pkt:	22	0.000	-0.000	0.0000
R	12	23	40.9670	-0.9	-0.000	0.000	-0.0000
R	22	23	100.0008	6.1	0.001	0.000	-0.0001
R	23	12	240.9670	0.9	0.000	-0.000	0.0003
R	23	22	299.9992	-5.4	-0.001	0.000	-0.0014
R	23	24	100.0008	0.2	-0.000	-0.000	0.0001
R	23	33	359.0339	1.1	0.000	0.000	0.0003
R	23	34	40.9675	3.2	-0.000	0.000	0.0009
R	24	23	299.9996	-3.8	0.000	0.000	-0.0001
R	33	23	159.0339	-1.2	-0.000	-0.000	0.0000
R	34	23	240.9670	0.8	-0.000	0.000	-0.0000
S	12	23	500.0040	1.7	-0.000	-0.000	0.0001
S	22	23	600.0030	3.9	-0.000	-0.001	0.0001
S	23	24	600.0040	0.6	0.000	0.000	-0.0000
S	23	33	499.9950	-1.3	-0.000	0.000	0.0000
S	23	34	500.0040	1.7	0.000	0.000	-0.0001
			Pkt:	23	0.000	0.000	-0.0000
R	23	24	100.0008	0.2	0.000	0.000	-0.0000
R	24	23	299.9996	-3.8	-0.001	0.001	-0.0027
R	24	34	359.0333	-0.2	-0.000	-0.000	-0.0001
R	24	35	40.9669	4.0	-0.002	0.000	0.0032
R	34	24	159.0339	0.5	0.000	0.000	-0.0001
R	35	24	240.9660	-5.1	0.001	-0.001	0.0004
S	23	24	600.0040	0.6	-0.000	-0.000	0.0002
S	24	34	500.0040	0.2	0.000	-0.000	0.0000
S	24	35	500.0010	2.0	0.001	0.001	-0.0006
			Pkt:	24	-0.000	-0.000	0.0000
R	21	31	359.0342	3.5	-0.001	-0.001	-0.0007
R	21	32	159.0328	-8.0	-0.000	0.002	0.0025
R	21	33	100.0006	7.6	-0.001	0.000	0.0004
R	21	34	40.9675	6.4	0.000	0.000	0.0000

31	32	600.0030	1.6	-0.000	0.000	-0.0000	m=	1.21	S	35	45	500.0020	-0.3	-0.000	0.000	0.0000	m=	0.71
31	42	500.0040	-3.1	-0.001	-0.001	-0.0011	m=	1.23				Pkt:	35	-0.000	0.000	0.0000	m=	0.64
		Pkt:	31	-0.000	0.000	-0.0000	m* =	1.11									1m* =	0.75
							1m* =	1.29	R	31	42	40.9675	0.4	0.000	-0.000	0.0000	m=	0.74
21	32	40.9661	-11.2	-0.001	0.001	0.0000	m=	0.82	R	32	42	359.0339	-1.7	0.000	0.000	0.0000	m=	0.74
22	32	359.0339	0.0	-0.000	-0.000	-0.0000	m=	0.94	R	42	31	240.9670	-2.6	-0.000	0.000	-0.0009	m=	0.76
31	32	100.0006	7.6	0.001	-0.000	0.0000	m=	0.88	R	42	32	159.0339	-2.2	0.000	0.000	-0.0006	m=	0.75
32	21	240.9670	2.6	0.000	-0.000	0.0004	m=	0.94	R	42	43	100.0008	7.2	-0.001	0.000	0.0018	m=	0.76
32	22	159.0339	-1.9	0.000	0.000	-0.0005	m=	0.94	R	42	51	359.0339	-2.9	-0.000	-0.000	-0.0010	m=	0.76
32	31	299.9992	-4.0	-0.000	0.000	-0.0009	m=	0.94	R	42	52	40.9675	0.5	-0.000	0.000	0.0001	m=	0.74
32	33	100.0008	5.3	-0.001	0.000	0.0012	m=	0.94	R	43	42	299.9992	-5.5	0.001	-0.000	0.0001	m=	0.69
32	42	359.0339	-1.7	-0.000	-0.000	-0.0004	m=	0.94	R	51	42	159.0339	-3.8	-0.000	-0.000	-0.0001	m=	0.73
32	43	40.9675	-0.4	0.000	-0.000	-0.0001	m=	0.94	R	52	42	240.9670	-3.0	0.000	-0.000	0.0001	m=	0.73
33	32	299.9992	-6.4	0.001	-0.000	0.0000	m=	0.90	S	31	42	500.0040	-3.1	0.000	0.000	0.0001	m=	0.72
42	32	159.0339	-2.2	-0.000	-0.000	-0.0000	m=	0.94	S	32	42	499.9950	0.2	-0.000	0.000	-0.0000	m=	0.74
43	32	240.9670	-2.5	0.000	-0.000	-0.0000	m=	0.94	S	42	43	600.0030	1.5	-0.000	0.000	-0.0000	m=	0.74
21	32	500.0040	1.6	-0.000	-0.000	-0.0000	m=	0.94	S	42	51	499.9950	1.3	0.000	-0.000	0.0000	m=	0.74
22	32	499.9950	-1.8	0.000	-0.000	-0.0000	m=	0.94	S	42	52	500.0040	-1.1	-0.000	-0.000	-0.0000	m=	0.74
31	32	600.0030	1.6	-0.000	-0.000	-0.0000	m=	0.94				Pkt:	42	-0.000	0.000	-0.0000	m=	0.71
32	33	600.0030	1.3	0.000	0.000	0.0000	m=	0.94								1m* =	0.85	
32	42	499.9950	0.2	0.000	-0.000	-0.0000	m=	0.94	R	32	43	40.9675	-0.4	-0.000	0.000	0.0000	m=	0.74
32	43	500.0040	0.1	0.000	0.000	0.0000	m=	0.94	R	33	43	359.0339	1.1	-0.000	-0.000	-0.0000	m=	0.74
		Pkt:	32	0.000	-0.000	-0.0000	m* =	0.91	R	42	43	100.0008	7.2	0.001	-0.000	-0.0000	m=	0.67
							1m* =	1.06	R	43	32	240.9670	-2.5	-0.000	0.000	-0.0006	m=	0.75
22	33	40.9675	2.2	0.000	-0.000	-0.0000	m=	0.68	R	43	33	159.0339	-0.3	0.000	0.000	-0.0001	m=	0.74
23	33	359.0339	1.1	-0.000	-0.000	-0.0000	m=	0.68	R	43	42	299.9992	-5.5	-0.001	0.000	-0.0012	m=	0.75
32	33	100.0008	5.3	0.000	-0.000	0.0000	m=	0.64	R	43	44	100.0008	6.1	-0.001	0.000	0.0014	m=	0.75
33	22	240.9670	-0.6	-0.000	0.000	-0.0001	m=	0.68	R	43	52	359.0339	0.0	0.000	-0.000	0.0000	m=	0.74
33	23	159.0339	-1.2	0.000	0.000	-0.0003	m=	0.69	R	43	53	40.9675	2.2	-0.000	0.000	0.0005	m=	0.75
33	32	299.9992	-6.4	-0.001	0.000	-0.0014	m=	0.69	R	44	43	299.9992	-6.4	0.001	-0.000	-0.0000	m=	0.68
33	34	100.0008	6.0	-0.001	0.000	0.0013	m=	0.69	R	52	43	159.0339	-1.8	-0.000	-0.000	-0.0000	m=	0.74
33	43	359.0339	1.1	0.000	0.000	0.0003	m=	0.69	R	53	43	240.9670	-1.3	0.000	-0.000	-0.0000	m=	0.74
33	44	40.9675	1.1	-0.000	0.000	0.0003	m=	0.69	S	32	43	500.0040	0.1	-0.000	-0.000	-0.0000	m=	0.74
34	33	299.9992	-5.1	0.000	-0.000	0.0000	m=	0.64	S	33	43	499.9950	-1.7	0.000	-0.000	-0.0000	m=	0.74
43	33	159.0339	-0.3	-0.000	-0.000	-0.0000	m=	0.68	S	42	43	600.0030	1.5	-0.000	-0.000	-0.0000	m=	0.74
44	33	240.9670	-1.9	0.000	-0.000	-0.0000	m=	0.68	S	43	44	600.0030	0.0	0.000	0.000	0.0000	m=	0.74
22	33	500.0040	1.0	-0.000	-0.000	-0.0000	m=	0.68	S	43	52	499.9950	-1.2	-0.000	0.000	0.0000	m=	0.74
23	33	499.9950	-1.3	0.000	-0.000	-0.0000	m=	0.68	S	43	53	500.0040	1.8	0.000	0.000	0.0000	m=	0.74
32	33	600.0030	1.3	-0.000	-0.000	-0.0000	m=	0.68				Pkt:	43	0.000	0.000	0.0000	m=	0.72
33	34	600.0030	2.1	0.000	0.000	0.0000	m=	0.68								1m* =	0.81	
33	43	499.9950	-1.7	-0.000	0.000	0.0000	m=	0.68	R	33	44	40.9675	1.1	0.000	-0.000	-0.0000	m=	0.78
33	44	500.0040	0.9	0.000	0.000	0.0000	m=	0.68	R	34	44	359.0339	2.8	-0.000	-0.000	-0.0000	m=	0.77
		Pkt:	33	-0.000	-0.000	-0.0000	m* =	0.66	R	43	44	100.0008	6.1	0.001	-0.000	-0.0000	m=	0.73
							1m* =	0.75	R	44	33	240.9670	-1.9	-0.000	0.000	-0.0004	m=	0.78
23	34	40.9675	3.2	0.000	-0.000	-0.0000	m=	0.55	R	44	34	159.0339	-0.1	0.000	0.000	-0.0000	m=	0.78
24	34	359.0339	-0.2	0.000	0.000	0.0000	m=	0.57	R	44	43	299.9992	-6.4	-0.001	0.000	-0.0014	m=	0.78
33	34	100.0008	6.0	0.001	-0.000	-0.0000	m=	0.50	R	44	45	100.0008	4.9	-0.001	0.000	0.0011	m=	0.78
34	23	240.9670	0.8	0.000	-0.000	0.0002	m=	0.57	R	44	53	359.0339	4.9	0.000	0.001	0.0012	m=	0.79
34	24	159.0339	0.5	-0.000	-0.000	0.0001	m=	0.57	R	44	54	40.9675	-1.5	0.000	-0.000	-0.0005	m=	0.78
34	33	299.9992	-5.1	-0.001	0.000	-0.0011	m=	0.57	R	45	44	299.9992	-4.7	0.000	-0.000	-0.0000	m=	0.75
34	35	100.0008	-0.5	0.000	-0.000	-0.0001	m=	0.57	R	53	44	159.0339	3.0	0.000	0.000	0.0000	m=	0.77
34	44	359.0339	2.8	0.000	0.000	0.0007	m=	0.57	R	54	44	240.9670	-2.8	0.000	-0.000	-0.0000	m=	0.77
34	45	40.9675	1.5	-0.000	0.000	0.0004	m=	0.57	S	33	44	500.0040	0.9	-0.000	-0.000	-0.0000	m=	0.78
35	34	300.0006	2.8	-0.000	-0.000	0.0000	m=	0.56	S	34	44	499.9950	-1.6	0.000	-0.000	-0.0000	m=	0.77
44	34	159.0339	-0.1	-0.000	-0.000	-0.0000	m=	0.57	S	43	44	600.0030	0.0	-0.000	-0.000	-0.0000	m=	0.78
45	34	240.9670	0.0	-0.000	0.000	0.0000	m=	0.57	S	44	45	600.0030	-0.1	-0.000	-0.000	-0.0000	m=	0.78
23	34	500.0040	1.7	-0.000	-0.000	0.0000	m=	0.56	S	44	53	499.9950	-3.3	-0.000	0.000	0.0000	m=	0.76
24	34	500.0040	0.2	-0.000	0.000	0.0000	m=	0.57	S	44	54	500.0040	-2.2	-0.000	-0.000	-0.0000	m=	0.77
33	34	600.0030	2.1	-0.000	-0.000	-0.0000	m=	0.56				Pkt:	44	-0.000	-0.000	0.0000	m=	0.75
34	35	600.0030	0.8	-0.000	0.000	0.0000	m=	0.57								1m* =	0.85	
34	44	499.9950	-1.6	-0.000	0.000	0.0000	m=	0.56	R	34	45	40.9675	1.5	0.000	-0.000	-0.0000	m=	0.71
34	45	500.0040	1.9	0.000	0.000	-0.0000	m=	0.56	R	35	45	359.0340	2.2	-0.000	-0.000	0.0000	m=	0.71
		Pkt:	34	-0.000	-0.000	-0.0000	m* =	0.55	R	44	45	100.0008	4.9	0.001	0.000	-0.0001	m=	0.67
							1m* =	0.64	R	45	34	240.9670	0.0	0.000	-0.000	0.0000	m=	0.71
24	35	40.9669	4.0	0.001	-0.001	-0.0008	m=	0.72	R	45	35	159.0339	-0.6	0.000	0.000	-0.0002	m=	0.71
34	35	100.0008	-0.5	-0.000	-0.000	0.0001	m=	0.71	R	45	44	299.9992	-4.7	-0.001	0.000	-0.0012	m=	0.72
35	24	240.9660	-5.1	0.000	0.001	-0.0030	m=	0.94	R	45	54	359.0339	1.2	0.000	0.000	0.0003	m=	0.71
35	34	300.0006	2.8	0.000	0.000	0.0012	m=	0.73	R	45	55	40.9675	4.1	-0.001	0.001	0.0014	m=	0.75
35	45	359.0340	2.2	-0.000	0.001	0.0015	m=	0.76	R	54	45	159.0339	-1.5	-0.000	-0.000	0.0000	m=	0.71

44	45	600.0030	-0.1	0.000	0.000	-0.0000	m=	0.71	S	54	63	499.9950	-7.3	-0.001	0.001	-0.0000	m=	0.96
45	54	499.9950	0.7	-0.000	0.000	0.0000	m=	0.70	S	54	64	500.0040	1.0	0.000	0.000	0.0000	m=	1.05
45	55	500.0040	4.0	0.001	0.000	-0.0001	m=	0.68				Pkt:	54	0.000	0.000	0.0000	m=	0.99
		FKt:	45	0.000	0.000	0.0000	m*	0.88									1m*	1.10
							1m*	0.82	R	45	55	40.9675	4.1	0.001	-0.001	-0.0009	m=	0.96
42	51	359.0339	-2.9	0.001	0.001	0.0008	m=	1.46	R	54	55	100.0008	3.0	0.001	0.000	-0.0008	m=	0.96
51	42	159.0330	-3.8	-0.000	0.001	-0.0026	m=	1.53	R	55	45	240.9660	-4.8	0.000	0.001	-0.0033	m=	1.12
51	52	99.9992	1.6	-0.000	-0.000	0.0007	m=	1.46	R	55	54	299.9997	0.8	0.000	-0.000	0.0003	m=	0.96
51	61	40.9665	2.3	0.000	0.001	0.0016	m=	1.48	R	55	64	359.0339	4.0	-0.000	0.001	0.0027	m=	1.03
52	51	299.9992	-5.0	0.002	-0.000	0.0014	m=	1.46	R	64	55	159.0340	1.7	0.000	0.000	-0.0004	m=	0.96
61	51	240.9659	-7.6	0.002	-0.002	0.0016	m=	1.49	S	45	55	500.0040	4.0	-0.002	-0.001	0.0014	m=	1.00
42	51	499.9950	1.3	-0.001	0.000	-0.0005	m=	1.46	S	54	55	600.0030	3.1	0.000	-0.001	0.0000	m=	0.96
51	52	600.0040	3.7	0.000	0.001	0.0000	m=	1.45	S	55	64	499.9970	-2.3	-0.001	0.000	0.0008	m=	0.97
51	61	499.9930	-9.0	-0.004	-0.002	-0.0032	m=	1.58				Pkt:	55	-0.000	-0.000	0.0000	m=	0.88
		FKt:	51	0.000	-0.000	0.0000	m*	1.33									1m*	1.03
							1m*	1.55	R	51	61	40.9665	2.3	0.000	-0.001	-0.0002	m=	1.97
42	52	40.9675	0.5	0.000	-0.000	-0.0000	m=	0.74	R	52	61	359.0339	3.4	-0.001	-0.001	-0.0010	m=	1.96
43	52	359.0339	0.0	-0.000	-0.000	-0.0000	m=	0.74	R	61	51	240.9659	-7.6	-0.002	0.001	-0.0062	m=	2.25
51	52	99.9992	1.6	0.000	-0.000	-0.0000	m=	0.74	R	61	52	159.0329	-3.9	0.000	0.000	-0.0016	m=	2.00
52	42	240.9670	-3.0	-0.000	0.000	-0.0007	m=	0.74	R	61	62	100.0006	11.5	-0.002	0.002	0.0083	m=	2.40
52	43	159.0339	-1.8	0.000	0.000	-0.0004	m=	0.74	R	62	61	299.9992	-3.7	0.001	0.000	0.0003	m=	1.96
52	51	299.9992	-5.0	-0.001	0.000	-0.0011	m=	0.74	S	51	61	499.9930	-9.0	0.000	0.002	0.0008	m=	2.05
52	53	100.0008	5.4	-0.001	0.000	0.0012	m=	0.74	S	52	61	499.9950	-0.6	0.000	-0.000	-0.0000	m=	1.97
52	61	359.0339	3.4	0.000	0.000	0.0008	m=	0.75	S	61	62	599.9960	-8.8	-0.000	-0.004	-0.0026	m=	2.03
52	62	40.9675	1.0	-0.000	0.000	0.0002	m=	0.74				FKt:	61	-0.000	0.000	-0.0000	m*	1.80
53	52	299.9992	-7.3	0.001	-0.000	-0.0000	m=	0.66								1m*	2.10	
61	52	159.0329	-3.9	-0.000	-0.000	-0.0000	m=	0.72	R	52	62	40.9675	1.0	0.000	-0.000	-0.0000	m=	1.38
62	52	240.9670	-2.6	0.000	-0.000	-0.0000	m=	0.73	R	53	62	359.0339	-1.4	0.000	0.000	0.0001	m=	1.38
42	52	500.0040	-1.1	0.000	0.000	0.0000	m=	0.74	R	61	62	100.0006	11.5	0.001	0.000	0.0002	m=	1.27
43	52	499.9950	-1.2	0.000	-0.000	-0.0000	m=	0.74	R	62	52	240.9670	-2.6	-0.000	0.000	-0.0008	m=	1.37
51	52	600.0040	3.7	-0.000	-0.000	-0.0000	m=	0.72	R	62	53	159.0339	-3.2	0.000	0.000	-0.0008	m=	1.39
52	53	600.0030	2.3	0.000	0.000	0.0000	m=	0.73	R	62	61	299.9992	-3.7	-0.001	-0.000	-0.0013	m=	1.40
52	61	499.9950	-0.6	-0.000	0.000	0.0000	m=	0.74	R	62	63	100.0008	11.1	-0.001	0.000	0.0029	m=	1.41
52	62	500.0040	-0.7	-0.000	-0.000	-0.0000	m=	0.74	R	62	72	40.9675	-1.6	0.000	-0.000	-0.0005	m=	1.39
		FKt:	52	0.000	-0.000	-0.0000	m*	0.72	R	63	62	299.9992	-0.7	0.000	0.000	0.0000	m=	1.38
							1m*	0.87	R	72	62	240.9672	-3.1	0.000	-0.000	-0.0000	m=	1.38
43	53	40.9675	2.2	0.000	-0.000	-0.0000	m=	1.26	S	52	62	500.0040	-0.7	0.000	0.000	0.0000	m=	1.38
44	53	359.0339	4.9	-0.000	-0.000	-0.0000	m=	1.25	S	53	62	499.9950	-2.3	0.000	-0.000	0.0000	m=	1.38
52	53	100.0008	5.4	0.000	-0.000	-0.0000	m=	1.24	S	61	62	599.9960	-8.8	0.000	0.002	0.0003	m=	1.30
53	43	240.9670	-1.3	-0.000	0.000	-0.0002	m=	1.26	S	62	63	600.0030	-5.5	-0.000	-0.001	-0.0002	m=	1.35
53	44	159.0339	3.0	-0.000	-0.000	0.0007	m=	1.27	S	62	72	500.0040	-5.6	-0.001	-0.001	-0.0003	m=	1.36
53	52	299.9992	-7.3	-0.001	-0.000	-0.0016	m=	1.26				FKt:	62	-0.000	-0.000	-0.0000	m*	1.33
53	54	100.0008	7.5	-0.001	-0.000	0.0017	m=	1.26								1m*	1.61	
53	62	359.0339	-1.4	-0.000	-0.000	-0.0003	m=	1.26	R	53	63	40.9675	-0.5	-0.000	0.000	0.0000	m=	1.42
53	63	40.9675	-0.5	0.000	-0.000	-0.0001	m=	1.26	R	54	63	359.0339	1.8	-0.000	-0.000	-0.0000	m=	1.42
54	53	299.9992	-2.9	0.000	-0.000	-0.0000	m=	1.26	R	62	63	100.0008	11.1	0.001	-0.000	-0.0002	m=	1.31
62	53	159.0339	-3.2	-0.000	-0.000	-0.0000	m=	1.26	R	63	53	240.9670	-3.1	-0.000	0.000	-0.0008	m=	1.42
63	53	240.9670	-3.1	0.000	-0.000	-0.0000	m=	1.26	R	63	54	159.0339	-1.5	0.000	0.000	-0.0004	m=	1.42
43	53	500.0040	1.8	-0.000	-0.000	0.0000	m=	1.26	R	63	62	299.9992	-0.7	-0.000	-0.000	-0.0002	m=	1.42
44	53	499.9950	-3.3	0.000	-0.000	-0.0000	m=	1.25	R	63	64	100.0008	0.5	-0.000	0.000	0.0002	m=	1.42
52	53	600.0030	2.3	0.000	-0.000	-0.0000	m=	1.26	R	63	72	359.0339	4.8	0.000	0.001	0.0016	m=	1.44
53	54	600.0180	8.1	0.000	0.001	0.0000	m=	1.19	R	64	63	300.0004	5.3	-0.001	0.000	0.0001	m=	1.39
53	62	499.9950	-2.3	-0.000	0.000	0.0000	m=	1.26	R	72	63	159.0340	3.1	0.000	0.000	0.0000	m=	1.41
53	63	500.0240	11.5	0.001	0.001	-0.0000	m=	1.11	S	53	63	500.0240	11.5	-0.001	-0.001	0.0000	m=	1.27
		FKt:	53	0.000	-0.000	-0.0000	m*	1.22	S	54	63	499.9950	-7.3	0.001	-0.001	-0.0003	m=	1.37
							1m*	1.39	S	62	63	600.0030	-5.5	-0.000	0.001	0.0002	m=	1.39
44	54	40.9675	-1.5	-0.000	0.000	0.0000	m=	1.03	S	63	64	600.0030	-2.8	0.000	-0.000	-0.0001	m=	1.41
45	54	359.0339	1.2	-0.000	-0.000	-0.0000	m=	1.03	S	63	72	499.9950	-2.1	-0.000	0.000	0.0001	m=	1.41
53	54	100.0008	7.5	0.001	-0.000	0.0000	m=	0.97				FKt:	63	-0.000	0.000	0.0000	m*	1.30
54	44	240.9670	-2.8	-0.000	0.000	-0.0007	m=	1.03								1m*	1.65	
54	45	159.0339	-1.5	0.000	0.000	-0.0004	m=	1.03	R	54	64	40.9675	2.5	0.001	-0.001	-0.0007	m=	1.01
54	53	299.9992	-2.9	-0.000	0.000	-0.0006	m=	1.03	R	55	64	359.0339	4.0	-0.001	-0.001	-0.0004	m=	1.00
54	55	100.0008	3.0	-0.000	0.000	0.0007	m=	1.03	R	63	64	100.0008	0.5	0.000	-0.000	-0.0000	m=	0.99
54	63	359.0339	1.8	0.000	0.000	0.0004	m=	1.03	R	64	54	240.9660	-7.0	-0.001	0.001	-0.0029	m=	1.16
54	64	40.9675	2.5	-0.000	0.000	0.0006	m=	1.03	R	64	55	159.0340	1.7	-0.001	-0.000	0.0014	m=	1.02
55	54	299.9997	0.8	-0.000	0.000	-0.0000	m=	1.02	R	64	63	300.0004	5.3	0.001	0.001	0.0038	m=	1.18
63	54	159.0329	-1.5	-0.000	-0.000	-0.0000	m=	1.03	S	54	64	500.0040	1.0	-0.000	-0.000	-0.0000	m=	0.99
64	54	240.9670	-2.1	0.000	0.000	0.0000	m=	1.03										

4

4

R	63	72	359.0339	4.8	-0.002	-0.003	-0.0029	m=	1.64
R	72	62	240.9672	-3.1	-0.001	-0.000	-0.0042	m=	1.67
R	72	63	159.0340	3.1	-0.001	-0.000	0.0042	m=	1.67
S	62	72	500.0040	-5.6	0.003	0.003	0.0028	m=	1.74
S	63	72	499.9950	-2.1	0.001	-0.001	-0.0011	m=	1.46
			Pkt:	72	0.000	-0.000	-0.0000	m ⁰ =	1.15
								lm ⁰ =	1.31

Verteilung der lm_0 T2FF

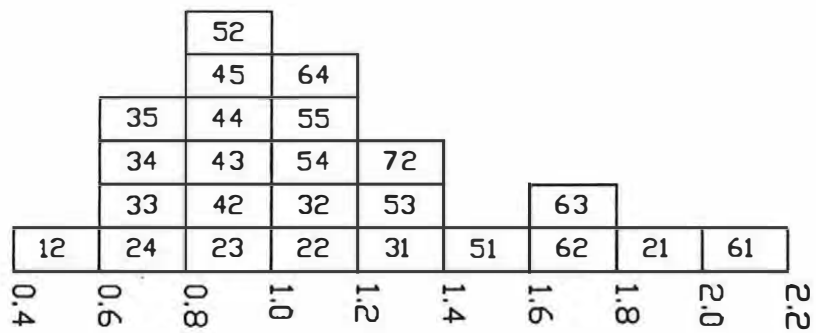


Abb. 8.3

Die Variante T2EB (Abb. 8.4) zeigt eine Datumsvorgabe mit Ersatzbeobachtungen zur Festpunktüberprüfung.

1: Überprüfung der Varianzkomponenten getrennt für x - und y - Koordinatendifferenzen zur Überprüfung des stochastischen Modells der Ersatzbeobachtungen.

2: Die Zuverlässigkeitskennzahlen für Richtungen und Strecken sind dabei nur als Mittelwerte angegeben. Für die Ersatzbeobachtungen sind alle Werte je Beobachtung wiedergegeben.

Die Variante T2HK zeigt die Berechnung der Hauptkomponente. Der Beobachtungsplan wurde dabei ausschließlich auf Streckenbeobachtungen beschränkt.

1: Ausdruck der Anteile der Hauptkomponente und Darstellung in Abb. 8.6.

T2EB
Ersatzbeobachtungen

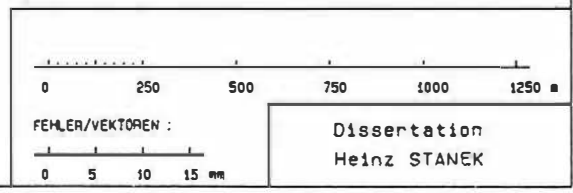
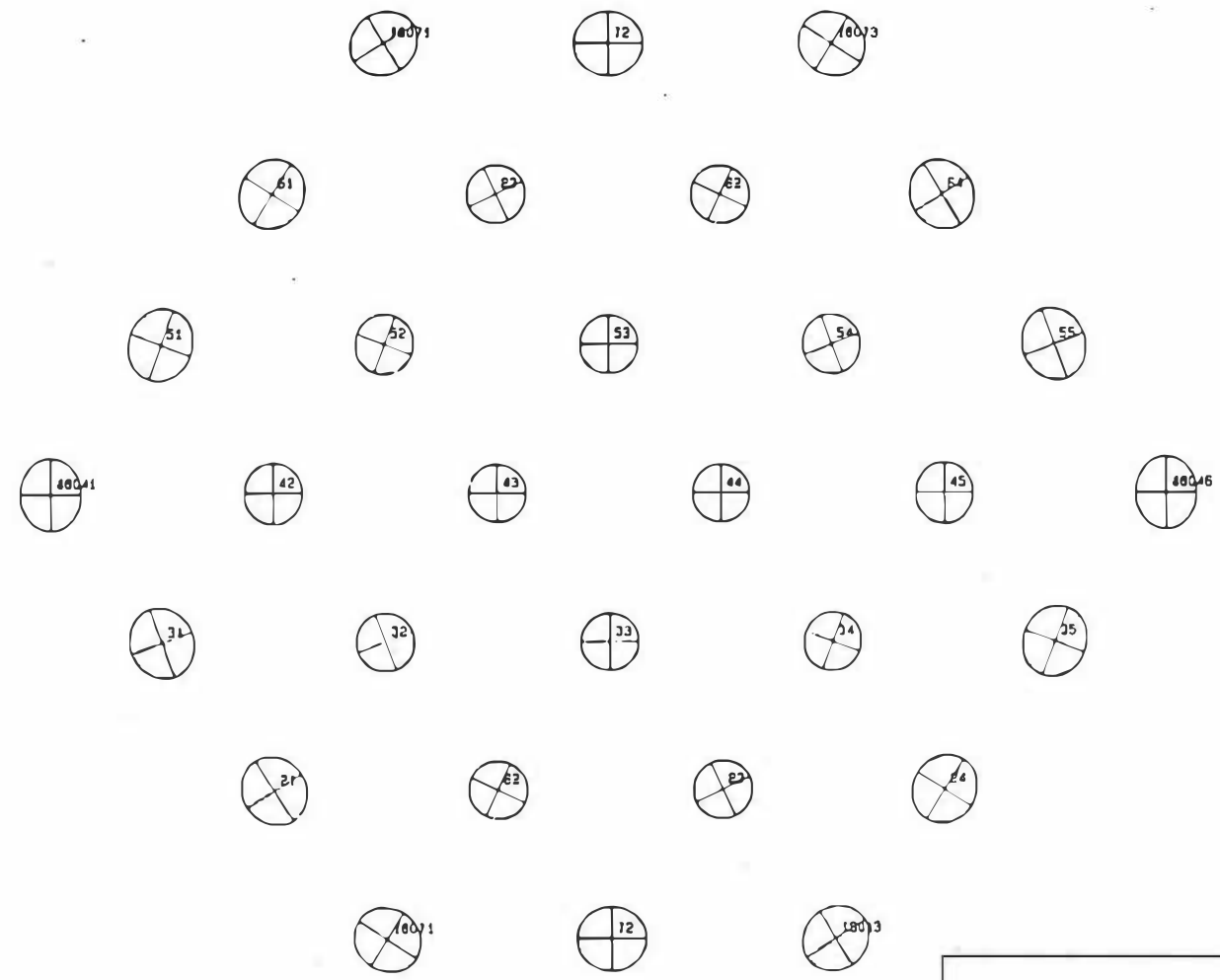


Abb. 8.4

ANALYSE AUSGLEICHUNG Dipl.-Ing. H. STANEK

PROJEKT : T2EB 88/09/01 16:48:54

Kommentar : Testbeispiel 2: Datumsvorgabe mit Ersatzbeobachtungen

Funkte : 36 (6 Festpunkte, 30 Neupunkte)
 Beobachtungen : 225 (142 Richtungen, 71 Strecken)
 (6 x-Differenzen, 6 y-Differenzen)
 Unbekannte : 90 (60 Koordinaten, 30 Orientierungen)
 Redundanz : 135
 Kondition : 8
 Bedingungen : 0

STELLERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MIT R a priori

Gewichtseinheitsfehler ± 1.0
 Festpunkt-Lagefehler ± 0.0 mm
 Richtungsfehler ± 5.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 4.0 mm
 Entfernungsabhängiger Streckenfehler ± 0.0 ppm
 x - Differenzfehler ± 5.0 mm
 y - Differenzfehler ± 5.0 mm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 1.1
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 1.2 cc
 Streckenfehler ± 0.9 mm
 x - Differenzfehler ± 0.4 mm
 y - Differenzfehler ± 0.1 mm

FESTPUNKTE

Nummer	Bezeichnung	Y	X
10011		200.008	
10012		200.014	

T2EB

ZUVERLÄSSIGKEIT DER BEOBSACHTUNGEN

m0/Jp bzw m0*Jq = mittl. Fehler der Beobachtung vor bzw nach dem Ausgleich.
 r = 1-p*q = Redundanzanteil der Beobachtung an der Gesamtredundanz Er.
 GWA = Grenzwert für die Aufdeckung grober Beob.fehler bei α=0.1% und β=80%.
 Ef. = Einflußfaktor (Maß für die äußere Zuverlässigkeit)
 Fehlerwerte und Verbesserungen für Richtungen in cc, für Strecken in mm.
 Der kritische Wert k für die normierten Verbesserungen ist 3.3.

RICHTUNGEN

Anteil vtFv: 109.94
 m0 der Gruppe Richtungen nach der Ausgleichung +/- 1.16 cc

Station	Ziele	m0/Jp Messwert	m0*Jq ausgeglt	r(%) Verb.	GWA w	Ef.
	Mittel	5.3	3.4	57.1	28.9	3.6

STRECKEN

Anteil vtFv: 38.61
 m0 der Gruppe Strecken nach der Ausgleichung +/- 0.90 mm

Station	Ziele	m0/Jp Messwert	m0*Jq ausgeglt	r(%) Verb.	GWA w	Ef.
	Mittel	4.2	2.4	66.6	21.4	2.9

X

Anteil vtFv: 0.50
 m0 der Gruppe x-Differenzen nach der Ausgleichung +/- 0.40 mm

Station	Ziele	m0/Jp Messwert	m0*Jq ausgeglt	r(%) Verb.	GWA w	Ef.
10011	11	5.3	3.4	58.4	28.4	3.5
		0.000	0.001	0.76	0.21	
10012	12	5.3	3.4	58.4	28.4	3.5
		0.000	0.002	1.57	0.46	
10041	41	5.3	3.9	43.5	32.9	4.7
		0.000	-0.000	-0.11	0.03	
10046	46	5.3	3.9	43.5	32.9	4.7
		0.000	0.001	0.98	0.25	
10071	71	5.3	3.4	58.4	28.4	3.5
		0.000	-0.000	-0.20	0.06	
10073	73	5.3	3.4	58.4	28.4	3.5
		0.000	-0.000	-0.07	0.07	

102

2

1

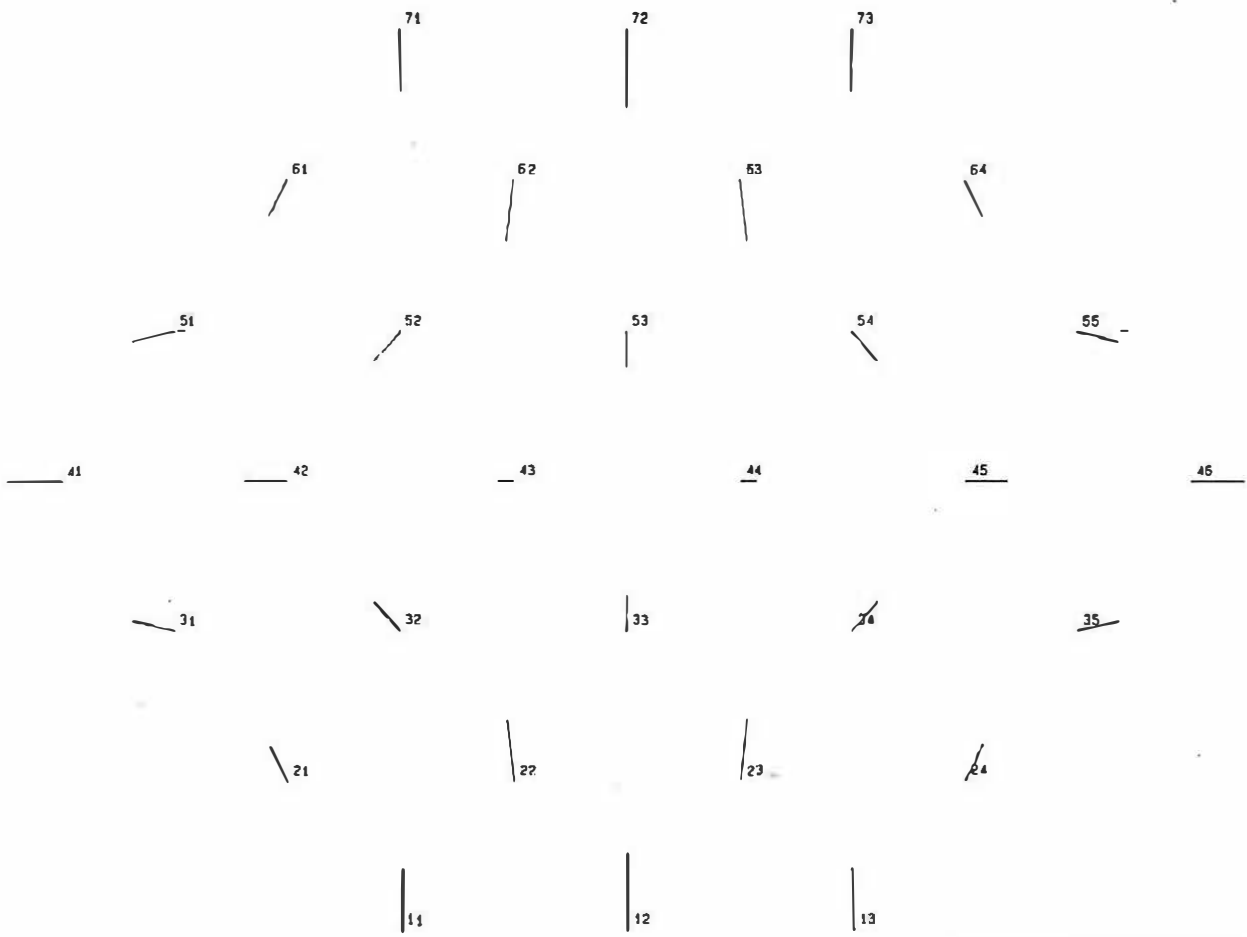
Mittel	5.3	3.6	53.4	29.9	3.9
--------	-----	-----	------	------	-----

Y-DIFFERENZEN

Anteil vtfiv: 0.02
 m0 der Gruppe y-Differenzen nach der Ausgleichung +/- 0.08 mm

Station	Ziele	m0/Jd Messwert	m0/Jd ausgegl	r(%) Verb.	GWA w	Ef.
10011	11	5.3	3.6	53.8	29.6	3.8
		0.000	0.000	0.22	0.06	
10012	13	5.3	3.6	53.8	29.6	3.8
		0.000	-0.001	-0.53	0.15	
10041	41	5.7	3.3	61.3 >>	27.7	3.3
		0.000	0.000	0.10	0.03	
10046	46	5.7	3.3	61.3 >>	27.7	3.3
		0.000	0.000	0.10	0.03	
10071	71	5.3	3.6	53.8	29.6	3.8
		0.000	-0.000	-0.19	0.05	
10073	73	5.3	3.6	53.8	29.6	3.8
		0.000	0.000	0.30	0.08	
Mittel		5.3	3.5	56.3	29.0	3.6

T2HK
Hauptkomponenten



0 250 500 750 1000 1250 m

FEHLER/VEKTOREN :

0 1 2 3 mm

Dissertation
Heinz STANEK

Abb. 8.5

HAUPTKOMPONENTE

< 1

In diese Richtungskomponenten ist das Netz am schlechtesten bestimmt.

	dX 11	dY 11	dX 12	dY 12	dX 13	dY 13	dX 21	dY 21
Hauptk.	-1.31	-0.04	-1.65	0.00	-1.31	0.04	-0.77	0.38
	dX 22	dY 22	dX 23	dY 23	dX 24	dY 24	dX 31	dY 31
Hauptk.	-1.29	0.15	-1.29	-0.15	-0.77	-0.38	-0.21	0.88
	dX 32	dY 32	dX 33	dY 33	dX 34	dY 34	dX 35	dY 35
Hauptk.	-0.62	0.54	-0.77	0.00	-0.62	-0.54	-0.21	-0.68
	dX 41	dY 41	dX 42	dY 42	dX 43	dY 43	dX 44	dY 44
Hauptk.	0.00	1.12	0.00	0.89	0.00	0.33	-0.00	-0.33
	dX 45	dY 45	dX 46	dY 46	dX 51	dY 51	dX 52	dY 52
Hauptk.	-0.00	-0.89	-0.00	-1.12	0.21	0.88	0.62	0.54
	dX 53	dY 53	dX 54	dY 54	dX 55	dY 55	dX 61	dY 61
Hauptk.	0.77	-0.00	0.62	-0.54	0.21	-0.88	0.77	0.38
	dX 62	dY 62	dX 63	dY 63	dX 64	dY 64	dX 71	dY 71
Hauptk.	1.29	0.15	1.29	-0.15	0.77	-0.38	1.31	-0.04
	dX 72	dY 72	dX 73	dY 73				
Hauptk.	1.65	-0.00	1.31	0.04				

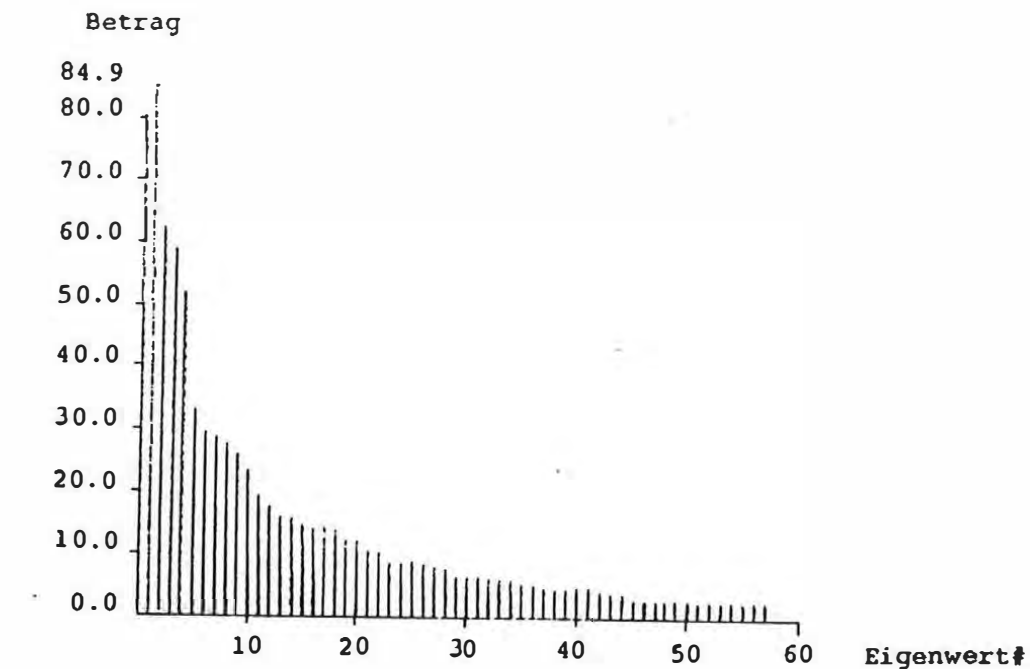


Abb. 8.6

8.2 Datumsüberprüfung EP - Pottendorf

Die ursprüngliche Auswertung erfolgte mit dem angepaßten Programm NETZ 3D (Elmiger 1972, Zalka 1988). Mittels der erwähnten Datenschnittstelle wurden die reduzierten Beobachtungswerte in das Programmsystem ANAG zur Analyse und Interpretation übernommen. Die Abbildung 8.7 zeigt den Beobachtungsplan und die Genauigkeitssituation in Form der Helmert'schen Punktfehlerellipsen für die übernommenen Werte. Alle vergleichbaren Ergebnisse zeigen Übereinstimmung mit der NETZ 3D - Auswertung.

1: Die Unverträglichkeit des stochastischen Modells wird auf Grund der berechneten Varianz der Gewichtseinheit bzw. den Varianzkomponenten aufgezeigt.

Da auf Grund der Bekanntgabe des zuständigen Vermessungsamtes Baden Festpunktverschiebungen auf Grund von äußeren Netzzwängen enthalten sind, wurde ein in Abschnitt 7.2 beschriebener Analyseablauf durchgeführt. In der Tabelle 8.2 werden Informationen zu den verwendeten Festpunkten zusammengefaßt.

Punkt Nr.	Punktname	Ordnung	Operat	Bemerkung
1	77-288	5		neu
2	77-267	3	N484 N36	
3	77-296	4		neu
4	77/1-633	5		neu
5	77/2-633	5		neu
6	77-512	5	N 666	gesperrt
7	77-233	1	B 100	
8	77-791	5	N 679	gesperrt
9	77-343	2	N 666	gesperrt
10	77-292	5		neu
11	77-280	4	N 666	gesperrt
12	77-328	5	N 666	gesperrt
13	77-302	5		neu
14	77-221	3	N 36	
15	76-288	2	N 666	gesperrt
16	76-93	4	N 666	
17	77-307	5		neu
18	77-338	5	N 666	gesperrt
19	77-313	5	N 666	gesperrt
20	77-458	5	N 666	gesperrt
21	EP TA 10	EP		Tattendorf
38	76-123	5	N 666	gesperrt
39	77-305	3	B 65	

Tab. 8.2

POTT
Originalauswertung

Punktverbindungen:
 Richtung >
 Distanz /
 Zenitdistanz ... Z

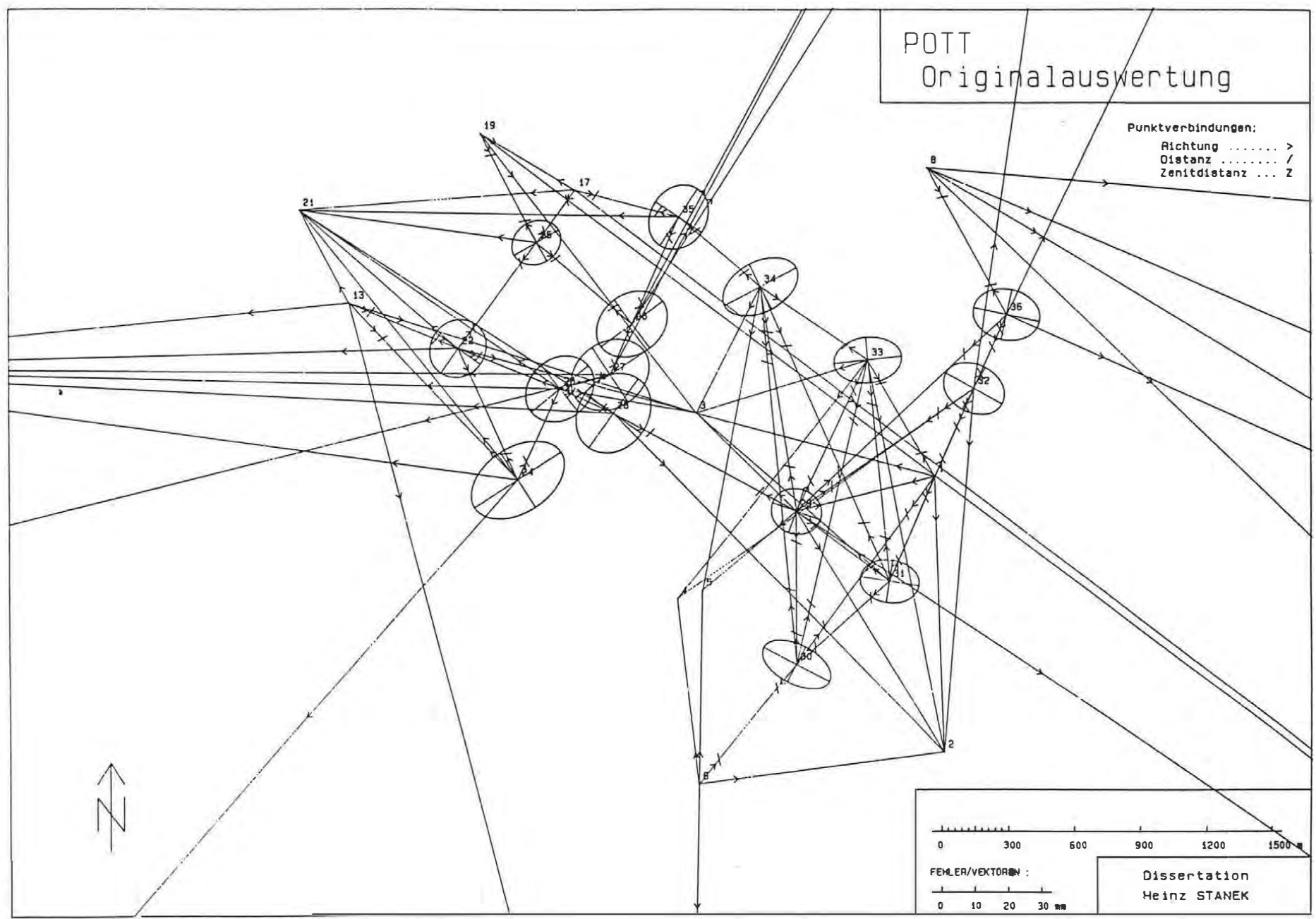


Abb. 8.7

0 300 600 900 1200 1500 m	
FEHLER/VEKTOR :	Dissertation
0 10 20 30 mm	Heinz STANEK

PROJEKT : POTT 88/09/01 18:07:03

Kommentar : EP - Pottendorf : Originalauswertung

Punkte : 37 (22 Festpunkte, 15 Neupunkte)
 Beobachtungen : 201 (132 Richtungen, 69 Strecken)
 Unbekannte : 51 (30 Koordinaten, 21 Orientierungen)
 Redundanz : 150
 Kondition : 11
 Bedingungen : 0

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 3.0
 Festpunkt-Lagefehler ± 5.0 mm
 Richtungsfehler ± 3.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 5.0 mm
 Entfernungsabhängiger Streckenfehler ± 2.0 ppm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 6.9

<1

mit den gruppenweisen Anteilen:

Richtungsfehler ± 6.2 cc
 Streckenfehler ± 6.0 mm

FESTPUNKTE

Nummer	Bezeichnung	Y	X
1	77-288	4050.770	5309153.730
	Or = -0.0011		
2	77-267	4090.010	5307299.130
3	77-296	2975.790	5309441.660
4	77/1-633	2889.380	5308599.290
5	77/2-633	3001.800	5308637.160
6	77-512	2981.680	5307760.590
	Or = 400.0005		
7	77-223	10754.280	5304098.220
8	77-781	4018.440	5310550.130
	Or = 400.0006		
9	77-343	4845.660	5313822.630

Eine Partitionierung des Netzes in einen östlichen (POTTOST) und einen westlichen Netzteil (POTTWES) wurde durchgeführt. Im Netzteil POTTWES wurden alle Festpunkte entfernt und anschließend eine freie Netzausgleichung zur Modellüberprüfung durchgeführt (POTTWES). Da keine groben Beobachtungsfehler aufgedeckt werden konnten und die Varianzkomponentenuntersuchung gute Übereinstimmung aufzeigte, wurde für diesen Netzteil eine Datumsvorgabe mit Ersatzbeobachtungen durchgeführt.

Die Konsequenz war ein deutliches Ansteigen des Varianzfaktors der Gewichtseinheit. Einige Ersatzbeobachtungen wurden als grob fehlerhaft ausgewiesen. Die Eliminierung erfolgte schrittweise, wobei jeweils die x - und die y - Koordinatendifferenz eines Punktes eliminiert wurden. Das bedeutet eine Umwandlung des Festpunktes in einen Neupunkt wenn er von einer größeren Anzahl von Netzbeobachtungen betroffen ist und zum geometrischen Netzaufbau beiträgt. Im anderen Fall werden solche Festpunkte aus der Ausgleichung genommen. Diese Vorgehensweise führte zur Umwandlung der Festpunkte 13, 17 und 19 in Neupunkte.

Die Abbildung 8.8 zeigt die Fehlerellipsen der abschließenden Ausgleichung des Netzteil (POTTWESU).

Dabei konnte die in der freien Ausgleichung überprüfte Annahme des stochastischen Modells auch bei Einbeziehung der festpunktbezogenen Beobachtungen und der Ersatzbeobachtungen bestätigt werden und damit eine Gruppe von verträglichen Datumspunkten gefunden werden.

Für den Ostteil wurde in gleicher Weise verfahren. Die Beobachtung S 30 -33 wurde als fehlerhaft erkannt und nach Überprüfung der Beobachtungsprotokolle eliminiert.

Die Analyse ergab wieder eine Gruppe von verträglichen Datumspunkten (Abb. 8.9).

POTTWESU
Westteil

Punktverbindungen
Richtung >
Distanz /
Zenitdistanz Z

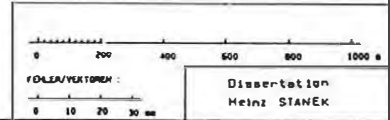
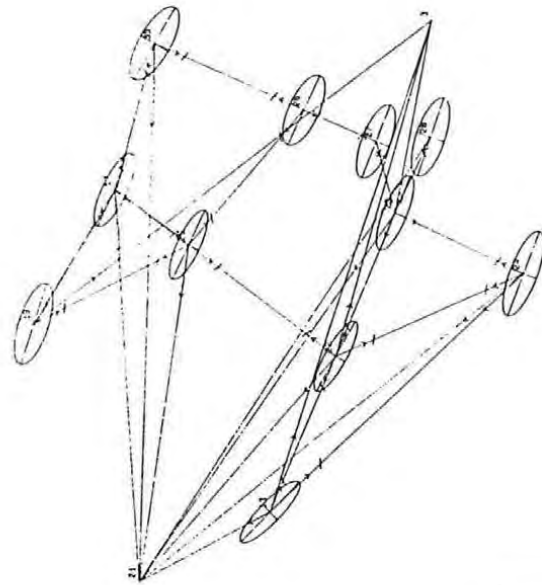


Abb. 8.8

POTTOSTU

Punktverbind.
Richtung >
Distanz /
Zenitdistanz Z

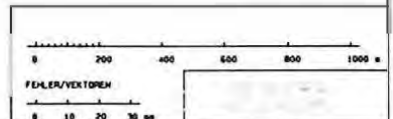
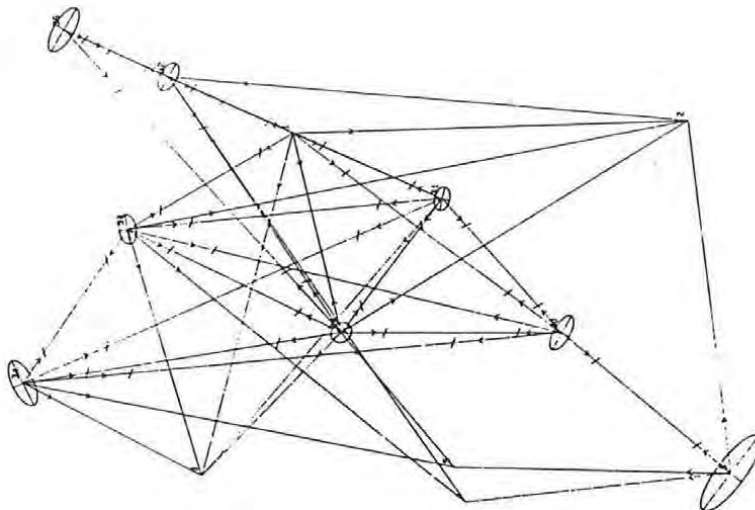


Abb. 8.9

Abschließend wurde zur Überprüfung der Netzpartitionierung eine gemeinschaftliche Analyse des gesamten Datenbestandes durchgeführt. Nach Elimination aller Festpunkte im Gesamtnetz wurde eine freie Ausgleichung zur Modellüberprüfung berechnet. Dabei konnte bezüglich der Elimination von Beobachtungen und Auswahl der Datumsbestimmung jeweils Bestätigung gefunden werden.

Die abschließende Ausgleichung POTTFP (Abb. 8.10) wurde mit hierarchischer Datumsfestsetzung mit den überprüften Datumspunkten 2, 3, 4, 5 und 21 als Festpunkte durchgeführt. Die Punkte 1, 6, 13, 17 und 19 wurden in dieser Version als Neupunkte behandelt.

1: Die Überprüfung der Varianzkomponenten in dieser Version zeigen im Vergleich mit der freien Ausgleichung, daß eine Beeinflussung durch Festpunkte nicht zu erwarten ist.

POTTFFP
Endgeult. Datumsvorg.

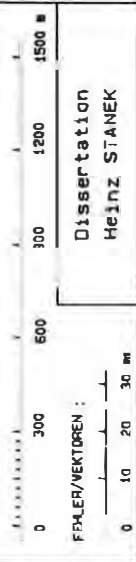
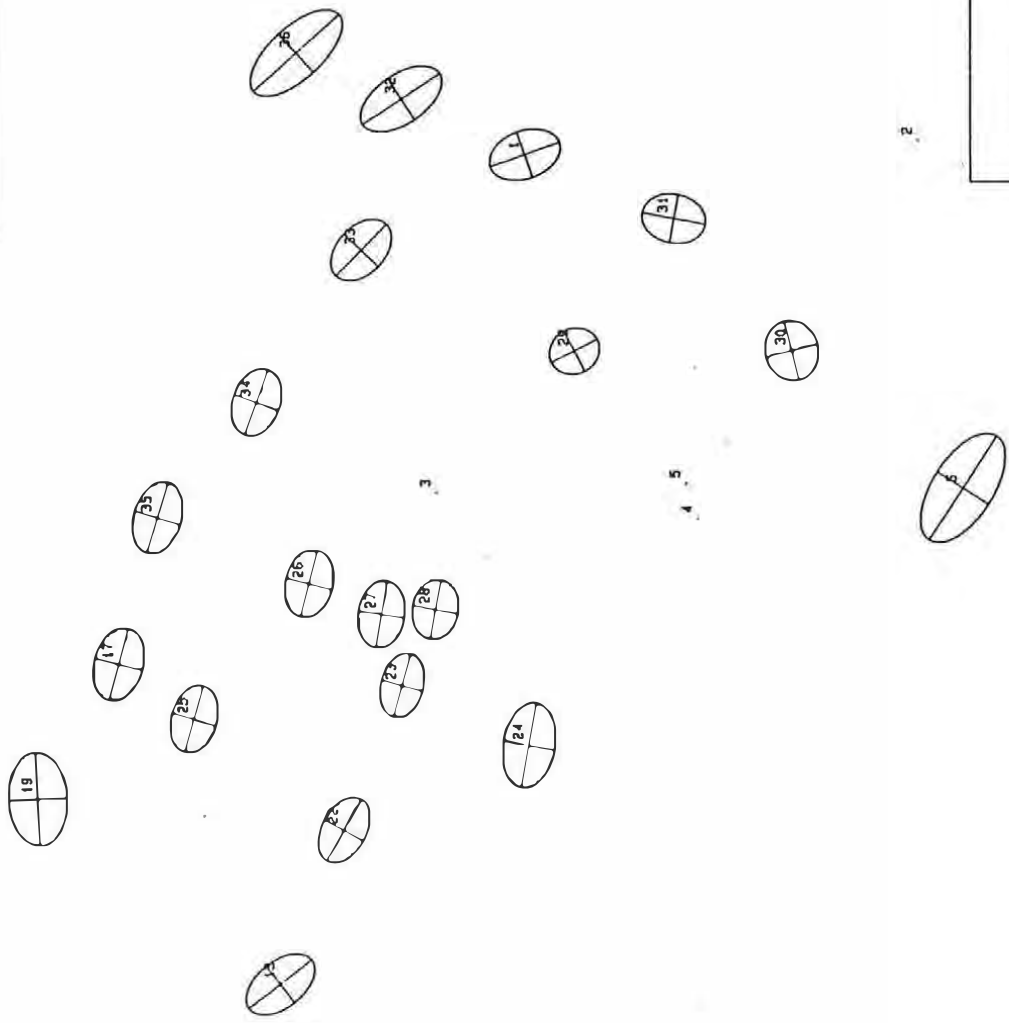


Abb. 8.10

PROJEKT : POTTFF 88/09/01 17:26:54

Kommentar : Pottendorf mit FP 2,3,4,5,21 25-08-1988

Punkte : 25 (5 Festpunkte, 20 Neupunkte)
 Beobachtungen : 169 (103 Richtungen, 66 Strecken)
 Unbekannte : 60 (40 Koordinaten, 20 Orientierungen)
 Redundanz : 109
 Kondition : 57
 Bedingungen : 0

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 1.0
 Festpunkt-Lagefehler ± 0.0 mm
 Richtungsfehler ± 7.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 5.0 mm
 Entfernungsabhängiger Streckenfehler ± 2.0 ppm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 1.0
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 1.0 cc
 Streckenfehler ± 1.0 mm

← 1

FESTPUNKTE

Nummer	Bezeichnung	X	Y
2	77-267	4090.010	5307899.130
3	77-296	2975.790	5309441.660
4	77/1-633	2889.380	5308599.290
5	77/2-633	3001.800	5308637.160
21	EF_TA_10	1187.910	5310359.230

8.3 Zuverlässigkeitsüberprüfung B - 79 Sieggraben

Dieses Netz wurde als eines der vom BA f.E u.Vw zur Verfügung gestellten und untersuchten Operate zur Analyse ausgewählt, da sich teilweise interessante Details ergeben. Es wurde hier die spezielle, in Abschnitt 2.2 erwähnte direkte Gewichtsvergabe angewendet. Damit konnte die Übereinstimmung vergleichbarer Werte mit dem dort verwendeten Programm NETZ 77 (Stubenvoll, Meissl 1977) überprüft und kontrolliert werden. Abbildung 8.11 zeigt die Darstellung der Fehlerellipsen.

1: Deutlich wird bei der Analyse der Zuverlässigkeitskennzahlen die Problematik der Einbeziehung von Exzentermessungen in die Ausgleichung. Diese sind im allgemeinen durch die anderen Beobachtungen kaum zu kontrollieren. Besonders die GWA - Werte zeigen deutlich die mögliche Beträge auf. Im Programm NETZ 77 (Stubenvoll, Meissl 1977) können solche Beobachtungsformen zu "lokalen Systemen" zusammengefaßt und als Ganzes in die Ausgleichung einbezogen werden.

B-79BA
B-79 Sieggraben

Punktverbindungen:

Richtung >
Distanz /
Zenitdistanz ... z

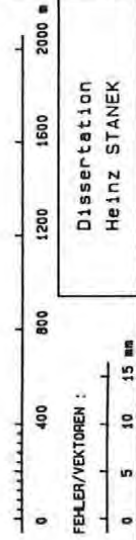
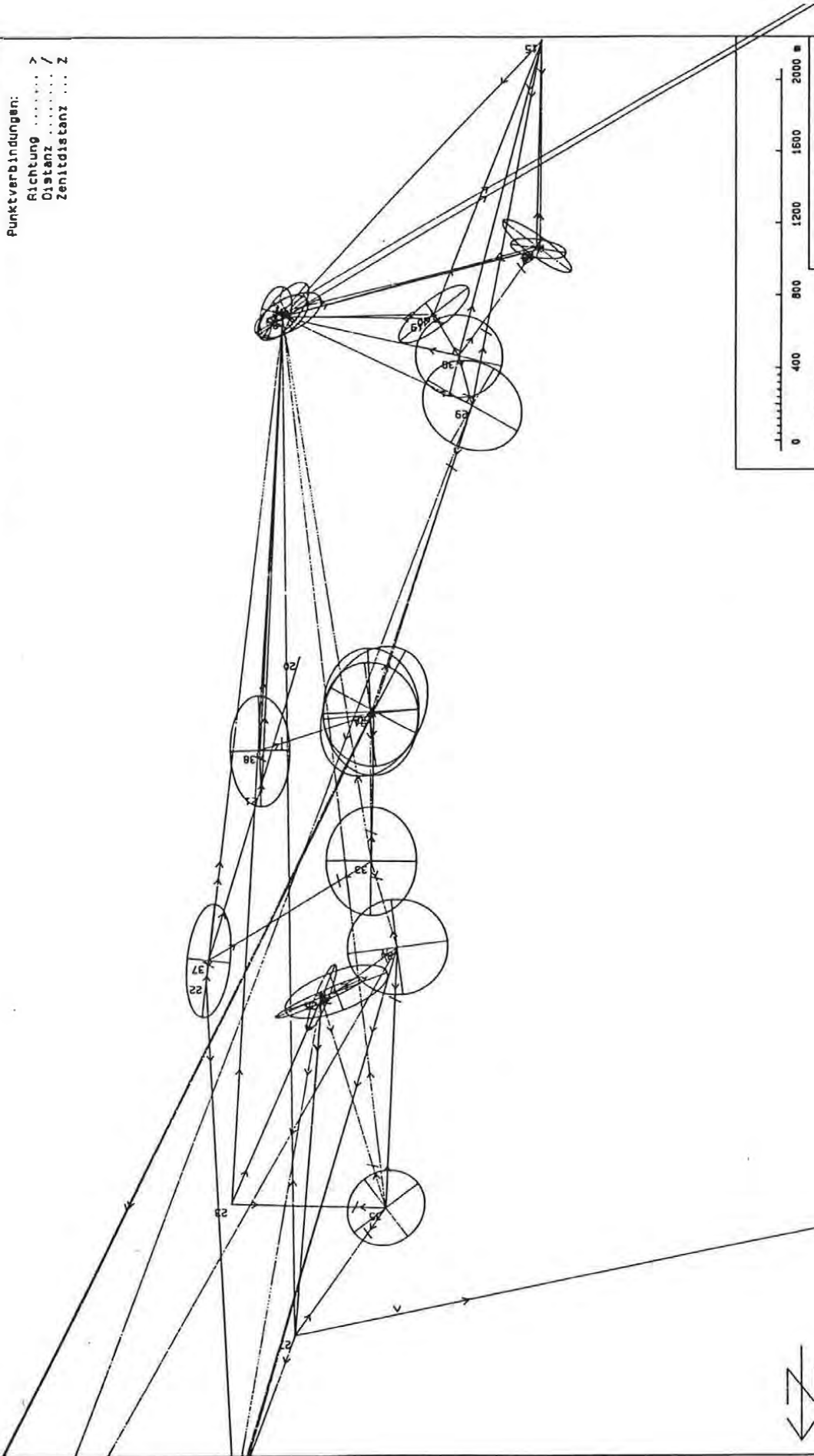


Abb. 8.11

ZUVERLÄSSIGKEIT DER BEOBSACHTUNGEN

m0/Jp bzw m0*Jq = mittl. Fehler der Beobachtung vor bzw nach dem Ausgleich.
 r = 1-p*q = Redundanzanteil der Beobachtung an der Gesamtredundanz Er.
 GWA = Grenzwert für die Aufdeckung grober Beob.fehler bei α=0.1% und β=80%.
 Ef. = Einflußfaktor (Maß für die äußere Zuverlässigkeit)
 Fehlerwerte und Verbesserungen für Richtungen in cc, für Strecken in mm.
 Der kritische Wert k für die normierten Verbesserungen ist 3.3.

RICHTUNGEN

Anteil vtPvt: 33.64
 m0 der Gruppe Richtungen nach der Ausgleichung +/- 0.74 cc

Station	Ziele	m0/Jp Messwert	m0*Jq ausgegl	r(%) Verb.	GWA w	Ef.
7	4	5.0	3.5	52.5	28.7	3.9
		181.5163	181.5168	4.62	1.33	
	8	34.2	21.7	59.8	182.5	3.4
		175.7313	175.7298	-15.06	0.70	
	9	34.9	23.1	56.1	192.4	3.7
		126.8046	126.8041	-5.21	0.23	
	10	28.7	14.7	73.8 >>	138.0	2.5
		159.5716	159.5708	-8.00	0.54	
	11	28.0	15.4	69.7 >>	138.5	2.7
		181.9225	181.9225	-0.43	0.03	
9	14	4.9	3.5	49.7	28.6	4.2
		117.2059	117.2055	-3.71	1.07	
	4	5.0	3.5	52.3	28.7	3.9
		269.4763	269.4766	3.32	0.96	
	7	34.9	23.2	55.7	192.9	3.7
		14.4545	14.4541	-4.10	0.18	
	8	39.8	33.5	29.4 <<	303.4 //	6.4
		335.7388	335.7401	12.88	0.36	
	10	39.5	24.4	61.7 >>	207.4	3.3
		290.6231	290.6226	-4.94	0.20	
10	11	32.1	16.2	74.6 >>	153.4	2.4
		314.1713	314.1702	-10.76	0.66	
	14	4.9	3.5	49.2	28.8	4.2
		204.8400	204.8397	-2.92	0.84	
	7	28.7	15.0	72.7 >>	139.0	2.5
		19.3601	19.3610	8.89	0.59	
	8	47.6	42.3	20.8 <<	430.8 @@	8.1
		386.9925	386.9931	6.07	0.14	
	9	39.5	24.6	61.3 >>	208.2	3.3
		62.7627	62.7628	0.84	0.03	
11	11	47.7	32.0	55.1	265.6	3.7
		321.7922	321.7923	0.74	0.02	
	14	4.9	4.0	34.1	34.5 //	5.7
		176.8996	176.8996	-0.32	0.08	
	16	6.9	4.1	64.6 >>	35.7	3.1
		260.3747	260.3747	-0.04	0.01	
	7	28.0	15.8	68.3 >>	139.9	2.8
		237.2906	237.2931	24.54	1.56	
	8	48.1	39.0	34.3	339.3 //	5.7

		281.8898	281.8908	10.11	0.62	
10		47.7	31.0	55.0	269.7	3.7
		317.3721	317.3727	5.82	0.16	
14		4.9	3.9	34.9	34.2 //	5.6
		372.4374	372.4371	-2.67	0.68	
16		7.0	4.1	65.6 >>	35.6	3.0
		54.8342	54.8346	3.81	0.93	
15	8	5.9	3.4	67.2 >>	29.5	2.9
		350.4863	350.4861	-2.14	0.64	
16		7.8	3.3	81.5 >>	35.5	2.0
		303.2718	303.2718	-0.35	0.10	
29		5.9	3.4	66.9 >>	29.7	2.9
		314.5333	314.5335	1.79	0.53	
39		6.2	3.4	70.4 >>	30.7	2.7
		318.8189	318.8190	0.63	0.19	
16	8	6.9	5.2	43.9	43.1 //	4.7
		391.7321	391.7323	2.47	0.48	
15		7.8	5.2	55.4	43.1	3.7
		108.6154	108.6151	-2.82	0.54	
17		56.1	39.0	51.6	322.5	4.0
		50.1159	50.1144	-14.97	0.38	
18		81.0	80.1	2.2 <<	2244.9 @@	27.4 ##
		12.5809	12.5808	-1.29	0.02	
17	8	6.9	4.9	50.1	40.5	4.1
		155.6575	155.6577	2.30	0.47	
15		7.8	5.8	45.3	47.9 //	4.5
		274.2988	274.2987	-1.33	0.23	
16		56.1	39.3	50.8	324.9	4.1
		14.8149	14.8148	-0.97	0.02	
18		69.1	68.8	0.9 <<	3050.4 @@	43.9 ##
		41.3228	41.3227	-0.69	0.01	
39		9.7	6.4	56.4	53.6	3.6
		111.7871	111.7869	-2.41	0.38	
19	8	9.3	9.2	2.6 <<	237.4 @@	25.1 ##
		8.0554	8.0556	2.12	0.23	
40		40.8	28.7	50.4	237.4	4.1
		164.8207	164.8166	-40.61	1.41	
21	8	5.6	5.5	5.4 <<	99.3 @@	17.2 ##
		171.8299	171.8298	-1.11	0.20	
38		17.6	12.0	53.9	99.3	3.8
		166.5345	166.5356	11.03	0.92	
22	8	5.3	3.8	48.7	31.6	4.2
		297.2090	297.2097	7.01	1.83	
28		5.5	3.8	51.9	31.7	4.0
		86.4278	86.4271	-7.49	1.95	
37		25.0	17.1	53.2	141.6	3.9
		296.9955	296.9955	0.19	0.01	
23	8	5.2	3.9	42.9	32.8 //	4.8
		316.5315	316.5316	1.25	0.32	
24		7.5	3.9	72.3 >>	36.3	2.6
		339.1447	339.1449	2.06	0.52	
35		9.0	5.0	69.3 >>	44.9	2.7
		14.4936	14.4929	-6.80	1.36	
24	23	7.5	4.7	60.5 >>	39.7	3.3
		361.9926	361.9924	-2.22	0.47	
25		103.9	73.2	50.4	604.3	4.1
		8.8340	8.8348	8.00	0.11	
26		123.3	123.3	0.0 <<	305749.1 @@	2480.7 ##
		360.4214	360.4214	-0.02	0.00	
27		6.1	4.7	40.9	36.5 //	5.0
		340.8031	340.8032	0.63	0.13	
36		29.0	20.6	49.6	170.0	4.2
		213.6950	213.6969	18.55	0.90	
25	23	7.5	3.3	80.4 >>	34.5	2.0

		233.7001	223.7000	-1.09	0.01		
27		6.1	3.3	71.0 >>	30.0	2.6	
		305.2196	305.2197	1.20	0.36		
28		5.5	3.4	62.9 >>	28.9	3.2	
		311.3000	311.2996	-3.96	1.17		
35		7.6	4.1	70.4 >>	37.3	2.7	
		281.3341	281.3343	2.30	0.56		
27	3	5.1	3.2	61.6 >>	26.9	3.3	
		321.9832	321.9840	8.34	2.63		
	8	5.2	3.2	62.1 >>	27.0	3.2	
		234.1179	234.1178	-1.38	0.43		
28		8.6	3.2	86.3 >>	38.0	1.6	
		57.8617	57.8609	-7.86	2.48		
35		9.0	4.5	74.8 >>	43.0	2.4	
		274.4977	274.4964	-13.00	2.87		
29	2	5.0	2.9	67.0 >>	25.1	2.9	
		230.0850	230.0848	-1.96	0.69		
	8	7.8	4.2	71.0 >>	38.1	2.6	
		334.7737	334.7732	-5.29	1.26		
15		5.9	3.6	63.4 >>	30.5	3.1	
		18.7495	18.7497	2.42	0.68		
30		6.3	3.4	70.1 >>	31.1	2.7	
		226.7860	226.7867	6.65	1.93		
31		6.3	3.4	70.3 >>	30.9	2.7	
		227.2311	227.2307	-3.51	1.03		
39		15.8	10.2	57.9	85.6	3.5	
		390.3271	390.3276	4.71	0.46		
30	5	4.9	3.7	43.3	30.9 //	4.7	
		243.1949	243.1945	-4.09	1.10		
29		6.3	4.2	55.1	35.1	3.7	
		33.6147	33.6153	5.54	1.31		
31		52.1	36.6	50.7	302.6	4.1	
		265.7485	265.7485	-0.22	0.01		
32		64.7	64.6	0.3 <<	4773.0 @@	73.6 ##	
		301.6361	301.6361	0.32	0.01		
33		9.2	5.8	61.1 >>	48.9	3.3	
		214.9915	214.9918	2.54	0.44		
31	5	4.9	2.5	74.6 >>	23.5	2.4	
		317.4079	317.4073	-6.16	2.48		
	6	4.9	2.5	74.4 >>	23.5	2.4	
		285.1762	285.1766	4.16	1.67		
12		4.9	2.5	74.2 >>	23.5	2.4	
		289.9164	289.9166	2.12	0.85		
29		6.3	3.5	69.1 >>	31.1	2.8	
		108.3002	108.3003	1.00	0.29		
30		52.1	36.6	50.7	302.4	4.1	
		139.9888	139.9894	6.40	0.17		
32		69.1	68.3	2.3 <<	1874.9 @@	26.8 ##	
		98.2578	98.2579	0.94	0.01		
33		9.4	5.1	70.2 >>	46.1	2.7	
		287.7682	287.7682	0.01	0.00		
38		10.4	6.8	57.1	56.9	3.6	
		367.9978	367.9974	-3.76	0.55		
33	8	5.5	4.0	46.5	33.2 //	4.4	
		185.4726	185.4727	0.66	0.17		
30		9.2	5.3	67.1 >>	46.6	2.9	
		197.0881	197.0885	3.61	0.68		
31		9.4	5.2	69.5 >>	46.3	2.7	
		195.6235	195.6239	4.46	0.86		
34		11.9	8.0	55.4	66.1	3.7	
		377.0240	377.0244	3.75	0.47		
37		8.1	5.6	53.0	46.2	3.9	
		61.1888	61.1879	-9.41	1.68		
34	1	5.1	2.8	70.3 >>	24.9	2.7	

		133.8269	133.8273	3.56	1.31		
28		5.4	2.7	75.2 >>	25.9	2.4	
		134.0406	134.0403	-3.45	1.27		
33		11.9	8.1	54.2	66.8	3.8	
		296.7731	296.7730	-1.17	0.15		
35		7.0	3.7	71.8 >>	34.0	2.6	
		118.4585	118.4583	-2.01	0.54		
36		13.0	8.9	53.2	73.3	3.9	
		175.6875	175.6883	8.54	0.94		
35	8	5.2	3.4	57.2	28.4	3.6	
		142.3223	142.3227	4.23	1.24		
23		9.0	4.4	76.0 >>	42.9	2.3	
		51.3692	51.3687	-5.18	1.17		
25		7.6	4.6	63.3 >>	39.3	3.1	
		130.7531	130.7526	-4.53	0.99		
27		9.0	5.1	67.8 >>	45.2	2.8	
		389.3118	389.3109	-8.89	1.74		
34		7.0	4.2	63.0 >>	36.2	3.2	
		152.5559	152.5564	4.62	1.09		
36	24	29.0	22.2	41.1	186.7 //	4.9	
		314.4504	314.4496	-7.94	0.36		
34		13.0	12.4	8.2 <<	186.7 @@	13.8 ##	
		96.5107	96.5109	1.58	0.13		
37	8	5.4	3.8	50.7	31.1	4.1	
		68.8481	68.8490	8.90	2.36		
20		6.4	3.8	64.4 >>	32.7	3.1	
		80.3254	80.3239	-14.87	3.93		
22		25.0	17.9	49.0	147.6	4.2	
		268.6284	268.6281	-3.35	0.19		
33		8.1	5.1	60.6 >>	43.2	3.3	
		126.8106	126.8110	4.31	0.84		
38	8	5.7	5.1	18.8 <<	54.3 @@	8.6 ##	
		152.6149	152.6152	3.06	0.60		
21		17.6	12.8	47.1	106.2 //	4.4	
		346.8233	346.8233	-0.47	0.04		
31		10.4	7.1	53.1	59.0	3.9	
		229.2523	229.2513	-10.00	1.41		
39	8	8.4	5.5	56.2	46.1	3.6	
		110.0163	110.0165	2.09	0.38		
15		6.2	4.5	48.6	36.9	4.2	
		212.1211	212.1211	0.04	0.01		
17		9.7	6.2	60.0	52.0	3.4	
		235.0452	235.0452	0.46	0.08		
29		15.8	11.0	51.6	90.7	4.0	
		379.4130	379.4134	4.49	0.41		
40		16.1	10.5	57.6	87.7	3.5	
		161.7194	161.7180	-13.95	1.33		
40	8	9.1	5.6	62.6 >>	47.7	3.2	
		276.5098	276.5116	18.25	3.26		
15		0.6	5.3	35.4	45.5 //	5.6	
		1.1485	1.1475	-10.21	1.54		
19		40.8	28.6	50.8	236.6	4.1	
		235.2804	235.2807	2.79	0.10		
39		16.1	11.6	48.3	95.8	4.3	
		143.2285	143.2290	4.59	0.40		

Mittel : 20.6 15.3 53.9 2914.5 @@ 27.3 ##

STRECKEN

Anteil vFv: 21.87
 mÖ der Gruppe Strecken nach der Ausgleichung: 1.14 mm

Station	Ziele	mÖ/Jp Messwert	mÖ*Jo ausgeglt	r (%) Verb.	Gwh w	Ef.
7	9	9.7	3.8	34.2	43.4	1.7
		57.779	57.770	-8.86	2.36	
		9.7	5.0	73.7	46.6	2.5
	10	85.334	85.327	-6.80	1.37	
		9.7	5.2	71.6	47.5	2.6
		89.617	89.617	-0.45	0.09	
9	10	9.7	3.4	87.7	42.7	1.5
		45.134	45.130	-4.45	1.31	
		9.7	4.6	77.2	45.5	2.2
	11	68.253	68.254	0.75	0.16	
		9.7	5.9	32.2	50.3	3.1
		22.345	22.346	0.71	0.12	
17	16	9.7	4.3	80.1	44.7	2.1
		14.710	14.709	-0.55	0.21	
		9.7	5.8	64.5	49.8	3.1
	39	740.328	740.343	15.04	2.61	
		9.7	7.0	47.8	57.8	4.3
		6.514	6.515	0.56	0.08	
24	26	9.7	5.0	73.2	46.7	2.5
		4.625	4.626	0.93	0.19	
		9.7	7.6	37.6	65.2 //	5.3
	36	83.651	83.650	-0.82	0.11	
		9.7	6.0	61.0	51.2	3.3
		2063.853	2063.877	23.98	3.97	
29	15	9.7	7.3	42.7	61.2 //	4.8
		1791.274	1791.296	21.59	2.95	
		9.7	7.0	48.3	57.5	4.3
	39	282.480	282.477	-2.84	0.41	
		9.7	7.1	45.9	59.0 //	4.5
		25.838	25.839	0.96	0.13	
30	32	9.7	4.8	75.1	46.1	2.4
		16.776	16.777	0.75	0.15	
		9.7	7.7	37.2	65.5 //	5.4
33	31	803.682	803.685	3.20	0.42	
		9.7	7.6	39.0	64.0 //	5.2
		495.455	495.449	-6.14	0.81	
	37	9.7	7.0	47.1	56.2 //	4.4
		1058.376	1058.385	9.10	1.29	
		9.7	7.0	48.2	57.6	4.3
34	35	1448.549	1448.552	3.36	0.48	
		9.7	7.6	38.0	64.8 //	5.3
		418.975	418.972	-3.29	0.43	
35	23	9.7	5.4	68.9	48.2	2.8
		858.078	858.088	9.94	1.84	
		9.7	5.1	71.7	47.2	2.6
	25	1224.117	1224.138	20.54	3.99	
		9.7	5.6	66.8	49.0	2.9
		865.808	865.810	2.24	0.40	
37	22	9.7	7.9	33.6	69.0 //	5.8
		112.345	112.356	11.27	1.43	
		9.7	7.6	38.5	64.4 //	5.2
38	21	225.562	225.576	13.86	1.83	
		9.7	6.8	50.2	56.4	4.1
		648.364	648.374	10.21	1.49	
40	19	9.7	6.0	61.7	50.9	3.3
		42.167	42.181	14.47	2.42	
		9.7	6.0	62.0	50.8	3.2
	39	270.551	270.538	-13.38	2.24	
Mittel		9.7	6.1	58.5	53.8	3.6

8.4 Gesamtanalyse Brand - Nagelberg

Dabei sollte die Analyseausgleichung auf ein Netz mit größerer Neupunktanzahl angewandt werden. Das Netz zeigt im Südteil zwei parallel verlaufende Polygonzüge. Im Mittelteil sind teilweise parallele bzw. übergreifende Polygonzüge beobachtet worden. Der Nordteil läßt eine gute Netzstruktur erkennen. Dieses aus der realen Vermessungspraxis stammende Netz, stellt ein für die Analyseausgleichung besonders interessantes Testoperat dar.

Tatsächlich zeigen sich die Probleme besonders anhand der Zuverlässigkeitskennzahlen. Die Darstellung der Fehlerellipsen zeigt Abbildung 8.12.

Nach einer Varianzkomponentenuntersuchung mußte das stochastische Modell abgeändert werden:

$$\begin{aligned}m_R &= \pm 8 \text{ } \infty \\m_S &= \pm (7 \text{ mm} + 2 \text{ ppm})\end{aligned}$$

Mit diesem angepaßten stochastischen Modell wurde eine erneute Berechnung durchgeführt. Die Darstellung der Helmert'schen Punktfehlerellipsen ist in Abbildung 8.14 angegeben.

Fehlerhafte Beobachtungen konnten weder am Gesamtnetz, noch nach Netzpartitionierung nachgewiesen werden. Die Datumsüberprüfung weist die Gruppe der 5 Festpunkte als verträgliche Datumspunkte aus.

1: Die durchschnittlichen Werte der Zuverlässigkeitskennzahlen liegen im Verhältnis zu den anderen Beispielen deutlich ungünstiger

In Abbildung 8.13 wird die Verteilung der lokalen Standardabweichung der Gewichtseinheit l_{m_0} in Form eines Histogramms wiedergegeben.

Die Fehlersituation einer erneuten Durchrechnung mit Berücksichtigung einer Maßstabsunbekannten (-13 ppm) ist in Abbildung 8.15 dargestellt.

BRAND
Originalauswertung

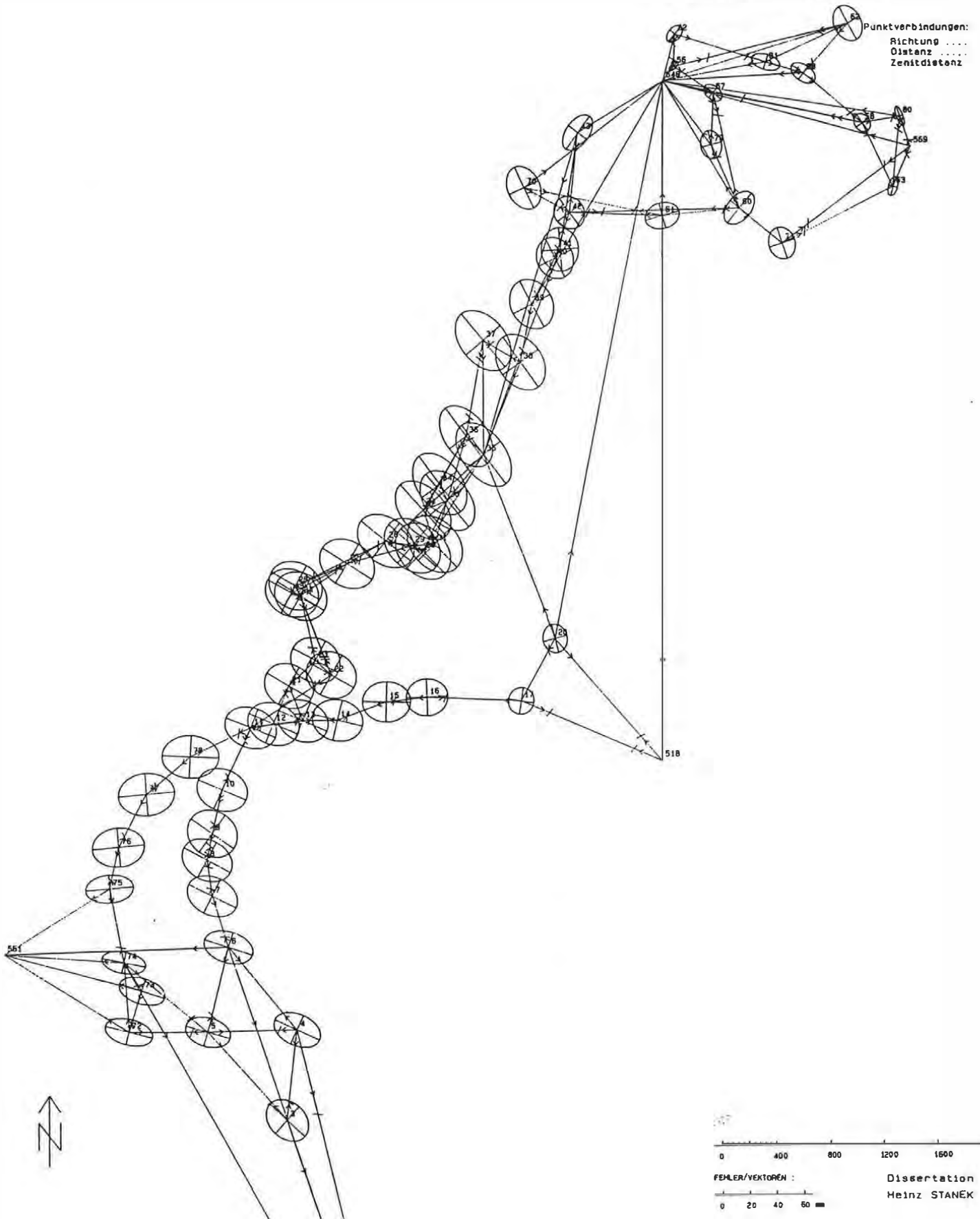


Abb. 8.12

PROJEKT : BRAND 88/09/01 17:57:44

Kommentar : EP Netz: BRAND - HÄGELBERG originale Gewichtung

Punkte : 65 (5 Festpunkte, 60 Neupunkte)
 Beobachtungen : 300 (200 Richtungen, 100 Strecken)
 Unbekannte : 178 (120 Koordinaten, 58 Orientierungen)
 Redundanz : 122
 Kondition : 3215
 Bedingungen : 0

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 3.0
 Festpunkt-Lagefehler ± 5.0 mm
 Richtungsfehler ± 3.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 5.0 mm
 Entfernungsabhängiger Streckenfehler ± 2.0 ppm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 7.0
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 8.2 cc
 Streckenfehler ± 5.2 mm

FESTPUNKTE

Nummer	Bezeichnung	Y	X
501	501	-98818.850	5404825.450
518	518	-96838.510	5409964.540
	Or = 325.9499		
546	546	-96795.100	5414999.670
551	551	-101744.070	5408572.350
569	569	-94951.720	5414500.650
	Or = 316.8309		

PROJEKT : BRANDA 88/09/02 12:49:55

Kommentar : EP Netz: Brand - HÄGELBERG aktualisierte Gewichtung 20-08-1988

Punkte : 65 (5 Festpunkte, 60 Neupunkte)
 Beobachtungen : 300 (200 Richtungen, 100 Strecken)
 Unbekannte : 178 (120 Koordinaten, 58 Orientierungen)
 Redundanz : 122
 Kondition : 1721
 Bedingungen : 0

STEUERDATEN

Dimension : Lage-Ausgleich
 Orientierungsunbekannte : auch in Festpunkten
 Maßstabsunbekannte : nein

MITTLERE MESSFEHLER a priori

Gewichtseinheitsfehler ± 1.0
 Festpunkt-Lagefehler ± 0.0 mm
 Richtungsfehler ± 8.0 cc
 Orientierungsfehler ± 50.0 cc
 Konstanter Streckenfehler ± 7.0 mm
 Entfernungsabhängiger Streckenfehler ± 2.0 ppm

VARIANZKOMPONENTEN a posteriori

Gewichtseinheitsfehler ± 1.2
 mit den gruppenweisen Anteilen:
 Richtungsfehler ± 1.1 cc
 Streckenfehler ± 1.2 mm

FESTPUNKTE

Nummer	Bezeichnung	Y	X
501	501	-98818.850	5404825.450
518	518	-96838.510	5409964.540
	Or = 325.9497		
546	546	-96795.100	5414999.670
551	551	-101744.070	5408572.350
569	569	-94951.720	5414500.650
	Or = 316.8309		

ZUVERLÄSSIGKEIT DER BEOBSACHTUNGEN

$m0/Jp$ bzw $m0*Jq$ = mittl. Fehler der Beobachtung vor bzw nach dem Ausgleich.
 $r = 1-p*q$ = Redundanzanteil der Beobachtung an der Gesamtredundanz Er.
 GwA = Grenzwert für die Aufdeckung grober Beob.fehler bei $\alpha=0.1\%$ und $\beta=80\%$.
 $Ef.$ = Einflußfaktor (Maß für die äußere Zuverlässigkeit)
 Fehlerwerte und Verbesserungen für Richtungen in cc, für Strecken in mm.
 Der kritische Wert k für die normierten Verbesserungen ist 3.3.

RICHTUNGEN

Anteil vtfv: 93.86
 $m0$ der Gruppe Richtungen nach der Ausgleichung +/- 1.11 cc

Station	Ziele	$m0/Jp$ Messwert	$m0*Jq$ ausgeg1	r (%) Verb.	GwA w	Ef.
(Mittel)						
		9.5	7.4	38.2	70.2 //	6.0



STRECKEN

Anteil vtfv: 76.75
 $m0$ der Gruppe Strecken nach der Ausgleichung +/- 1.30 mm

Station	Ziele	$m0/Jp$ Messwert	$m0*Jq$ ausgeg1	r (%) Verb.	GwA w	Ef.
(Mittel)						
		8.4	6.1	45.7	59.4 //	5.3

Verteilung der $lm0$ BRANDA

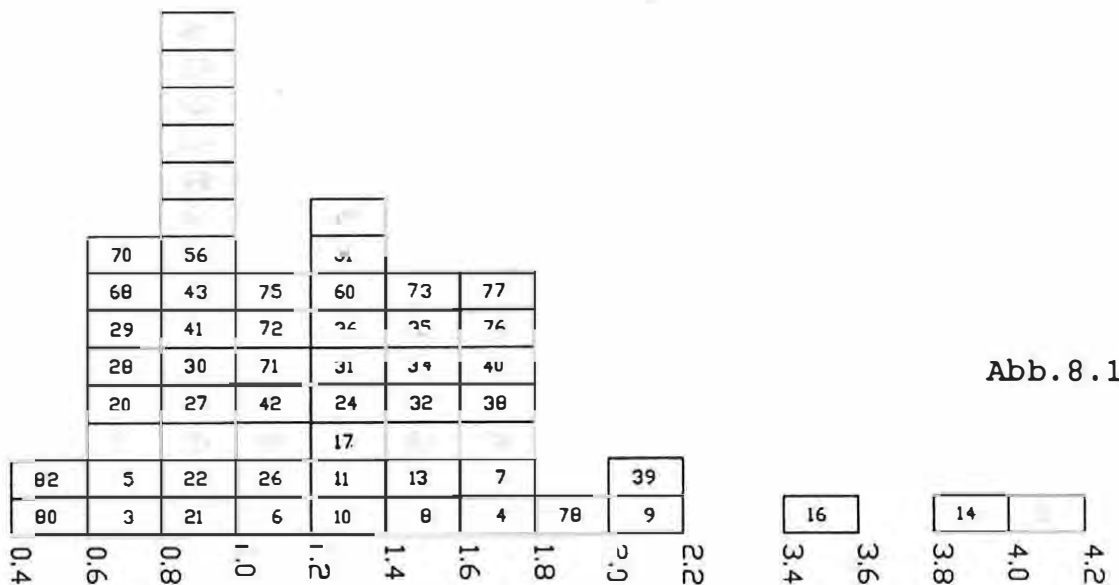


Abb. 8.13

BRANDA
Angepasste Gewichte

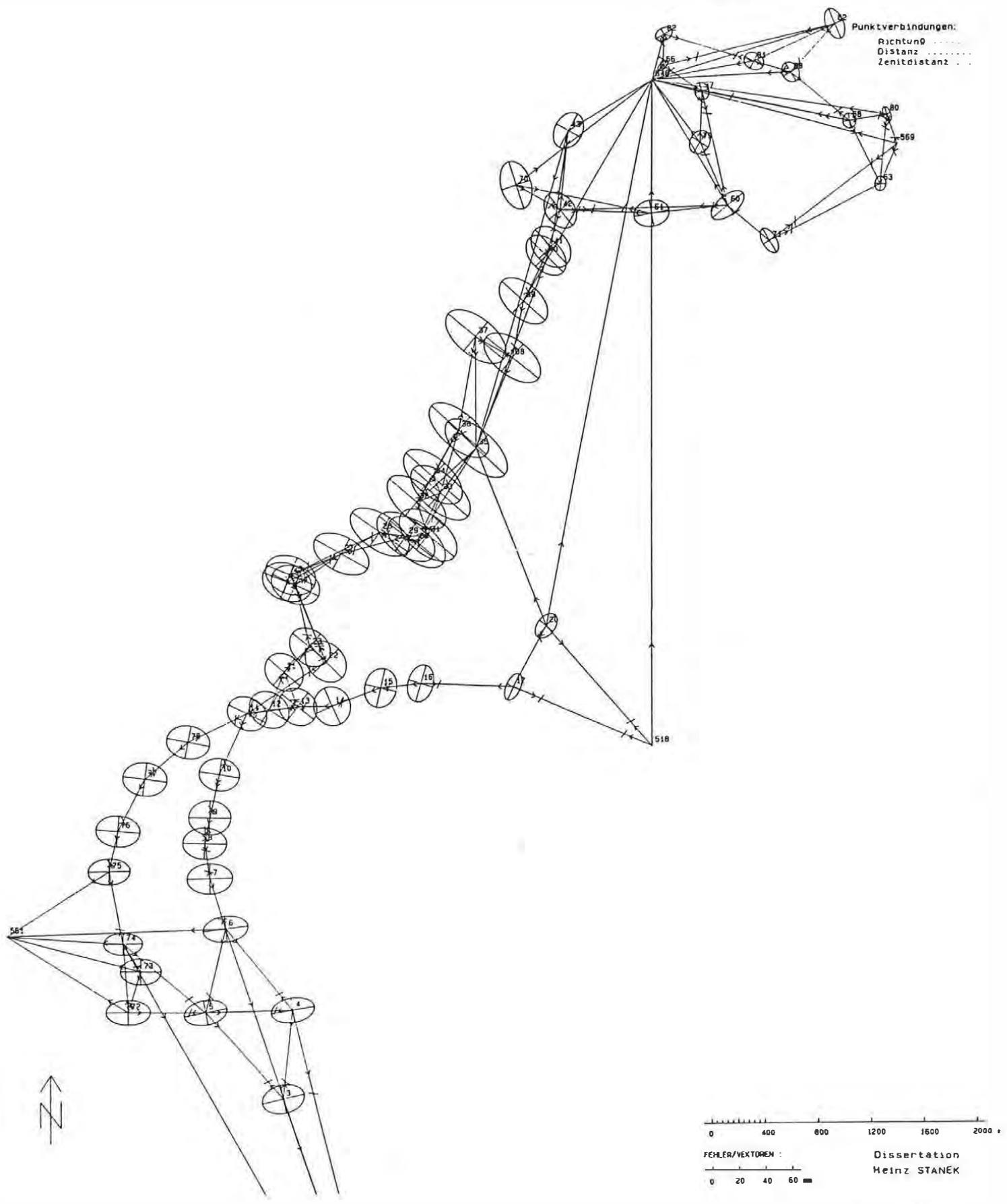


Abb. 8.14

BRANDAK
Angepasste Gewichte

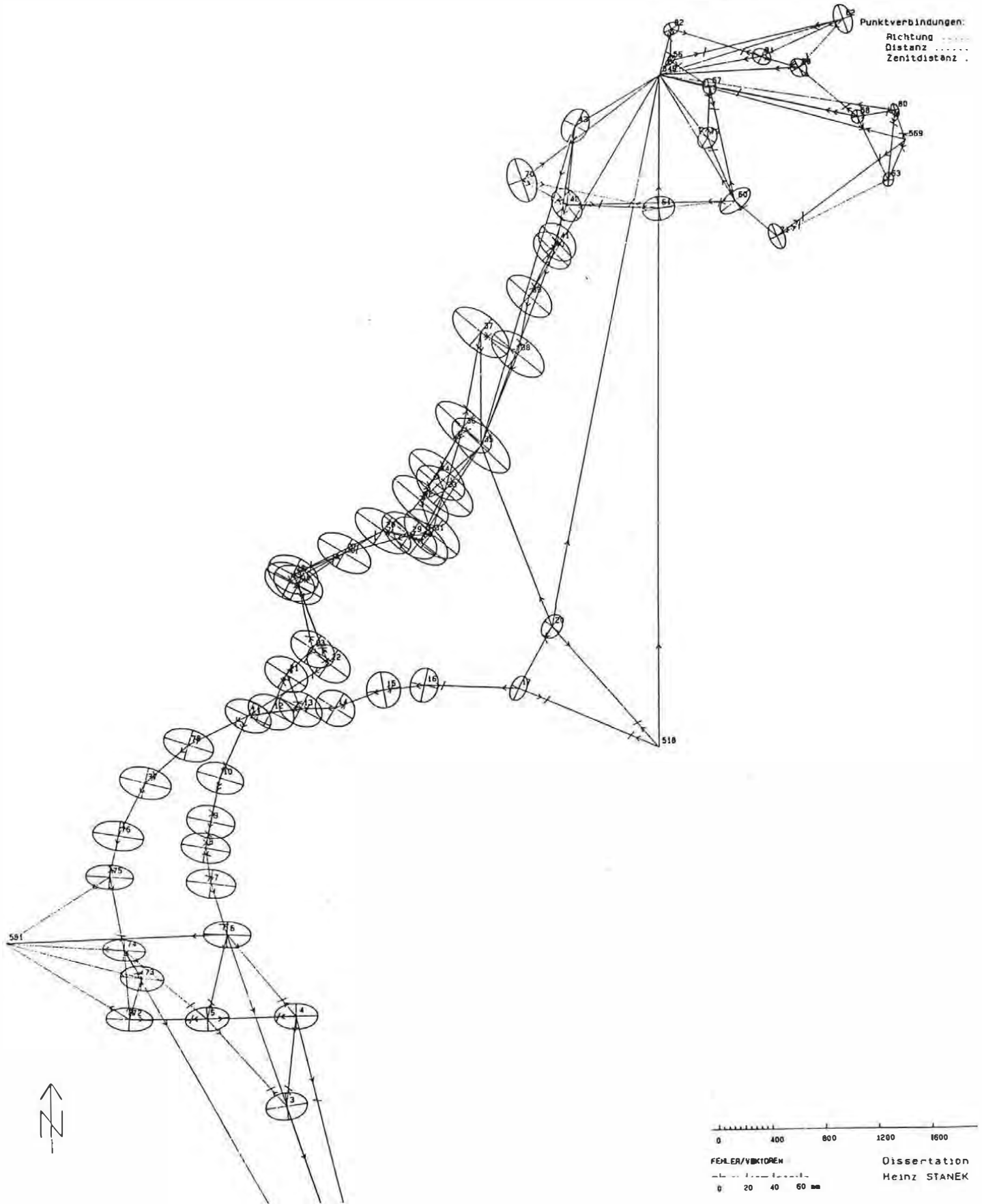


Abb. 8.15

9. Ausblicke, Weiterentwicklungen

Das vorgestellte Programmsystem ANAG stellt eine Möglichkeit zur interaktiven Bearbeitung und Analyse von Lagenetzen dar. Über die Möglichkeiten üblicher Ausgleichungsprogramme hinaus stellen dabei die Informationen bezüglich Zuverlässigkeiten wesentliche Informationen zur Interpretation geodätischer Lagenetze bereit. Eine große Anzahl von durchgeführten Berechnungen fand seinen Niederschlag in einer Reihe von Modifikationen und Anpassungen während der Testphase der Programmentwicklung. Diese beinhalteten zum einen die Vereinfachung der Kommunikation zwischen Bearbeiter und Programm und zum anderen die Erweiterungen des Funktionsspektrums. Folgende, angeführte Modifikationspunkte, betreffen Erweiterungen des Programms, die zum einen Teil vorbereitet wurden und zum anderen Teil in Zukunft erfolgen sollen.

- Einbeziehung räumlicher Netze:

Verschiedene räumliche Aufgabenstellungen sollten ebenfalls hinsichtlich deren Zuverlässigkeiten untersucht werden. Die Erweiterung ist im wesentlichen abgeschlossen. An Beobachtungsformen treten dann Zenitdistanzen und gemessene Höhenunterschiede zu den angegebenen hinzu. Die zu untersuchenden Anwendungen betreffen in erster Linie Vorausabschätzungen. Leider liegen derzeit keine Realnetze mit ausreichendem Beobachtungsmaterial vor. Eine Schwachstelle dieser Netzauffassung liegt in der Tatsache begründet, daß korrelierte Beobachtungsformen derzeit nicht behandelt werden können. Dies stellt aber ebenfalls einen weiteren Punkt der möglichen Erweiterungen dar.

- Modifikation des Speicherschemas für Matrizen:

Wie erwähnt, wurde das vorliegende Programm ANAG in erster Linie auf hohe Bearbeitungsgeschwindigkeit ausgelegt. Eine kompakte Speicherform konnte daher bisher nicht umgesetzt werden. Um größere Unbekanntmengen verarbeiten zu können (derzeitige Größe maximal 220) werden spezielle Speichertechniken (Sparsetechnik) zur Anwendung gebracht werden. Diese Modifikation ist derzeit in einer ersten Testphase. Wahrscheinlich wird sie für das PC - Programm nicht realisiert werden.

Es ist jedoch geplant, das Programm auch auf einem Großrechnersystem

zu implementieren und dann das Speicherschema entsprechend zu überarbeiten.

- Einbeziehung korrelierter Beobachtungen

Im speziellen werden korrelierte Koordinatendifferenzen (GPS-Beobachtungen) in naher Zukunft verbreitet zur Verfügung stehen. Die dabei auftretenden Korrelationen können nicht vernachlässigt werden. Als Ursache für diese Korrelationen sind zum einen die in allen drei Koordinatenrichtungen wirksamen Bahndaten und zum anderen die Transformation der im Satellitensystem beobachteten Koordinatendifferenzen in das jeweilige Berechnungssystem zu nennen. Sinnvollerweise werden dabei lediglich Korrelationen innerhalb von Beobachtungsgruppen Berücksichtigung finden. Diese Modifikation stellt die einschneidendste Veränderung innerhalb des Programmes dar. Zu diesem Zweck müssen die Datenstrukturen überarbeitet werden. Weiters stellt sich bei der Einbeziehung solcher Beobachtungen die Frage, ob Veränderungen am mathematischen Modell - etwa durch Einbeziehung von Transformationsparametern in die Ausgleichung - vorgenommen werden müssen.

Literaturverzeichnis

- ALBERDA J.E. 1980
A Review of Analysis Techniques for Engineering
Survey Control Schemes
Proc. Industrial and Engineering Survey Conferenc
London 1980
- AUGATH W. 1985
Lagenetze
In: PELZER (Hrsg.) Geodätische Netze in Landes- und
Ingenieurvermessung II
Stuttgart 1985 Wittwer Verlag
- BAARDA W. 1968
A Testing Procedure for Use in Geodetic Networks
Neth. Geod. Comm. Publ. in Geodesy
New Series 2 No. 5 Delft 1968
- BAARDA W. 1973
S Transformation and Criterion Matrices
Neth. Geod. Comm. Publ. in Geodesy
New Series 5 No. 1 Delft 1973
- BÄHR H.G. 1973
Zum Ausgleichungsverfahren bei der Ergänzung
bestehender geodätischer Netze
DGK Reihe C Heft Nr. 187 München 1973
- BÄHR H.G. 1983
Reihenfolge für Messungen und Punkte eines
geodätischen Netzes nach Gesichtspunkten der
Genauigkeit und Zuverlässigkeit
AVN 107/1983
- BENNING 1988
KAFKA Komplexe Analyse flächenhafter Kataster
Aufnahmen
- BILL R. 1984
Eine Strategie zur Ausgleichung und Analyse von
Verdichtungsnetzen
DGK Reihe C Heft Nr. 295 München 1984
- BILL R. 1985
Die Leistungsfähigkeit moderner Programme zur
Ausgleichung ebener geodätischer Netze
Zfv 11/1985
- BJERHAMMER A. 1958
A generalized matrix algebra
Kungl. Tekn. Högsk. Handl. Stockholm 124
- CAROSIO A. 1983
Verfahren der multivariaten Statistik zur
Beurteilung der Resultate und der Zuverlässigkeit
geodätischer Meßsysteme
Inst. f. Geodäsie und Photogrammetrie
ETH Zürich 1983 Mitteilungen Nr. 35
- DÖLLER H. 1988
Strategien zur Behebung von Verzerrungen in langen,
schlauchartigen Netzen am Beispiel des EP Netzes
Brand / Nagelberg
Diplomarbeit T.U. Wien 1988
- DUPRAZ H. NIEMEIER W. 1978
Beurteilungskriterien geodätischer Netze
Beitrag zum II. Internationalen Symposium über
Deformationsmessungen mit geodätischen Methoden
Bonn 1978

- EBENHOCH W. 1986
Genauigkeits- und Zuverlässigkeitsabschätzungen
durch Apriori - Ausgleich für zwei- und
dreidimensionale Netze
Diplomarbeit T.U. Wien 1986
- EHLERT D. 1984
Die Diagnoseausgleichung 1980 des deutschen
Hauptdreiecksnetzes
DGK Reihe B Heft Nr. 267 Frankfurt am Main 1984
- ELMIGER A. 1972
ALGOL Programm NETZ - 3D vermittelnde Netzaus-
gleichung in Lage und Höhe und dreidimensional
Inst. f. Geodäsie und Photogrammetrie
ETH Zürich Computerprogramme Nr. 1 1972
- ENGELN - MÜLLGES G. REUTTER F. 1987
Formelsammlung zur Numerischen Mathematik mit
TURBO - PASCAL - Programmen 2. Auflage
B.I. Wissenschaftsverlag Mannheim Wien Zürich 1987
- ERKER E. 1981
Charakteristische Daten und Testberechnungen im
österreichischen Anteil an ED 79
Österr. Z.f.V.u.Ph. Jg. 69 / 1981 Heft 1
- FORSTHUBER J. SCHNEID J. STANEK H. 1987
Partitionierte Orthogonalisierung und Singulärwert-
analyse zur Lösung der Ausgleichung nach
vermittelnden Beobachtungen
Österr. Z.f.V.u.Ph. 75 Jg. / 1987 Heft 1
- FÖRSTNER W. 1978
Die Suche nach groben Fehlern in photogrammetrischen
Lageblöcken
DGK Reihe C Heft Nr. 240 München 1978
- FÖRSTNER W. 1979
Das Programm TRINA zur Ausgleichung und Güte-
beurteilung geodätischer Lagenetze
zfv 1979/2
- FRITZENSMEIER K. NIEMEIER W. KASPELHERR R. HELLRIEGL F.
TENGEN D. 1988
Aufbau und Leistungsfähigkeit eines hybriden Über-
wachungsnetzes aus terrestrischen und GPS -
Beobachtungen aufgezeigt am Pumpspeicherwerk VIANDEN
In: SCHNADLBACH K. EBNER H. (Hrsg.)
Ingenieurvermessung 88
Bonn 1988 Dümmler (7809)
- GOLD W. 1987
Grafische Darstellung von Punktlage, Beobachtungs-
plan und Genauigkeitssituation geodätischer Netze
unveröffentl. PASCAL Programm Abt. LV T.U. Wien
- GRAFAREND E.W. 1972
Genauigkeitsmaße geodätischer Netze
DGK Reihe A Heft Nr. 73 München 1972
- GRAFAREND E.W. SANZO F. 1985
Optimization and Design of Geodetic Networks
Springer Berlin Heidelberg New York Tokyo 1985
- GRIMM - PITZINGER A. HANKE K. 1985
Parameterschätzung in Lagenetzen mit dem
Programmsystem NETCAD - 2D
Inst. f. Geodäsie Institutsmitteilungen Heft 10
Uni Innsbruck 1985
- HANKE K. GRIMM - PITZINGER A.
Eine Optimierungsstrategie für Tunnelnetze
Österr. Z.f.V.u.Ph. 75. Jg. / 1987 Heft 2

- HECK B. 1980
 Statistische Ausreißerkriterien zur Kontrolle
 geodätischer Beobachtungen
 VIII. Internationaler Kurs für Ingenieurvermessung
 Zürich 1980
- HECK B. 1981
 Der Einfluß einzelner Beobachtungen auf das Ergebnis
 einer Ausgleichung und die Suche nach Ausreißern
 in den Beobachtungen
 AVN 88 / 1981 Heft 1
- HECKMANN B. 1985
 Diagnoseausgleichung des trigonometrischen Netzes
 II. Ordnung und des Zwischennetzes I. Ordnung
 in Hessen
 DVW Hessen Mitteilungen Heft 2/1985
- HÖGGERL N. 1986
 Die Ausgleichung des österreichischen
 Präzisionsnivelllementnetzes
 Osterr. Z.f.V.u.Ph. 74. Jg. / 1986 Heft 4
- HÖPCKE W. 1980
 Fehlerlehre und Ausgleichsrechnung
 W. de Gruyter Berlin, New York 1980
- ILLNER M. 1986
 Anlage und Optimierung von Verdichtungsnetzen
 DGK Reihe C Heft Nr. 317 München 1986
- ILLNER M. 1987
 Ausgleichungs- und Optimierungsmodelle der
 Netzverdichtung
 AVN 3/1987
- JENSEN K. WIRTH N. 1975
 PASCAL user manual and report 2. Auflage
 Springer New York 1975
- KOCH K.R. 1987
 Parameterschätzung und Hypothesentests in linearen
 Modellen
 Bonn 1987 Dümmler (7892)
- KRAUS K. 1984
 Photogrammetrie
 Band 2: Theorie und Praxis der Auswertesysteme
 Bonn 1984 Dümmler (7865)
- LAWSON Ch.L. HANSON R.J. 1974
 Solving least squares problems
 Prentice - Hall Inc. Englewood Cliffs New Jersey 1974
- LITSCHAUER J. 1973
 Die zweite Ausgleichung des österreichischen
 Dreiecksnetzes 1. Ordnung
 Bundesamt für Eich- und Vermessungswesen 1973
- LITSCHAUER J. 1979
 Das österreichische Dreiecksnetz 1. Ordnung in ED 77
 Osterr Z.f.V.u.Ph. Jg. 67 /1979 Heft 2
- MATTHIAS H.J. 1984
 Tunnelnetze: Grundlagen, Numerischen Untersuchungen
 zur Optimierung
 Inst. f. Geodäsie und Photogrammetrie
 ETH Zürich 1984 Mitteilungen Nr. 38
- MATTHIAS H.J. 1988
 Tunnelnetze 2
 In: SCHNADLBACH K. EBNER H. (Hrsg.)
 Ingenieurvermessung 88
 Bonn 1988 Dümmler (7809)

- MEISSL P. 1962
Die innere Genauigkeit eines Punkthaufens
Osterr. Z.f.V.u.Ph. 1962
- MEISSL P. 1969
Zusammenfassung und Ausbau der inneren Fehlertheorie
eines Punkthaufens
In: RINNER, KILIAN, MEISSL: Beiträge zur Theorie
geodätischer Netze im Raum
DGK Reihe A Heft Nr. 61 München 1969
- MÜRLE M. BILL R. 1984
Zuverlässigkeits- und Genauigkeitsuntersuchungen
ebener geodätischer Netze
AVN 2/1984
- NIEMEIER W. 1985
Netzqualität und Optimierung
In: PELZER (Hrsg.) Geodätische Netze in Landes- und
Ingenieurvermessung II
Stuttgart 1985 Wittwer Verlag
- NIEMEIER W. 1987
Zur Bestimmung von Datumsparemtern aus
Beobachtungen
ZfV. 112 1987
- PELZER H. 1980
Dynamische oder hierarchische Netze ?
In: Geodätische Netze in Landes- und
Ingenieurvermessung
Stuttgart 1980 Wittwer Verlag
- PELZER H. 1980b
Beurteilung der Genauigkeit und Zuverlässigkeit
geodätischer Netze
In: Geodätische Netze in Landes- und
Ingenieurvermessung
Stuttgart 1980 Wittwer Verlag
- PELZER H. 1985
Grundlagen der mathematischen Statistik und der
Ausgleichsrechnung
In: PELZER (Hrsg.) Geodätische Netze in Landes- und
Ingenieurvermessung II
Stuttgart 1985 Wittwer Verlag
- PETERS C. 1971
Moderne Tendenzen der Ausgleichsrechnung
In: Aktuelle Entwicklungen in Vermessungswesen und
Raumplanung. Fünfte Fachtagung für Vermessungswesen
- RAO C.R. and MITRA S.K. 1971
Generalized Inverse of Matrices and its Applications
New York, London, Sydney, Toronto
J. Wiley & Sons Inc 1971
- REISSMANN G. 1980
Die Ausgleichsrechnung 5. Auflage
VEB Verlag für Bauwesen Berlin 1980
- RINNER K. 1976
Über die zweckmäßige Anlage von Tunnelnetzen
Darmstadt 1976

- SCHMID H. 1986
Ausgleichsrechnung I
Vorlesungsmanuskript
TU Wien 1986
- SCHMID H. 1987
Ausgleichsrechnung II
Vorlesungsmanuskript
TU Wien 1987
- SCHMID H.H. SCHERRER R. 1980
Vom freien zum gelagerten Netz
Inst. f. Geodäsie und Photogrammetrie
ETH Zürich 1980 Mitteilungen Nr. 29
- SCHMITT G. 1985
Third Order Design
In: GRAFEREND E.W. SANSONI F. 1985 Optimization and
Design of Geodetic Networks
Springer Berlin Heidelberg New York Tokyo 1985
- SCHWARZ H.R. 1986
Numerische Mathematik
B.G. Teubner Stuttgart 1986
- STUBENVOLL K. MEISSL P. 1977 mit Beiträgen von J. ZEGER
Ein Computer - Programmsystem zur Verdichtung
trigonometrischer Netze
Programmbeschreibung EDV - Netz 77
- VAN MIERLO 1978
A testing procedure for analysing geodetic
measurements
In: proceedings of Deformation Measurement by
Geodetic Networks Bonn 1978
- WELSCH W. 1980
A posteriori Varianzschätzung im erweiterten Aus-
gleichungsmodell nach der Methode der kleinsten
Quadrate
In: Geodätische Netze in Landes- und
Ingenieurvermessung
Stuttgart 1980 Wittwer Verlag
- WELSCH W. 1981
Übersicht über die Schätzung von Varianzen und Ko-
varianzen geodätischer Beobachtungen für die
praktische Durchführung
In: CONZETT, MATTHIAS, SCHMID (Hrsg.) Ingenieur-
vermessung 80, Beiträge zum VIII. Internationalen
Kurs für Ingenieurvermessung Bonn 1981 Dümmler
- WIRTH N. 1984
Datenstrukturen und Algorithmen
Spektrum der Wissenschaft November 1984

- WOLF H. 1975
Ausgleichsrechnung Formeln zur praktischen Anwendung
Bonn 1975 Dümmler (7835)
- ZALKA T. 1988
Überarbeitung und Erweiterung des Ausgleichungs-
programms Netz - 3D und Verbindung der Ein- und
Ausgabe zu einer Datenbank
Diplomarbeit T.U. Wien 1988
- ZURMÜHL R. FALK S. 1984
Matrizen und ihre Anwendung 5. Auflage
Teil 1: Grundlagen
Springer Berlin Heidelberg New York Tokyo 1984
- ZURMÜHL R. FALK S. 1986
Matrizen und ihre Anwendung
Teil 2: Numerische Methoden
Springer Berlin Heidelberg New York Tokyo 1986

Anhang

Während der Testphase der Programmentwicklung zu dieser Arbeit konnte in Zusammenarbeit mit dem BA f.E u.Vw ein interessantes Projekt bearbeitet werden.

Für ein großes Tunnelprojekt in Westösterreich sollte die Frage untersucht werden, in welchem Ausmaß die Beobachtungen der bestehenden Triangulierungsnetze als Kontrolle für ein, mittels GPS - Satellitenbeobachtungen zu erstellendes Tunnelbasisnetz dienen kann. Als glücklicher Umstand ist dabei die große Anzahl von Streckenbeobachtungen zu erwähnen.

Im ersten Berechnungsschritt wurde das umfangreiche Beobachtungsmaterial, nach einer ersten Durchrechnung auf Grund von Genauigkeits- und Zuverlässigkeitskriterien im Hinblick auf die Zielvorstellung eines Tunnelbasisnetzes "ausgedünnt". Da die Beobachtungen bereits vorhanden sind konnten wirtschaftliche Überlegungen vernachlässigt werden.

In einigen Netzteilen, insbesondere den Portalbereichen wurden mittels einer Simulationsausgleichung zusätzliche Streckenbeobachtungen eingeplant.

Diese Streckenbeobachtungen wurden im September 1988 durchgeführt.

Die engültigen Auswertungen sind nach Modifizierung des Reduktionsmodells der Streckenbeobachtungen mit Einbeziehung von Korrekturen wegen Lotabweichungseinflüssen zur Zeit in Bearbeitung.

TRANSFORMATION VON GEODÄTISCHEN PROGRAMMEN

von

H. Töpfer

Zusammenfassung 'Transformation von geodätischen Programmen'

Die vorliegende Arbeit ist ein Nachschlagwerk für Geodäten über die physikalische und logische Übertragung von Programmen und Daten von einem Computersystem in ein anderes.

Es werden alte und neue Methoden sowie die Programmiersprache TRAFO vorgestellt, die es ermöglicht, weitgehend automatisch Programme von einer Programmiersprache in eine andere zu übertragen oder die Daten verschiedener elektronischer Registriereinheiten zu normieren.

Mit diesem Werkzeug ist der Geodät in der Lage, Transformationen von Programmen bzw. die Normierung von Vermessungsdaten selbst durchzuführen oder wenigstens die dabei auftretenden Probleme zu beurteilen.

1	Einführung	1
1.1	Situation	1
1.2	Problem	1
1.3	Ziel	3
1.4	Inhaltsübersicht	5
2	Grundlagen	8
2.1	Physikalische Übertragung	8
2.1.1	Einführung	8
2.1.2	Betriebsarten Senden/Empfänger	10
2.1.2.1	Simplex	10
2.1.2.2	Halb-Duplex	10
2.1.2.3	Voll-Duplex	10
2.1.3	Parallele/Serielle Über	11
2.1.3.1	Parallele Übertragung	11
2.1.3.2	Serielle Übertragung	11
2.1.3.2.1	Allgemein	11
2.1.3.2.2	Asynchrones Format	12
2.1.3.2.3	Synchrones Format	12
2.1.4	Protokolle	13
2.1.4.1	Allgemein	13
2.1.4.2	Softwareprotokoll	14
2.1.4.3	Hardwareprotokoll	14
2.1.5	Schnittstellen	15
2.1.5.1	Centronics	15
2.1.5.2	V.24	15
2.2	Bits	17
2.2.1	Logische Definition	17
2.2.2	Spezielle Bit-Kombinationen	17
2.2.3	Transformation in ein andere	17
2.2.4	Anzahl der Bits pro Zeichen	18
2.2.5	Bits in Datentypen	18
2.2.5.1	Einführung	18
2.2.5.2	Ganze positive Zahlen	20
2.2.5.3	Ganze Zahlen	22
2.2.5.4	Gleitkommazahlen	24
2.2.6	Bits bei Übertraqu	33
2.3	Zeichen	34
2.3.1	Bitmuster	34
2.3.2	Sonderzeichen	34
2.4	Zeilen	35
2.4.1	Logische Definition	35
2.4.2	Zeilen bei Umstellungen	35
2.5	Dateien	36
2.5.1	Logische Definition	36
2.5.2	Dateien bei Transformationen	36
2.5.3	Erweiterungen (Extensions) der	36
2.5.4	Programme mit mehreren Dateien	36
3	Bitorientierte Transformationen	38
3.1	Bits ersetzen	38
3.1.1	Konvertierung	38
3.1.2	Prozedur NEWB	39
3.1.2.1	Einführung	39
3.1.2.2	Struktogramm von NEWBIT	39
3.1.2.3	Source-Code von NEWBIT	39
3.1.3	Kompliziertere Formen	40
3.2	Genereller Austausch von Bits	41
3.2.1	Konvertierung	41
3.2.2	Prozedur CHANGE_BITS	41
3.2.2.1	Einführung	41

3.2.2.2	Struktogramm von CHANGE_BITS	41
3.2.2.3	Source-Code von CHANGE_BITS	42
4	Zeichenorientierte Transformation	43
4.1	Zeichen ersetzen	43
4.1.1	Konvertierung	43
4.1.2	Prozedur CHANGE	44
4.1.2.1	Einführung	44
4.1.2.2	Struktogramm von CHANGE	44
4.1.2.3	Source-Code von CHANGE	45
4.1.3	Grenzfälle	46
4.2	Genereller Austausch	48
4.2.1	Tabelle CHANGE_TAB	48
4.2.2	Tabelle CHANGE_FILE	49
4.2.3	Tabellenprogrammier	49
4.2.4	Prozedur CHANGE_ALL	51
4.2.4.1	Einführung	51
4.2.4.2	Struktogramm von CHANGE_ALL	51
4.2.4.3	Source-Code von CHANGE_ALL	52
4.2.5	Text-Veränderungen	53
4.2.6	Hierarchie in Tabellen	53
4.2.7	Rekursion in Tabellen	54
4.2.8	Weitere Möglichkeiten	56
5	Maskenorientierte Transfo	58
5.1	Einführung	58
5.2	Konstante und Variable	59
6	Strukturorientierte Transformation	61
6.1	Einführung	61
6.2	Lexikalische Analyse	63
6.3	Strukturelle Analyse	65
6.4	Programmbaum ändern	66
6.5	Ergebnis ausgeben	66
7	Existierende Werkzeuge	67
7.1	BNF	67
7.2	Attributierte Grammat	69
7.3	LEX	70
7.4	YACC	73
7.5	SOURCE-TO-SOURCE TRANSLATION	75
7.6	INVERSE TRANSLATION	75
7.7	A TWO-WAY LANGUAGE INTERFACE	76
7.8	CIRCULAR ATTRIBUTE GRAMMARS	76
7.9	VAX SCAN	76
7.10	Zusammenfassung	77
8	TRANS - Kurzbeschreibung	78
9	TRAFO - Kurzbeschreibung	80
9.1	Einführung	80
9.2	Anforderungen/Design von TRAFO	84
10	Beispiele	96
10.1	HP85-BASIC->PASCAL	96
10.2	WILD-GRE4->AGA-GEODAT	107
11	Schlußwort	111
12	Literatur	112
12.1	Allgemein	112
12.2	Manuals	114
Register	i/v

1 Einführung

1.1 Situation

Der Geodät steht heute häufig vor dem Problem, einen Computer (Hardware) als eines der Werkzeuge der Auswertung zu besitzen, den er gegen ein neueres, größeres, schnelleres und hoffentlich auch besseres Gerät austauschen muß.

Zusätzlich wird durch die stetige Entwicklung der elektronischen Tachymeter und der dazugehörigen Registriereinheiten eine Anpassung der physikalischen und logischen Schnittstellen für den Datenaustausch zwischen den Registriereinheiten und dem jeweiligen Computersystem notwendig.

Zum Beispiel wurden an der Technischen Universität Wien am Institut für Landesvermessung und Ingenieurgeodäsie, Abteilung Landesvermessung bis 1987 noch viele Programme (Software) für die dort vorhandenen Computer HP85 und HP86 entwickelt. Diese Programme dienen verschiedenen geodätischen Berechnungen, vor allem aber der Erfassung und Auswertung der meteorologischen Verhältnisse entlang und in der Umgebung des Zielstrahles von geodätischen Instrumenten.

Die Programme wurden im Laufe der Zeit immer weiter entwickelt und sind bis heute im Einsatz. Allerdings treten, bedingt durch das relativ hohe Alter der Computer, vermehrt Schwierigkeiten auf, diese in neue Systeme mit einem wesentlich höheren Standard zu integrieren. Ebenso steigt die Fehleranfälligkeit der Elektronik.

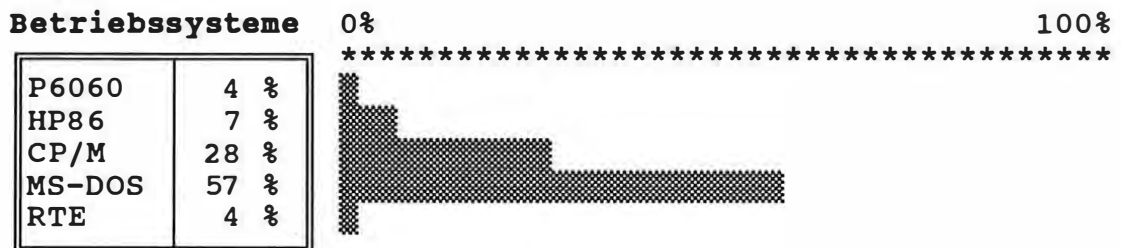
So müssen die Aufgaben auf neue Computer übertragen werden, wobei die in vielen Jahren entwickelten Programme möglichst automatisch transferiert werden sollen, um dann im neuen System ebenso wie im alten zu funktionieren - schließlich stellt ein vielfach verbessertes und gründlich erprobtes Programm eine Sammlung an Wissen dar, welches erhalten werden muß, um nicht immer wieder neu erzeugt werden zu müssen.

1.2 Problem

- ▶ Die physikalische Übertragung der Vermessungsprogramme (Software) kann besonders beim Übergang von den älteren Computern zu erheblichen technischen Problemen führen.
- ▶ Die Umstellung der Programme verursacht immer hohe Kosten, da die Systeme sehr verschieden sind und sich selten die Möglichkeit ergibt, mit dem neuen Computer auch neue Programme zu erwerben, die genau den Bedürfnissen des Geodäten entsprechen (z.B. dem Bedürfnis nach Berücksichtigung der gewohnten technischen und administrativen Büroorganisation, und dem nach einem niedrigen Preis ...).

- ▶ Zu erwähnen sind weiter die meist erheblichen Schwierigkeiten in der Büroorganisation während und nach einer Umstellung auf eine andere EDV-Anlage. Während Techniker das neue System installieren und neue Programme testen wollen, muß gleichzeitig unverändert die Produktion auf derselben Anlage weitergehen. Da in der Geodäsie besonders wegen der immer besseren Aufnahmegeräte die Software ständig angepaßt werden muß, ist dieser Konflikt zwischen Entwicklung und Produktion schon fast klassisch zu nennen. Ganz abgesehen davon werden Faktoren wie Lieferfristen, Aufstellungsort, Lärmentwicklung, Stromverbrauch und Klimatisierung im vorhinein immer wieder unterschätzt.
- ▶ Außerdem wird eine Umstellung meist mit einer Verbesserung der Vermessungsprogramme (benutzerfreundlichere Bedienoberfläche, Vermehrung der Funktionen, ...) verbunden, was eine Änderung des Ablaufes der Produktion mit sich bringen kann. Trotzdem muß das neue System in den alten Ablauf integrierbar sein - es sollte nur ein Austausch eines Werkzeuges durch ein neues mit mindestens den gleichen Funktionen erfolgen. Da dies in der Praxis nicht voll erreicht wird, ist eine Umstellung in der Büroorganisation nicht zu vermeiden.
- ▶ Eine weitere Schwierigkeit besteht darin, daß derzeit die meisten bei Ingenieurkonsulenten für Vermessungswesen in Verwendung stehenden Programme in "niedrigen" Programmiersprachen wie BASIC geschrieben sind.

Die Ingenieurkammer für Wien, Niederösterreich und das Burgenland (Fachgruppe Vermessungswesen) hat unter den Ingenieurkonsulenten für Vermessungswesen in ihrem Zuständigkeitsbereich von März bis Mai 1987 eine Umfrage durchgeführt, wobei unter anderem deren EDV-Ausstattung erfragt wurde. Der Rücklauf betrug 38% und hat folgende Ergebnisse gebracht:



Massenspeicher		0%	100%
<10M	39 %	*****	
10M-20M	39 %	*****	
>20M	22 %	*****	

43 % der Befragten gaben an, eine **andere Computer-Ausstattung** anzustreben !

Wie deutlich zu sehen ist, wird hauptsächlich BASIC als Programmiersprache angegeben. Nur relativ wenige verwenden die Programmiersprache FORTRAN, welche auch nicht typischerweise portabel ist. Solche Programme sind nicht strukturiert und daher besonders schwer umzustellen. Andererseits steckt in den Programmen sehr viel Entwicklungsarbeit. Der Umstieg auf moderne Computer, der von immerhin 43% angestrebt wird, sollte möglich sein, ohne die bewährten Programme vollständig "wegwerfen" zu müssen.

- ▶ Die Umschulung der Anwender auf das neue System verursacht außerdem direkte Kosten durch die Umschulung selbst und indirekte durch den gleichzeitig kaum zu vermeidenden Produktionsausfall.
- ▶ Die physikalische Datenübertragung der Registriereinheiten zum neuen Computer oder umgekehrt der Anschluß einer neuen Registriereinheit an einen bestehenden Computer muß mangels allgemeiner Lösung jedesmal neu verwirklicht werden, da dieses Problem stark vom jeweiligen Computer, dem Betriebssystem des Computers sowie besonders von der verwendeten Programmiersprache abhängt.
- ▶ Die logische Datenübertragung der Messungen von der Registriereinheit in die Vermessungsprogramme und in weiterer Folge die Normierung verschiedener Messungen zu einem gemeinsamen Format ist ebenfalls mit jedem Neuanschluß neu zu programmieren, da fast jede derzeit in Verwendung stehende Registriereinheit ein anderes Datenformat verwendet.

1.3 Ziel

Die vorliegende Arbeit soll all jenen Geodäten als Nachschlagwerk dienen, die sich entschlossen haben, auf andere Computer umzusteigen. Es werden Methoden vorgestellt, wie Umstellungen von Vermessungsprogrammen und Normierungen von Vermessungsdaten durchgeführt werden können.

Grundsätzlich werden 8 Elemente der EDV behandelt:

- 1) Programme : a) physikalische Übertragung
b) logische Umstellung
- 2) Daten : a) physikalische Übertragung
b) Normierung
- 3) Dokumentation der vorher genannten Punkte.

- ad 1a) Für die physikalische Übertragung der Vermessungsprogramme ist die Entwicklung eines generellen Konzeptes sowie konkreter Strategien notwendig. Dabei sollen als Beispiel detaillierte Vorschläge für einige der derzeit typischerweise bei Ingenieurkonsulenten für Vermessungswesen in Verwendung stehenden Computersysteme aufgezeigt werden.
- ad 1b) Die Umstellung Vermessungsprogramme soll durch ein eigenes Computerprogramm so weitgehend wie möglich automatisiert werden können. Dieses Programm soll als reales, allgemeines Werkzeug dienen - innerhalb der für die Geodäsie typischen EDV-Umgebung (siehe oben - Kammerumfrage): Dies sind Personalcomputer, das Betriebssystem MS-DOS und die Programmiersprachen BASIC, FORTRAN und zusätzlich PASCAL oder verwandte Programmiersprachen mit all ihren Dialekten. Ausdrücklich nicht behandelt werden Taschenrechner und auf der anderen Seite Hochsprachen wie LISP, PROLOG, MODULA u.s.w. Hier soll auch ein Beispiel einer geodätischen Anwendung der Erläuterung dienen; z.B. die Übertragung BASIC->PASCAL.
- ad 2a) Die physikalische Datenübertragung von den Registriereinheiten soll ebenfalls von einem möglichst allgemeinen anwendbaren Modul bewerkstelligt werden. Der Anschluß von Registriereinheiten soll durch ein typisches praktisches Beispiel demonstriert werden; z.B. Anschluß der Registriereinheiten GLE1, GRE4, GRE4, GEODAT an den Computer IBM-PC unter dem Betriebssystem MS-DOS mit der Programmiersprache TURBO-PASCAL 5.0.
- ad 2b) Für die logische Datenübertragung der Messungen bzw. die Normierung der Vermessungsdaten gilt es, ein allgemein anwendbares Konzept zu erstellen. Als erklärendes Beispiel sollte wieder folgende Anwendung dienen: Anschluß der Registriereinheiten GLE1, GRE4, GRE4, GEODAT an den Computer IBM-PC unter dem Betriebssystem MS-DOS mit der Programmiersprache TURBO-PASCAL 5.0.
- ad 3) Für all diese Aufgaben gilt: Die wichtigste Funktion ist die Dokumentation. Die Basis einer Umstellung sind die physikalischen Übertragungen. Es muß eindeutig feststellbar sein, ob die Programme bzw. Daten vollständig und richtig transferiert wurden. Ferner müssen die Programme zur Umstellung der Vermessungsprogramme bzw. zur Normierung der Daten die Möglichkeit beinhalten, den logischen Inhalt und die durchgeführte Änderung ausführlich und übersichtlich darzustellen (Cross-Compiler).

Es sei nochmals betont, daß diese Arbeit dem Geodäten ein Werkzeug im praktischen, kommerziell orientierten Einsatz sein soll. Es werden zwar auch neue, aber nur praktisch erprobte Verfahren vorgestellt.

1.4 Inhaltsübersicht

Im folgenden wird eine Übersicht über die wesentlichen Punkte dieser Arbeit gegeben:

Kapitel 2

'Grundlagen':

Dieser Abschnitt ist relativ ausführlich gestaltet, um nicht nur einen Überblick über die grundsätzlichen Elemente und die möglichen Verfahren der Transformation von Programmen zu geben, sondern um auch im Anschluß an die jeweiligen Definitionen konkrete Hinweise auf die Möglichkeiten und Schwierigkeiten zu geben, die in der Praxis auftreten. Erst die Kombination von Theorie und Praxis macht diese Arbeit zum Nachschlagwerk.

Folgende 4 Begriffe werden in diesem Kapitel definiert und ausführlich behandelt; jeweils beziehend auf physikalische bzw. logische Übertragungen der Programme bzw. Daten:

- ▶ Bit = kleinste Informationseinheit
- ▶ Zeichen = Folge von Bits
- ▶ Zeile = Folge von Zeichen
- ▶ Datei = Folge von Zeilen

Kapitel 3

'bitorientierte

Transformationen':

In diesem Abschnitt wird die primitivste Form der Transformation von Programmen oder der Normierung von Meßdaten vorgestellt. Dabei werden einzelne Bits innerhalb von Zeichen systematisch verändert. Als Beispiel wird das Programm CHANGE_BITS entwickelt und weiter auf Möglichkeiten, Schwierigkeiten und Grenzen der Methode hingewiesen.

Kapitel 4

'zeichenorientierte

Transformationen':

Dies ist die erste qualitative Steigerung einer Umformung. Die zeichenorientierte Transformation wird definiert durch das Austauschen von Zeichenketten in den Zeilen der Dateien. Als konkrete Realisierung der Methode wird das Programm CHANGE_ALL erstellt und mit seiner Hilfe dann der mögliche Einsatzbereich in der Geodäsie abgegrenzt. Ebenso fehlen auch hier nicht zahlreiche Hinweise auf Schwierigkeiten in der Praxis.

Kapitel 5

**'maskenorientierte
Transformationen'**

Dieser Abschnitt behandelt die nächsthöhere Stufe der Transformation von Programmen und die Normierung von Meßdaten. Im Prinzip ist die maskenorientierte Transformation eine Erweiterung der zeichenorientierten Transformation. In den Definitionen der auszutauschenden Zeichenketten können nun auch Variable vorkommen. Die Realisierung der Methode wird im Kapitel 'TRANS' ausführlich beschrieben.

Kapitel 6

**'strukturorientierte
Transformationen'**

Dies ist die vorläufig letzte, qualitativ höchste Stufe einer Umformung. Somit stellt dieser Abschnitt den theoretischen Kern der Arbeit dar. Hier wird in der lexikalischen Analyse der Strom von Zeichen in einen Strom von Symbolen umgewandelt, nach vorgegebenen Strukturen (unvollständige Programm-Bäume) gesucht, diese gespeichert und nach definierten Regeln umgeformt, und schließlich das Ergebnis als Strom von Zeichen ausgegeben. Die Realisierung ist dann die neue Programmiersprache TRAFO, die im Kapitel 'TRAFO' entsprechend detailliert beschrieben wird.

Kapitel 7

**'existierende
Werkzeuge'**

: Hier wird zuerst die verbreitete Syntaxbeschreibungssprache BNF erläutert. Dann wird auf LEX/YACC, einem Werkzeug zur Entwicklung von Übersetzungsprogrammen für Computer mit dem Betriebssystem UNIX, näher eingegangen. Anschließend werden noch weitere Werkzeuge kurz vorgestellt.

Kapitel 8

'TRANS'

: Dieses Kapitel beschreibt kurz die Programmiersprache TRANS und stellt beispielhaft eine höchst einfache Möglichkeit der Realisierung einer maskenorientierten Transformation dar. Es handelt sich dabei um ein Programm in der Programmiersprache MS-BASIC für die Computer IBM-PC unter dem Betriebssystem MS-DOS, welches seit 2 Jahren erfolgreich im praktischen Einsatz steht.

Kapitel 9

'TRAFO'

: Dieser Abschnitt ist der Kern dieser Arbeit. Es ist eine Kurzbeschreibung meiner neuen Programmiersprache mit dem Namen TRAFO. Diese ist eine Realisierung eines Werkzeuges für die strukturorientierte Transformation von Programmen und für die Normierung von Meßdaten. TRAFO wurde bereits für die Computer IBM-PC unter dem Betriebssystem MS-DOS in der Programmiersprache TURBO-PASCAL 5.0 implementiert.

2 Grundlagen

2.1 Physikalische Übertragung

2.1.1 Einführung

Bei der Übertragung von Daten zwischen verschiedenen Computersystemen wird man ebenso wie beim Anschluß von peripheren Geräten (z.B. Drucker, Meßgeräte,...) an den Computer als Zentraleinheit zwangsläufig mit dem Begriff 'Schnittstelle' (engl. interface) konfrontiert. Mit einer Schnittstelle wird eine physikalische Verbindung zwischen einem Computer und einem peripheren Gerät oder einem anderen Computer hergestellt und damit die Basis für eine Kommunikation zwischen diesen Komponenten geschaffen. Die Steuerung der Kommunikation übernehmen im allgemeinen die betroffenen Computer ([11],[16],[20]).

Schnittstellen bestehen also zunächst aus Hardwarebindigliedern zwischen einem Computer und weiteren Einheiten. Neben dem Hardwareaufbau ist außerdem die Steuerung der Übertragung von Nachrichten zwischen den einzelnen Geräten notwendig, die durch die Software geregelt wird. Nur die Kombination der Hardware- und Softwarekomponenten einer Schnittstelle ermöglicht einen regulären Betrieb.



Schnittstelle als Bindeglied
zwischen zwei Computersystemen

Solchermaßen definierte Schnittstellen erlauben eine Kombination zwischen verschiedenen Computersystemen, wobei die Computer im Regelfall mehrere Schnittstellen enthalten.

Dies gilt natürlich nur dann, wenn die beiden Systeme sozusagen dieselbe Sprache sprechen. Es gibt zwar verschiedene Normen für diese Datenübertragung, aber es tritt in der Praxis doch nur ausnahmsweise der Fall ein, daß man zwei Computer miteinander ohne weiteres verbinden kann. Abgesehen davon ist immer wieder ein gravierender Mangel an vollständigen und richtigen Beschreibungen der entsprechenden Schnittstellen zu bemerken; Jedenfalls empfehle ich ein gesundes Mißtrauen gegenüber den Angaben der Hersteller über ihre Schnittstellen.

Weiters werden in der Datenfernübertragung die an einem Datenaustausch beteiligten Komponenten als Datenendeinrichtung (engl. terminal equipment: DTE) und als Datenübertragungseinrichtung (engl. data communication equipment: DCE) bezeichnet. Außerdem werden Begriffe wie Datenquelle (z.B. ein Sender) und Datensenke (z.B. ein Empfänger) innerhalb eines Übertragungssystems benutzt. Als ein DÜE-Gerät wird beispielsweise ein Modem bezeichnet, das Datenübertragungen über das Telephonnetz ermöglicht, indem es digitale Daten in analoge Signale umwandelt und umgekehrt (Modulator/Demodulator). Als DEE-Gerät kann zum Beispiel ein Computer bezeichnet werden.

In den folgenden Kapiteln werden zwar auch allgemeine Aspekte von Datenübertragungen beleuchtet, die konkreten Aussagen seien aber in ihrem Gültigkeitsbereich auf den in der Geodäsie am weitesten verbreiteten Computer, nämlich auf den IBM-PC und die dazu (weitgehend) kompatiblen PC, beschränkt.

2.1.2 Betriebsarten Senden/Empfangen

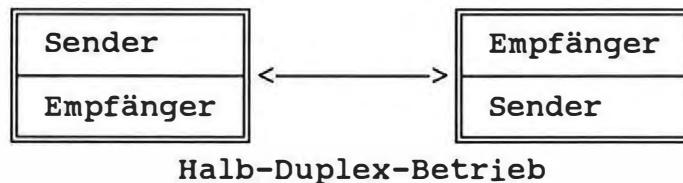
2.1.2.1 Simplex

In dieser Betriebsart werden Daten nur in einer Richtung übertragen und es erfolgt kein gegenseitiger Datenaustausch:



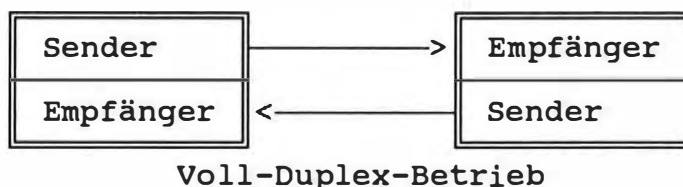
2.1.2.2 Halb-Duplex

In diesem Modus können beide Computer miteinander Daten austauschen. Beide Geräte können damit die Funktion des Senders und die Funktion des Empfängers übernehmen - allerdings nicht gleichzeitig. Sendet also ein Computer, so muß der andere Computer auf Empfangen geschaltet sein und umgekehrt. Somit findet ein sogenannter 'Wechselverkehr' durch Umschalten statt.



2.1.2.3 Voll-Duplex

In diesem Modus können, wie im Halb-Duplex-Betrieb, Daten zwischen zwei Computern ausgetauscht werden. Beide Teilnehmer können jedoch gleichzeitig die Funktion des Senders und des Empfängers übernehmen. Zur Realisierung sind aber im Unterschied zu den beiden anderen Modi zwei Datenleitungen notwendig. Dieser Nachteil wird allerdings durch den Vorteil der weitaus schnelleren wechselseitigen Kommunikation bei weitem aufgewogen.

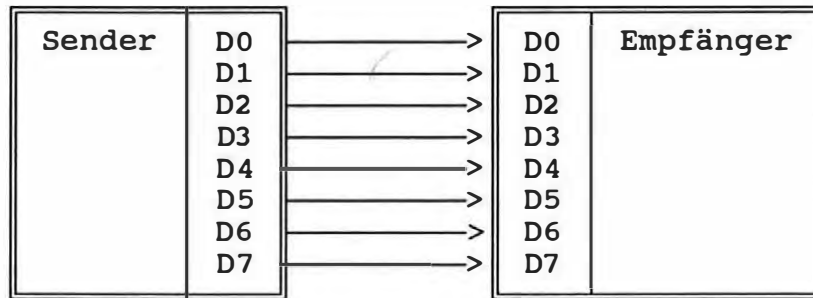


Hier sei allerdings noch eine kritische Anmerkung angebracht: Ein wechselseitiger Datenaustausch kann natürlich in der Praxis nur sehr beschränkt wirklich gleichzeitig durchgeführt werden, denn schließlich ist in einem Computer nur EIN Betriebssystem vorhanden, welches im Prinzip Schritt für Schritt abgearbeitet wird und deshalb niemals gleichzeitig zu sendende Daten vorbereiten und empfangene Daten weiterverarbeiten kann, selbst wenn das jeweilige Interface dazu in der Lage wäre.

2.1.3 Parallele/Serielle Übertragung

2.1.3.1 Parallele Übertragung

Die parallele Datenübertragung ist eine der beiden Möglichkeiten, Informationen zu senden und zu empfangen. Dabei werden (beim IBM-PC) gleichzeitig 7 oder 8 Datenbits übertragen, d.h. alle Bits eines Zeichens (Bytes) werden gleichzeitig - parallel - übertragen. Genauer gesagt handelt es sich dabei um eine bitparallele Übertragung von Daten.

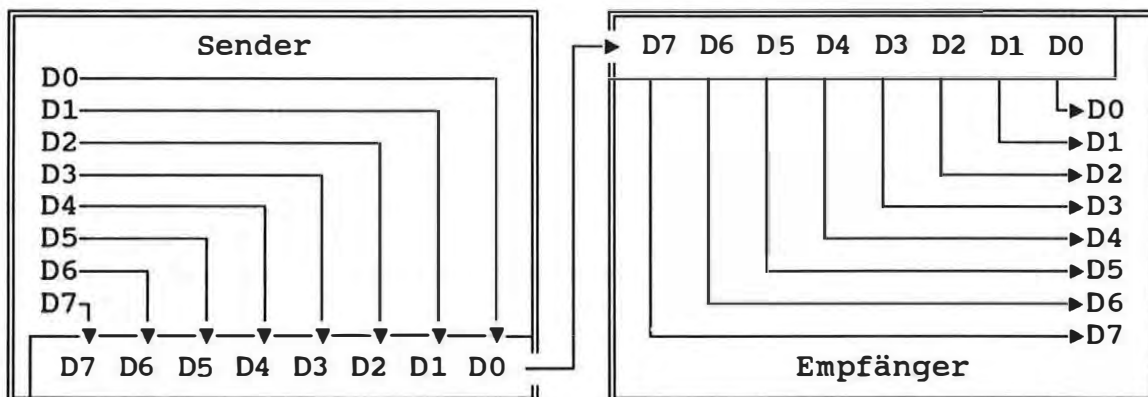


Der Vorteil der parallelen Übertragung ist jedenfalls die höhere Geschwindigkeit, der nicht zu unterschätzende Nachteil aber liegt in der Tatsache, daß damit nur sehr kurze Entfernungen überbrückt werden können (beim IBM-PC ca. 2-3m). Dieser Modus wird z.B. bei der Schnittstelle zwischen IBM-PC und Drucker angewendet.

2.1.3.2 Serielle Übertragung

2.1.3.2.1 Allgemein

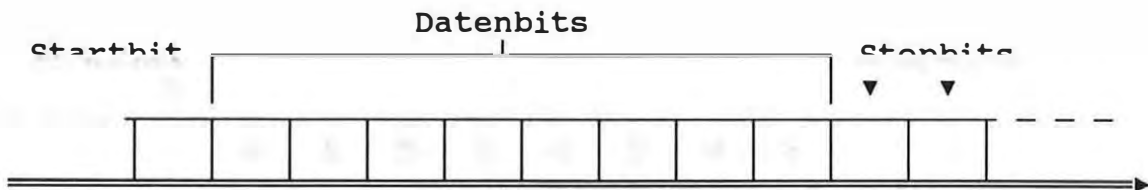
Dies ist die zweite Möglichkeit: Die einzelnen Bits jedes Zeichens (Bytes) werden vom Sender in einen Puffer gestellt und von dort hintereinander - seriell - in einen Puffer des Empfängers übertragen und anschließend zu einem vollständigen Byte zusammengesetzt.



Die serielle Übertragung ist zwar langsamer als die parallele Übertragung, es werden aber wesentlich weniger Leitungen benötigt und außerdem können damit fast beliebig große Entfernungen überbrückt werden.

2.1.3.2.2 Asynchrones Format

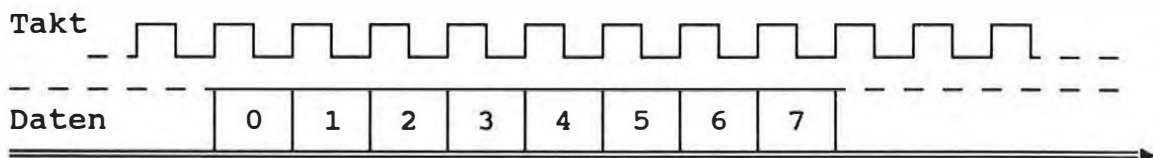
Das Datenformat gibt die inhaltliche Form an, in der ein Zeichen (Byte) gesendet wird. Wurde dieses Synchronisationsverfahren gewählt, wird vor jedem Datenbyte zunächst ein Startbit gesendet, im Anschluß an die Datenbits folgen dann 1 oder 2 Stopbits. Diese Stopbits sorgen dafür, das der Empfänger ausreichend Zeit hat, um sich vor dem Empfangen des nächsten Zeichens wieder auf den Sender einstellen zu können. Diese Synchronisierung (Start/Stop-Verfahren) wird also vor jedem Zeichen wiederholt. Daher benötigt man auch kein gesondertes Takt-Signal. Die Folge Startbit-Datenbits-Stopbits wird nebenbei bemerkt Zeichenrahmen genannt.



Diese Methode wird vor allem dann angewandt, wenn der Empfänger das Taktsignal leicht herstellen kann (z.B. mit einem Quarz) oder wenn gar nicht die Möglichkeit besteht, das Taktsignal über eine eigene Leitung zu übertragen (z.B. bei einer Verbindung über eine Telefonleitung). Die serielle asynchrone Datenübertragung ist auch die am häufigsten verwendete Methode der Datenübertragung überhaupt.

2.1.3.2.3 Synchrones Format

In diesem Format werden mehrere Zeichen (Bytes) zu einem Block zusammengefaßt und nach dem Senden eines Synchronisationsbytes unmittelbar hintereinander übertragen. Danach wird ein Ende-Byte gesendet. Während die Bytes übertragen werden, muß der Zeitrahmen für jedes Bit genau eingehalten werden. Um eine Abstimmung zu erreichen, wird eben vorab ein Synchronisationsbyte gesendet. Bei manchen Systemen wird außerdem noch eine zusätzliche Leitung verwendet, die das Takt-Signal des jeweiligen Senders überträgt.



Diese Form der Datenübertragung benötigt eine eigene Leitung für das Takt-Signal, andererseits erübrigt sich die Synchronisation über die Datenleitung und somit die Verwendung von Start- bzw. Stopp-Bits, wodurch die Daten in einer kompakteren Form übertragen werden können, was wiederum die Geschwindigkeit erhöht.

2.1.4 Protokolle

2.1.4.1 Allgemein

Neben der Übertragung der einzelnen Zeichen (Bytes) werden weitere Informationen benötigt, die über den Status der Datenübertragung zwischen Sender und Empfänger Auskunft geben. Dies läßt sich durch ein Übertragungsprotokoll regeln. Dies ist ein Werkzeug, um beispielsweise den Start und das Ende einer Übertragung, die Übertragungsrichtung oder auch Fehler mitzuteilen. Zur Führung eines solchen Protokolles gibt es mehrere Möglichkeiten, die mehr oder weniger verschiedenen Normen entsprechen.

Ein Element von Übertragungsprotokollen sind die Steuerzeichen. Die Übertragungssteuerzeichen regeln den Ablauf der Datenübertragung zwischen verschiedenen Computern; dazu gehört z.B. ACK (acknowledge), womit der Empfänger dem Sender mitteilt, daß er einen Datenblock korrekt empfangen hat; oder EOT (end of transmission), welches das Ende der Übertragung signalisiert.

Die Formatsteuerzeichen bestimmen die Form der Ausgabe der übertragenen Daten; z.B. LF (line feed) bewirkt einen Zeilenvorschub, CR (carriage return) einen Wagenrücklauf.

Die Gerätesteuerzeichen, wie DC1, DC2, DC3 und DC4, können verschiedene Geräteteile ein- bzw. ausschalten. DC1 und DC3 werden auch häufig für das XON/XOFF-Protokoll (siehe später) verwendet.

Im Gegensatz dazu können die Codeerweiterungszeichen SO (shift out) und SI (shift in) zwischen verschiedenen Zeichensätzen umschalten (wie das Drücken auf die SHIFT-Taste). ESC (escape) wird auch oft für Steuerzeichen, zum Beispiel für das Ein- bzw. Ausschalten von Fettdruck bei Druckern verwendet.

2.1.4.2 Softwareprotokoll

Das Softwareprotokoll einer Übertragung (software handshake) ist eine der Möglichkeiten, eine Übertragung so zu organisieren, daß die Daten sicher ankommen. Hier werden noch aktive und passive Verfahren unterschieden.

Ein passives Verfahren besteht zum Beispiel darin, daß der Empfänger durch Senden eines Zeichens (XOFF) dem Sender mitteilt, daß er die bereits empfangenen Daten verarbeiten muß, bevor er weitere Daten empfangen kann. Ist der Empfänger dann soweit, sendet er ein Zeichen (XON), welches dem Sender signalisiert, daß nun wieder gesendet werden kann. Dies nennt man dann XON/XOFF-Protokoll, wobei die Zeichen XON und XOFF ausgewählte Zeichen sind. Diese Methode ist in der Geodäsie am weitesten verbreitet.

Ein aktives Verfahren hingegen kann darin bestehen, daß der Sender vorerst einen Datenblock sendet und auf eine Bestätigung vom Empfänger wartet. Wenn der Empfänger den Block verarbeitet hat, sendet er ein spezielles Zeichen ACK (acknowledge), welches den korrekten Empfang quittiert und sozusagen die Genehmigung für weitere Zeichen erteilt. Wenn der Empfänger hingegen Inkonsistenzen in den empfangenen Daten feststellt, sendet er ein NAK (not acknowledge) zum Sender, worauf dieser den letzten Datenblock nochmals sendet u.s.w. Dieses Verfahren nennt man ACK/NAK-Protokoll.

2.1.4.3 Hardwareprotokoll

Eine weitere Möglichkeit besteht darin, durch ein Hardware-Protokoll (hardware handshake) die Übertragung zu organisieren.

Der einzige Unterschied zum Software-Protokoll besteht darin, daß nicht durch das Senden von speziellen Zeichen z.B. eine Bereitschaft zum Empfangen signalisiert wird, sondern daß dies mit Hilfe von separaten Leitungen geschieht. So setzt der Empfänger beispielsweise eine bestimmte Leitung auf eine positive Spannung, wenn er bereit ist, Daten zu empfangen und auf eine negative, wenn nicht. Der Sender muß in diesem Fall jedesmal die entsprechende Leitung überprüfen, bevor er ein Zeichen sendet.

Diese Methode ist deshalb nicht sehr beliebt, da zusätzliche (physikalische) Leitungen benötigt werden.

2.1.5 Schnittstellen

2.1.5.1 Centronics

Die Centronics-Schnittstelle ist von der Firma Centronics als Druckerschnittstelle für Personalcomputer entwickelt worden und hat sich für diese Anwendung weitgehend durchgesetzt. Es handelt sich dabei (angeblich) um einen sogenannten Industrie-Standard, der im Gegensatz zur V.24-Schnittstelle (siehe später) nicht genormt ist.

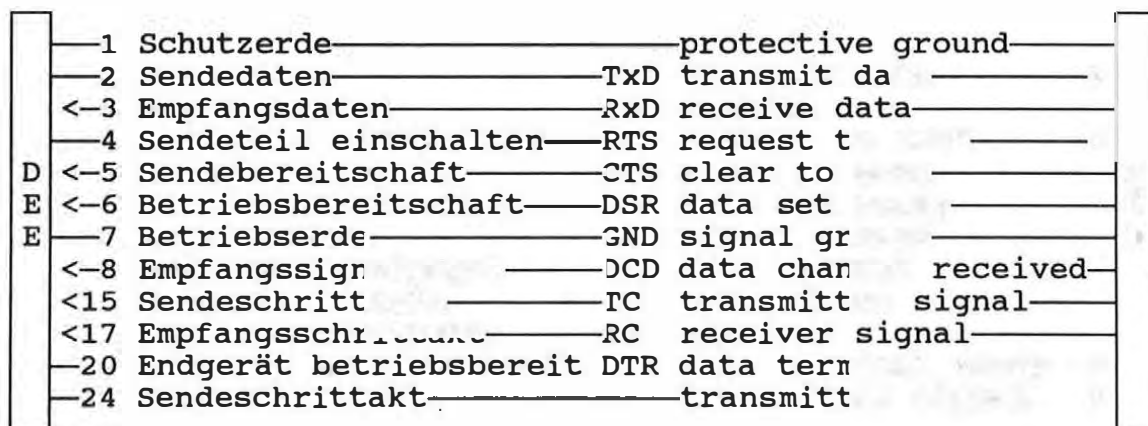
Diese Schnittstelle verwendet die parallele Datenübertragung und ist in vielen Computeresystemen bereits fest eingebaut.

Im Gegensatz zur seriellen Übertragung erweisen sich Konventionen, wie konkrete Belegung der einzelnen Leitungen eines Centronis-Steckers, als relativ einheitlich bei den verschiedenen Computerherstellern.

2.1.5.2 V.24

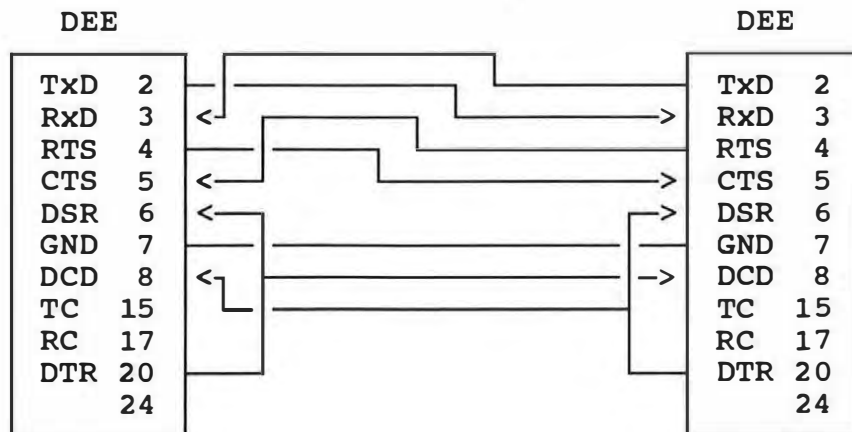
Für diese Schnittstelle sind leider verschiedene Bezeichnungen üblich. Die am häufigsten verwendeten Namen sind V.24 und RS232c. Dies sind Abkürzungen der französischen (CCITT V.24) bzw. der amerikanischen (RS 232 C) Norm. Die entsprechende deutsche Norm (DIN 66 020) wird ebenso wie die der internationalen Organisation für Standardisierungen (ISO-Norm 2110) kaum genannt. Die RS 232 C und die DIN 66 020 definieren die funktionalen elektrischen Eigenschaften, die V.24 aber nur die funktionalen Anforderungen.

In Gegensatz zur Centronics-Schnittstelle kann ein Computer über eine V.24-Schnittstelle Daten senden und empfangen. Außerdem können damit fast beliebig große Entfernungen überbrückt werden. Leider besteht in der Praxis eine Verwirrung über die möglichen Anwendungen der verschiedenen Leitungen, was zu immer abenteuerlicheren Steckerkonstruktionen führt.



Leitungsbelegung nach V.24

Eine typische Verbindung zwischen zwei IBM-PCs sieht daher folgendermaßen aus:



Aus den verschiedenen Kombinationsmöglichkeiten folgt, daß es sehr viele verschiedenen Verbindungsarten gibt. Um nun beim Zusammenschluß zwischen zwei Computersystemen nicht immer wieder neue Stecker löten zu müssen, habe ich die V24Box entwickelt. Dies ist eine elektronische Einheit, die im Prinzip dann über die RTS-Leitung serielle Signale empfangen kann, wenn die DTR-Leitung abgeschaltet (low) ist. Diese Signale werden dann als Befehle für das sogenannte Auskreuzen interpretiert.

Damit ist es möglich, während einer laufenden Übertragung zusätzliche Befehle an die V24-Box zu übermitteln, ohne daß der Empfänger davon gestört wird. Diese Befehle bewirken dann, daß die Leitungen elektronisch so umgelegt werden, daß die verschiedenartige Belegung der Buchsen an den Computersystemen und ihrer jeweilige unterschiedliche Funktion kompensiert werden kann. Dies geschieht ganz einfach durch eine digitale Ansteuerung von Relais. Dies benötigt zwar mehr Strom, ist aber gutmütiger gegenüber den leider üblichen sehr unterschiedlichen Spannungspegeln.

2.2 Bits

2.2.1 Logische Definition

Ein Bit ist die kleinste Informationseinheit - es kann immer nur zwei Werte haben: Ja oder Nein, Richtig (True) oder Falsch (False) und so weiter.

Ein Bit repräsentiert meist auch die physikalische, die elektronische Realität in einem Computer oder in einer Registriereinheit. Es entspricht dort verschiedenen elektrischen oder magnetischen Potentialen.

2.2.2 Spezielle Bit-Kombinationen

In der Praxis der Umstellung von Programmen oder auch bei der Normierung von Meßdaten bekommt die kleinste Informationseinheit oft mehr Bedeutung als vorab angenommen wurde.

Was verschiedene Code-Systeme betrifft, wird auf die folgenden Kapitel verwiesen. Man sollte die dabei auftretenden Schwierigkeiten nicht unterschätzen. Das erste Problem besteht nun darin, daß man zwar eine Zeichen-Tabelle erstellen kann, die jedes einzelne Zeichen einer Quell-Datei in das entsprechende des Ziel-Systems umwandelt, daß aber manche Zeichen innerhalb eines Computer-Systems besondere Bedeutung haben. Dabei spielen oft die jeweiligen Dateitypen eine wesentliche Rolle.

Als einfachstes Beispiel sei die Übertragung einer sequentiellen Text-Datei (beispielsweise ein Programm) unter dem Betriebssystem MS-DOS in einen Computer mit dem Betriebssystem UNIX genannt. Hier geben beide Systeme als Code-System die ASCII-Konvention an. Der Unterschied ist hier die verschiedene Handhabung der Zeichen, die ein Ende einer Zeile signalisieren. Diese sind nämlich verschieden: in MS-DOS ein CR (ASCII 13) und in UNIX ein LF (ASCII 10). Würde dem bei einer Transformation nicht Rechnung getragen, käme im UNIX-System ein unleserlicher Text heraus - ohne Zeilen-Struktur.

2.2.3 Transformation in ein anderes System

Als zweites Beispiel und als Demonstration des zweiten typischen Problem-Kreises sei eine Übertragung eines Programm-Textes eines Programmes unter dem Betriebssystem MS-DOS in einen sogenannten Abteilungsrechner des Typs HP3000 mit dem Betriebssystem MPE genannt. Das fragliche Zeichen ist hier das ^Z (ASCII 26). Dieses Zeichen kann auf einem Bildschirm eines PCs als Pfeilchen abgebildet werden; natürlich auch auf Druckern, die an eine HP3000 angeschlossen wären.

Das Problem besteht nun darin, daß man unter MS-DOS unmöglich eine Datei mit diesem Zeichen erstellen oder gar als Ergebnis einer Transformation Übertragung speichern kann, weil dieses Zeichen bei Text-Dateien unter MS-DOS das Ende der Datei signalisiert. Es geht somit darum, daß man Quelle und Ziel jeder solchen Transformation auf einem Gerät speichern muß, bevor die physikalische Übertragung stattfindet.

2.2.4 Anzahl der Bits pro Zeichen

Der dritte klassische Problemkreis zu diesem Thema ist schließlich die Anzahl der Bits pro Zeichen. Diese variiert leider oder ist oft durch die Art der physikalischen Übertragung der Vermessungsprogramme und besonders der Inhalte von Registriereinheiten verschieden.

Wenn die Anzahl der Bits im Quell-System kleiner ist als die im Ziel-System gibt es im allgemeinen weniger Probleme als beim umgekehrten Fall; hier muß man dann eine Art Generalisierung der Zeichen vornehmen. Dies geht so vor sich, daß man Zeichen für Zeichen überlegen muß, durch welches neue und einfachere Zeichen das alte ersetzt werden kann, ohne auf das Ergebnis einen schädlichen Einfluß zu haben.

2.2.5 Bits in Datentypen

2.2.5.1 Einführung

Zuletzt sei noch ein vierter Problemkreis angesprochen; dies sind gespeicherte binäre Daten, zum Beispiel Koordinaten. Leider variiert die Art der Speicherung nicht nur von System zu System, sondern auch von Programmiersprache zu Programmiersprache.

Es geht darum, wie die Verteilung von Bits bei den verschiedenen Datentypen jeweils gehandhabt wird. Das konkrete Beispiel sei die binäre Speicherung von reellen Zahlen. Hier gibt es besonders bei der Speicherung des Exponenten große Unterschiede zwischen den Systemen.

Dieses Problem tritt in der Geodäsie besonders häufig auf, da mit relativ großen Zahlenwerten gearbeitet wird und diese auch mit der vollen Genauigkeit gespeichert werden müssen. Wenn wir zum Beispiel die Landeskoordinaten betrachten, so ist eine typische X-Koordinate z.B. 5333123.123. Wenn wir diese mit [mm]-Genauigkeit speichern wollen, müssen mindestens 10 signifikante Dezimal-Stellen verwendet werden. So muß in diesem Beispiel ein Datentyp verwendet werden, der zur sicheren Speicherung und weiteren Verwendung den Berechnungen mindestens 11 signifikante Ziffern verwendet.

So hat man zwar im Quell-Programm einen bestimmten Datentyp vorliegen, es muß aber unabhängig davon ein geeigneter Datentyp für das Zielsystem ausgewählt werden; es darf keinesfalls der ursprüngliche Datentyp ohne Kontrolle übernommen werden. Dabei sind drei Aspekte unbedingt zu beachten:

- 1) Der neue Datentyp muß im neuen Programm die Koordinaten und andere Werte mit hohen Ziffernanzahlen ausreichend genau speichern können. Es ist nicht sinnvoll, 'sicherheitshalber' auf einen Datentyp höherer Genauigkeit überzugehen, weil dies einerseits oft mit einer Vervielfachung des Speicheraufwandes verbunden ist und andererseits diesbezügliche Angaben in Handbüchern nicht immer stimmen müssen. Ein einfacher Versuch lohnt sich meistens: Eine Zahl auf einen Datenträger speichern, im Programm nochmals initialisieren und gleich darauf wieder vom Datenträger lesen und das Ergebnis am Bildschirm ausgeben. Dieses kann dann visuell mit der ursprünglichen Zahl verglichen werden. Man kommt aber keinesfalls um die Arbeit herum, sich zu überlegen, welche Stellenanzahl bei der Speicherung wirklich benötigt wird.
- 2) Ebenso muß mit dem neuen Datentyp ausreichend genau gerechnet werden. Es muß auch hier genauestens überlegt werden, welche Genauigkeit beim Rechnen mit Landeskoordinaten notwendig ist. Nach der Wahl sollten dann ebenfalls experimentell die Genauigkeitsangaben der Handbücher überprüft werden. Es sei angemerkt, daß ja die Berechnung einer Zahl nicht nur auf der 'Oberfläche' eines Computers geschieht, sondern daß viele Komponenten zusammenspielen, deren schwächste schließlich die erreichbare Genauigkeit vorgibt. Im Prinzip wird eine Zahl sozusagen vom übersetzten Programm zum Laufzeitsystem der jeweiligen Sprache weitergereicht, von dort zum Betriebssystem und von diesem wiederum zur CPU (also zur Hardware). Erst dort wird die Berechnung tatsächlich durchgeführt, und das Ergebnis wird wieder bis zur Anwender-Oberfläche zurückgeschickt.
- 3) Der dritte Punkt ist die Überlegung, wie berechnete und gespeicherte Werte (meist Koordinaten mit ihren Beziehungen untereinander - zum Beispiel Messungen) von einem System in ein neues physikalisch und logisch übertragen werden können. Schon bei der Auswahl eines neuen Datentyps ist zu überlegen, ob die Wahl auch diese Möglichkeit beinhaltet und nicht von Beginn an die Übertragung sehr kompliziert oder gar unmöglich macht.

An dieser Stelle möchte ich am Beispiel des IBM-PC näher auf die Art eingehen, wie im allgemeinen Zahlen (und im besonderen reelle Zahlen, die zur Handhabung von Koordinaten und geodätischen Meßwerten geeignet sind), in Computern gehandhabt werden.

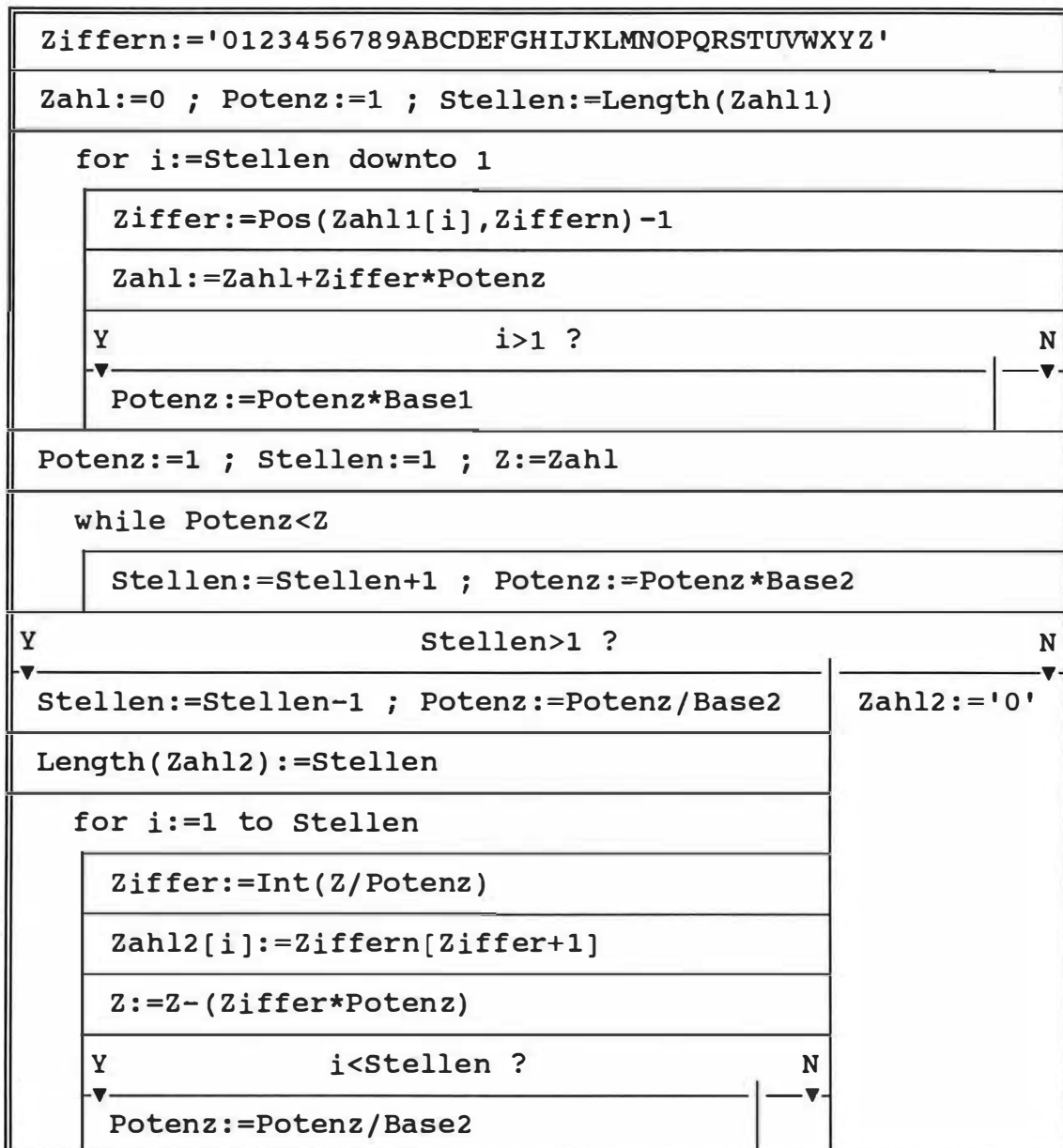
2.2.5.2 Ganze positive Zahlen

Vorläufig werden ganze Zahlen in der Binärrarithmetik betrachtet. Normalerweise rechnen wir in einem dezimalen Zahlensystem, d.h. wir rechnen mit den Ziffern 0 bis 9, haben daher 10 verschiedene Ziffern. Im Computer muß binär gerechnet werden, da elektronisch am einfachsten nur 2 verschiedene Zustände erzeugt werden können. Somit haben wir im binären System nur die Ziffern 0 und 1 zur Verfügung.

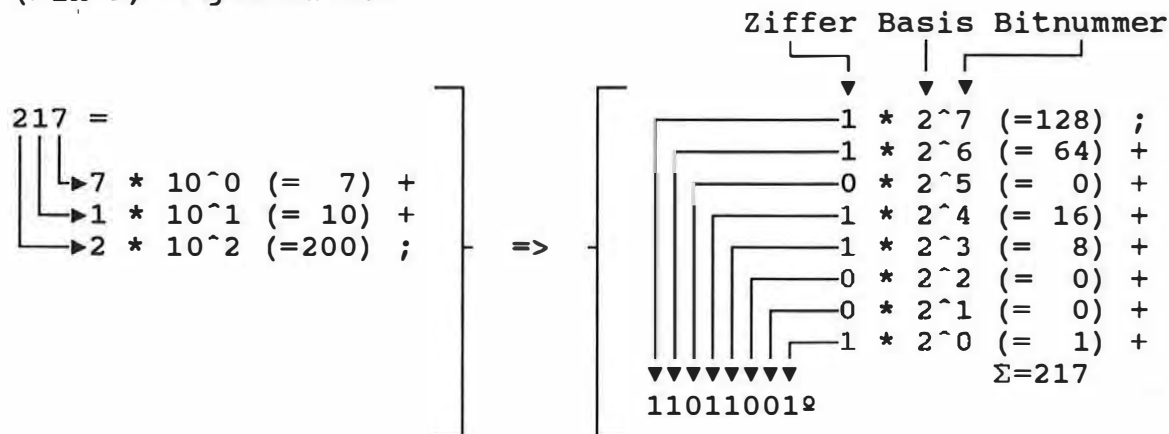
Die erste Aufgabe besteht nun darin, eine ganze Zahl im dezimalen System in das binäre System umzuwandeln. Zu diesem Zweck möchte ich einen einfachen allgemeinen Algorithmus vorstellen, der eine Zahl (ZAHL1) von einem beliebigen Zahlensystem (BASE1) in die entsprechende Darstellung (ZAHL2) in einem anderen beliebigen Zahlensystem (BASE2) umwandeln kann (für obige spezielle Aufgabe könnte der Algorithmus natürlich noch stark vereinfacht werden):

CONVERT

Zahl1(Base1) => Zahl2(Base2)



Unsere spezielle Anwendung obiger Prozedur sei nun die Umwandlung einer dezimalen Zahl in ihre binäre Darstellung. Im folgenden seien binäre Zahlen durch ein nachfolgendes '2' deutlich gekennzeichnet. Als Beispiel sei die Zahl 217 im System 10 (dezimal) in die entsprechende Zahl im System 2 (binär) umgewandelt:



Die einzelnen Ziffern werden mit dem Wert des Zahlensystems (10) - welcher um 1 vermindert mit der Stellennummer potenziert wird - multipliziert, und die Summe ergibt den dezimalen Wert. Durch den umgekehrten Vorgang, angewendet auf den Wert des neuen Zahlensystems, kommt man dann zur gesuchten Zahlendarstellung.

Nehmen wir nun unsere oben genannte X-Koordinate her und legen wir weiter fest, daß wir Werte zwischen 0 und 550000000 speichern wollen (Koordinaten der österreichischen Landesvermessung) - die Koordinaten also als ganze Zahlen in [mm] speichern. Der besseren Übersichtlichkeit halber seien im folgenden die langen binären Zahlen nach jeder vierten Ziffer durch ein Leerzeichen unterteilt.

$$5333123123 = 1\ 0011\ 1101\ 1110\ 0001\ 0000\ 0000\ 0011\ 0011_2$$

So sieht unsere Koordinate [mm] in ihrer binären Darstellung aus. Es werden hier also mindestens 33 Bits oder 5 Bytes (wenn ein Byte eine Folge von 8 Bits ist) benötigt, um eine typische Koordinate als ganze Zahl in [mm] im binären Format zu speichern. Sehen wir uns nun oben festgelegten Koordinatenbereich im binären Format an:

$$\begin{aligned} 0 &= & 0_2 \\ 550000000 &= 1\ 0100\ 0111\ 1101\ 0011\ 0101\ 0111\ 0000\ 0000_2 \end{aligned}$$

Somit können wir folgende Aussage treffen : Wenn wir annehmen, daß wir X-Koordinaten im Bereich von 0 bis 5500000m in [mm] speichern wollen, so benötigen wir einen Datentyp, der ganze Zahlen verwalten kann und mit mindestens 33 Bits oder 5 ganze Bytes arbeitet. 32-Bit-Variable sind in der Geodäsie daher nicht allgemein ausreichend !

Eine grobe Abschätzung, wieviele Bytes man für einen Datentyp für die Speicherung einer ganzen positiven Zahl benötigt, kann man noch auf eine andere Weise treffen:

x Anzahl Bits
 b Anzahl Bytes (1 Byte = 8 Bits)
 n Anzahl signifk. Stellen des pos. ganzz. Datentyps
 z Größte mögliche Zahl des pos. ganzz. Datentyps
 LOG Logarithmus zur Basis 10
²LOG ... Logarithmus zur Basis 2

$$2^x > z \Rightarrow x > {}^2\text{LOG}(z) \Rightarrow x > \text{LOG}(z)/\text{LOG}(2) \Rightarrow$$

$$x > \text{LOG}(10^n)/\text{LOG}(2) \Rightarrow x > n/\text{LOG}(2) \Rightarrow b*8 > n/\text{LOG}(2)$$

$$b > n/(8*\text{LOG}(2)) \Rightarrow \text{als Faustformel:}$$

$x > n*3.4$
$b > n/2.4$

Angewendet auf das vorige Beispiel der X-Koordinaten bedeutet dies $n=10$ signifikante Stellen und damit $b>4.16$, also aufgerundet mindestens 5 ganze Bytes für die Speicherung von X-Koordinaten als ganze positive Zahl in [mm].

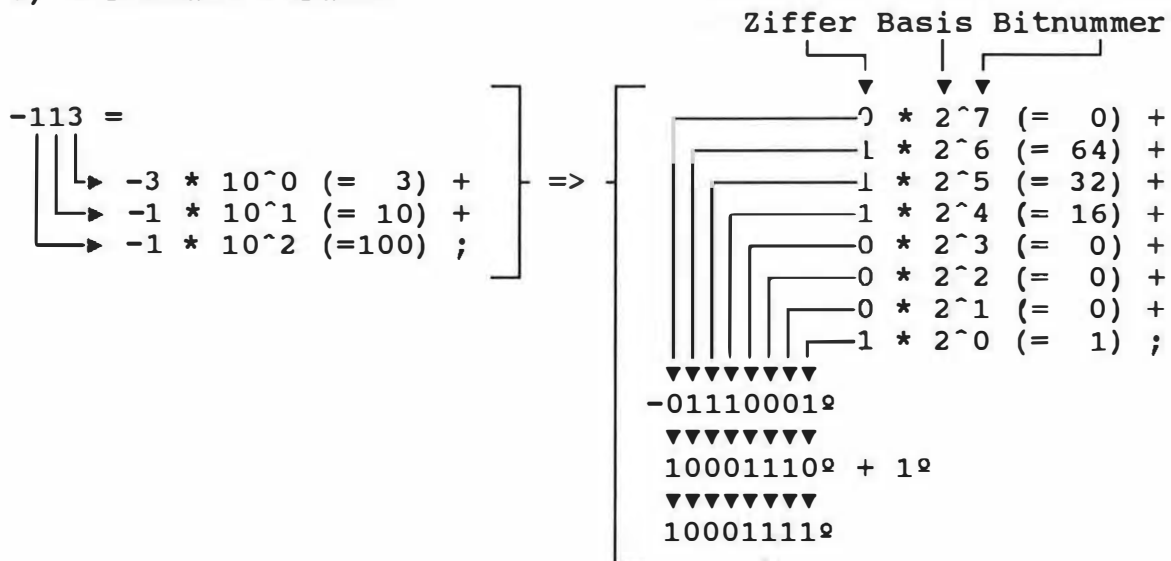
2.2.5.3 Ganze Zahlen

Wenden wir uns nun der zweiten Koordinate zu: der Y-Koordinate. Wir können auch diese in [mm] speichern, allerdings kann die Y-Koordinate im Gegensatz zur X-Koordinate auch negativ sein. Deshalb möchte ich an dieser Stelle kurz die Behandlung von Vorzeichen in der Binärarithmetik behandeln.

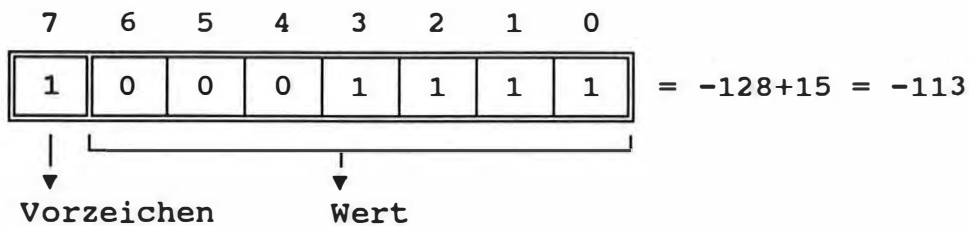
Ganze Zahlen, die negativ oder positiv sein können, werden ebenso vom dezimalen System in das binäre übertragen. Dabei zeigt das höchstwertige Bit (das am weitesten links stehende) meist das Vorzeichen an. Ist dieses Bit 0, so ist die Zahl positiv, ist es hingegen gleich 1, so ist die Zahl negativ. Dabei sehen wir schon, daß man bei Zahlen, die auch negativ werden können, vor der Konversion die gesamte Anzahl von verwendeten Bits unbedingt festlegen muß.

Positive Zahlen werden genau wie vorher beschrieben gespeichert. Negative Zahlen werden allerdings meist anders behandelt: Sie werden ebenfalls in das binäre System nach dem oben beschriebenen Verfahren umgewandelt, dann aber wird die Zahl binär komplementiert und anschließend um 1 erhöht (dies nennt man das Zweierkomplement). Dies bedeutet, daß vorerst alle Bits, die 0 sind, auf 1 gesetzt werden und umgekehrt und daß anschließend binär die Zahl 1 addiert wird. Damit ist auch immer das am weitesten links stehende Bit bei negativen Zahlen 1.

Um diese Ausführungen zu verdeutlichen, nehme ich als Beispiel die negative ganze Zahl -113 im dezimalen System und demonstriere die Umwandlung in ihre binäre Darstellung; Dabei sei angenommen, daß 8 Bits (Bit Nummer 0 bis Bit Nummer 7) verwendet werden:



Diese Zahl wird dann in folgender Form gespeichert:



Weiters möchte ich wie vorher eine Faustformel für den Speicherbedarf von ganzen Zahlen entwickeln, die positiv oder negativ sein können:

- x Anzahl Bits
- b Anzahl Bytes (1 Byte = 8 Bits)
- m Anzahl signifik. Stellen des ganzzahligen Datentyps
- I Größte mögliche Zahl des ganzzahligen Datentyps
- LOG Logarithmus zur Basis 10
- ²LOG ... Logarithmus zur Basis 2

$$-2^{(x-1)} \leq I \leq +2^{(x-1)} - 1 \text{ vereinfacht zu } 2^{(x-1)} > I \Rightarrow$$

$$(x-1) > {}^2\text{LOG}(I) \Rightarrow (x-1) > \text{LOG}(I) / \text{LOG}(2) \Rightarrow$$

$$(x-1) > \text{LOG}(10^m) / \text{LOG}(2) \Rightarrow$$

$$(x-1) > m / \text{LOG}(2) \Rightarrow b * 8 - 1 > m / \text{LOG}(2)$$

$$b > \frac{m}{8 * \text{LOG}(2)} + 0.125 \text{ oder als Faustformel:}$$

$$b > m / 2.4 + 1/8$$

Wenden wir dies auf die Y-Koordinate an. Sie kann in Österreich Werte im Bereich von $-200000\text{m} \leq Y \leq +200000\text{m}$ annehmen. In [mm] ausgedrückt benötigen wir also für die Y-Koordinate mindestens 9 signifikante Stellen sowie ein Vorzeichen. Obige Formel angewendet ergibt einen Bedarf von 3.875 Bytes, aufgerundet daher 4 ganze Bytes, um eine Y-Koordinate in [mm] speichern zu können.

Der Vollständigkeit halber muß festgehalten werden, daß man im genannten Beispiel mit insgesamt 8 Bytes für eine gepackte Speicherung der Y- und X-Koordinaten auskommen würde.

Will man nun feststellen, wie groß Zahlenwerte in der Geodäsie werden können, darf man sich nicht auf unsere Landeskoordinaten beschränken. Man muß auch z.B. geographische oder geozentrische Koordinatensysteme berücksichtigen, die auch den Raum bis hinauf zu den geostationären Satelliten umfassen. In der Praxis werden allerdings meist Differenzverfahren angewandt.

2.2.5.4 Gleitkommazahlen

In diesem Kapitel soll nun die Behandlung von Gleitkommazahlen = reelle Zahlen in der Binärarithmetik für die Transformation von Programmen behandelt werden.

Leider sind die Speicherungen von reellen Daten immer wieder ein Problem in der Geodäsie, da sie so unterschiedlich sind. Es geht oft nicht nur darum, alte Daten überhaupt und dann genügend genau lesen und evident halten zu können, sondern der Austausch von binären Daten zwischen verschiedenen Computern ist sogar ein typisch geodätisches Problem.

Besonders dann, wenn in einer Anlage ein Landinformationssystem gehalten wird und somit Objekt-Daten mit Attributen u.s.w. in alphanumerisch verwalteten Datenbanken verwaltet werden, ist es oft sinnvoll, graphische Arbeitsplätze mit lokaler Intelligenz zu verwenden, um den zentralen Rechner mit den Datenbanken zu entlasten. Dies hat aber zur Folge, daß reelle Daten vom Zentralrechner zum lokalen Rechner und zurück übertragen werden müssen. In der Geodäsie sind dies meist Landeskoordinaten mit einer signifikanten Stellenanzahl von durchschnittlich 10 und darüber. Bei einfachen Graphik-Systemen werden normalerweise nur Bildkoordinaten übertragen; das ist viel einfacher und kann durch eine Konvertierung in eine ganze positive Zahl realisiert werden. Diese ganzen positiven Zahlen sind auch verhältnismäßig einfach zu speichern und bei verschiedenen Systemen einheitlich.

Dies ist in der Geodäsie nicht möglich, da sozusagen nie ein Bild erzeugt wird, sondern graphisch Datenbankinhalte erzeugt, verändert und natürlich evident gehalten werden müssen, womit die Maße in der Natur behandelt werden - nicht ein Ergebnis daraus - und somit eine wesentlich höhere Präzision der Daten erforderlich ist als in anderen Disziplinen. Da aber die binäre Speicherung von System zu System unterschiedlich ist, ist bei Neueinführung eines Rechners in einem derartigen Computer-Netz höchste Vorsicht geboten und die Art der Speicherung muß auch unbedingt überprüft werden !

Prinzipiell funktioniert die Rechnung genau so wie bei ganzen Zahlen. Der Unterschied ist das Vorhandensein eines Dezimalpunktes und im allgemeinen eine höhere Stellenanzahl. Bei der Summation während einer Konversion in das binäre Zahlensystem sind die Exponenten eben negativ. Die erste Stelle hinter dem Komma hat als Summand den Wert Ziffer*Basis⁻¹, der zweite Summand den Wert Ziffer*Basis⁻² u.s.w. Dies soll nun anhand eines praktischen Beispielles erläutert werden: Die dezimale Zahl 48.18 sei in die binäre Darstellung überzuführen:

$$43.18 =$$

8	*	10 ⁽⁻²⁾	(=	.08)	+
1	*	10 ⁽⁻¹⁾	(=	.1)	+
3	*	10 ⁽⁺⁰⁾	(=	3)	+
4	*	10 ⁽⁺¹⁾	(=	40)	;



1	*	2 ⁽⁺⁵⁾	(=	32)	+
0	*	2 ⁽⁺⁴⁾	(=	0)	+
1	*	2 ⁽⁺³⁾	(=	8)	+
0	*	2 ⁽⁺²⁾	(=	0)	+
1	*	2 ⁽⁺¹⁾	(=	2)	+
1	*	2 ^(0)	(=	1)	+
0	*	2 ⁽⁻¹⁾	(=	.0)	+
0	*	2 ⁽⁻²⁾	(=	.0)	+
1	*	2 ⁽⁻³⁾	(=	.125)	+
1	*	2 ⁽⁻⁴⁾	(=	.0)	+
1	*	2 ⁽⁻⁵⁾	(=	.03125)	+
1	*	2 ⁽⁻⁶⁾	(=	.015625)	+
1	*	2 ⁽⁻⁷⁾	(=	0.0078125)	=

Σ= 43.1796875

▼▼▼▼▼ ▼▼▼▼▼▼▼▼
101011.0110111²

Wir sehen, daß obige Methode bei Gleitkommazahlen ebenso funktioniert. Es wird aber auch deutlich, daß diese Darstellung nicht denselben Wert beinhaltet wie die dezimale, sondern nur eine Näherung. Um noch genauer zu werden, müßte man nur einige Bits mehr verwenden.

Im folgenden wird die Speicherung einer reellen Zahl (am Beispiel des IBM-PC) erläutert. Eine reelle Zahl kann in allen Zahlensystemen auch durch Mantisse*Basis^{Exponent} dargestellt werden. Die dezimale Zahl 43.18 kann auch als $4.318 \cdot 10^1$ oder $4.318E1$ dargestellt werden - dies wäre eine normalisierte Darstellung, wo die eigentliche Zahl in 2 Teile geteilt wird; nämlich in die Mantisse oder Signifikand und den Exponenten, wobei für den Wert der Mantisse immer gilt: $1 \leq \text{Mantisse} < 10$. Dies gilt analog auch für das binäre Zahlensystem; die oben gezeigte binäre Zahl 101011.0110111_2 kann auch als $1.010110110111_2 \cdot 2^{(-5)}$ angezeigt werden, da im binären System eine Multiplikation einer Zahl mit der Basis des Zahlensystems (nämlich 2) eine Binärpunktverschiebung nach rechts bewirkt - ebenso bewirkt eine Division der Zahl durch die Basis eine Verschiebung des Binärpunktes nach links; dann ist auch die Zahl im Exponent negativ.

Diese binäre Darstellung wird auch bei der Speicherung von reellen Zahlen verwendet. Es gibt daher immer 3 Felder: Ein Vorzeichen der Zahl, das Feld für den Exponenten und das Feld der Mantisse. Negative Zahlen werden wieder durch das Vorzeichen angezeigt und durch das Zweierkomplement erzeugt. Das Vorzeichen des Exponenten (eigentlich das zweite Vorzeichen einer Zahl) wird nicht durch das Zweierkomplement realisiert, sondern es wird ein fester Betrag (bias) zum Exponenten addiert.

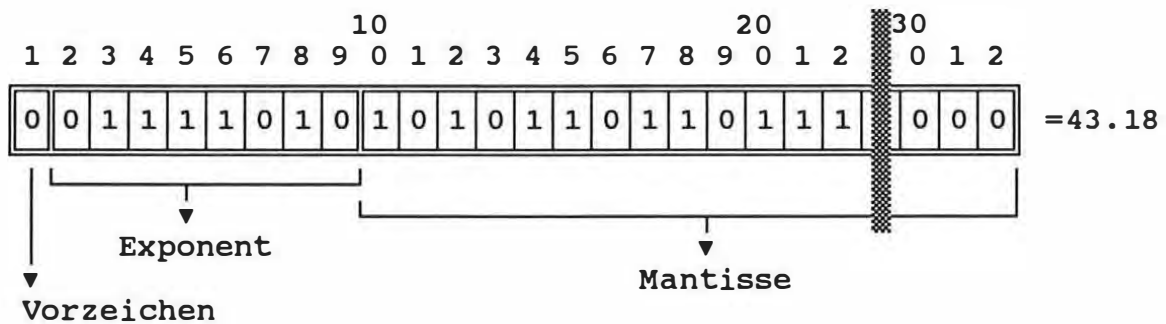
Die Größe des Mantissenfeldes bestimmt die Präzision (engl. precision) oder - umgangssprachlich - die Genauigkeit der Zahl. Je mehr Bits in einem Mantissenfeld definiert sind, desto höher wird die Genauigkeit. Um eine möglichst hohe Genauigkeit zu erlangen, werden auf beinahe allen Computersystemen die Mantisse als normalisierte Zahl in der oben beschriebenen Weise gespeichert - der binäre Wert der Mantisse ist immer größer oder gleich 1 und kleiner als 2. Zwei Überlegungen machen dies notwendig: Erstens erhöht keine Art von führenden Nullen die Genauigkeit einer Zahl. Dies gilt natürlich nicht für nachfolgende Nullen. Führende Nullen würden daher die Genauigkeit verringern. Zweitens verlangt das Speichern einer Mantisse einer reellen Zahl, daß sich der "Dezimalpunkt" im Binärsystem immer an derselben Position befindet. So wird dieser immer hinter die erste Position der Mantisse gesetzt.

Da das erste Bit in der Mantisse immer 1 ist, besteht eigentlich kein Grund mehr, es (und somit den Binärpunkt) überhaupt zu speichern. Allgemein wird daher seine Existenz nur angenommen, obwohl es nirgendwo aufscheint. Für die folgenden Erläuterungen wird auf die eben genannte Praxis keine Rücksicht genommen, um die Methoden der binären Arithmetik übersichtlicher zu gestalten.

Die Ausnahme von dieser Regel besteht darin, daß nur bei der Zahl 0 die gesamte Mantisse 0 wird.

So wie die Größe der Mantisse die Präzision einer Zahl festlegt, so bestimmt die Größe des Exponentenfeldes die jeweils größte und kleinste darstellbare Zahl. So gesehen definiert die Anzahl der Bits im Exponenten-Feld die absoluten und die in der Mantisse die relativen Größen.

Obiges Beispiel sei noch an Hand des IEEE-Standards für einfache REAL-Zahlen beispielhaft veranschaulicht : Hier werden reelle Zahlen in 4 Bytes (32 Bits) gespeichert. Die Mantisse enthält 23 Bits, der Exponent 8 (der additive Wert für den Exponenten ist 127) und das erste Bit ist das Vorzeichen:



Zum Schluß wird wieder eine Faustformel für den Aufwand bei gegebenen Genauigkeitsanforderungen entwickelt:

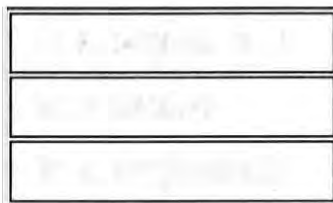
- E Anzahl signifikanter Stellen im Exponenten
- M Anzahl signifikanter Stellen in der Mantisse
- e benötigte Bits im Exponenten
- m benötigte Bits in der Mantisse
- b benötigte Bytes insgesamt
- LOG Logarithmus zur Basis 10

$2^{(e-1)} > 10^E \Rightarrow (e-1) > \text{LOG}(10^E)/\text{LOG}(2) \Rightarrow$

$(e-1) > E/\text{LOG}(2) \Rightarrow e > 1+(E/\text{LOG}(2)) ;$

$2^m > 10^M \Rightarrow m > \text{LOG}(10^M)/\text{LOG}(2) \Rightarrow$

$m > M/\text{LOG}(2) \Rightarrow m > M/\text{LOG}(2) \Rightarrow$ Faustformel:



Im Anschluß an obige Feststellungen wird nun ein Verfahren präsentiert, mit welchem man die Art der binären Speicherung von reellen Zahlen in den Computersystemen überprüfen kann, und zwar am Beispiel von den verschiedenen Varianten der Programmiersprache FORTRAN. Diese wähle ich deshalb, weil es eine in der Geodäsie (meist wegen der großen Zahlenwerte der Landeskoordinaten) sehr verbreitete Programmiersprache ist und weil sie durch ihre Einfachheit geeignet ist, ein solches numerisches Verfahren ohne großen Aufwand zu verdeutlichen.

Die entscheidende Idee ist, daß man in einem kleinen Testprogramm eine Variable mit einem reellen Zahlentyp auf eine Datei unformatiert - also binär - schreibt und dann wieder einliest, allerdings auf eine ganzzahlige Variable. Da man in FORTRAN Variable beliebig auf Dateien oder Zeichen-Variable binär schreiben kann und dann auf eine Variable eines anderen Datentyps kopieren kann, braucht man sich nicht um die Konversion zwischen den verschiedenen Datentypen zu kümmern, da diese vom Laufzeitsystem von FORTRAN erledigt wird.

Dieses Konzept ist in FORTRAN bis in lokale Variable verwirklicht, nämlich durch die EQUIVALENCE-Vereinbarung. Diese erlaubt, 2 Variable verschiedenen Typs auf die gleiche Speicheradresse zu speichern und völlig gleichberechtigt zu behandeln.

Beispiel (in FORTRAN 66):

```
PROGRAM REALTEST
REAL*4 R
INTEGER*4 I
R=1.234
WRITE(1@1) R
READ(1@1) I
WRITE(*,*) I
STOP
END
```

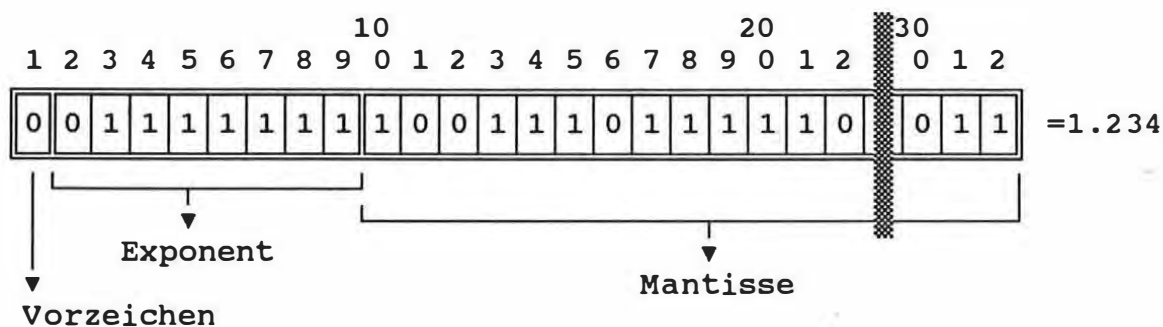
So könnte man als zweite Methode durch eine EQUIVALENCE-Vereinbarung festlegen, daß eine reelle und eine ganzzahlige Variable in demselben physikalischen Speicherbereich des Computers gespeichert wird. Weist man dann der reellen Variable einen Wert zu, so wird in dem entsprechenden Speicherbereich ein bestimmtes Bit-Muster abgelegt. Gibt man dann den Wert der ganzzahligen Variable beispielsweise auf dem Bildschirm aus, so kann man diesen Wert leicht in das binäre System (rück-) umwandeln und erhält so das Bit-Muster der internen Darstellung der reellen Zahl. Dieses Ergebnis kann man anschließend mit den Angaben des jeweiligen Handbuches (soweit überhaupt vorhanden) vergleichen.

Beide Methoden gehen allerdings davon aus, daß man zweifelsfrei weiß, wie die interne Speicherung einer ganzen Zahl im Computer definiert ist. Es gibt z.B. Systeme, welche Daten allgemein wortweise speichern (1 Computer-Wort = 2 Bytes) und dabei die beiden Bytes vertauschen. Leider gibt es nur selten die Möglichkeit, direkt den Inhalt des Speichers übersichtlich anzuzeigen (z.B. in hexadezimaler Form). Weiters ist es manchmal sehr schwer festzustellen, ob der Typ der Variable, das Übersetzungsprogramm, das Betriebssystem oder gar die Hardware des Computers die Art des Bit-Musters steuert und ob der angezeigte Wert nicht bereits Ergebnis einer Interpretation (z.B. des Betriebssystems) ist.

In diesem Beispiel wird eine reelle Variable mit 4 Bytes mit dem dezimalen Wert 1.234 initialisiert. Dieser Wert wird in eine Datei mit der Nummer 1 in die erste Zeile geschrieben und unmittelbar anschließend wieder auf eine INTEGER-Variable eingelesen, die dann am Bildschirm ausgegeben wird. Sehen wir uns nun an, wie die dezimale Darstellung dieser Zahl aussehen müßte, wenn die reelle Zahl in der oben geschildert Weise gespeichert wird. Zu diesem Zweck wandeln wir die dezimale Zahl 1.234 in die binäre Darstellung um:

$$1.234 = 1.00111011111001110110110^2$$

Da dies bereits die normalisierte Zahl ist, ist der Exponent 0, und wir finden im Exponentenfeld daher nur das additive Glied (127). Obige Zahl müßte daher in folgender Weise gespeichert sein:



Wenn wir nun die 32 Bits als ganzes betrachten, so kommt folgendes heraus (Zur besseren Übersichtlichkeit habe ich Leerzeichen eingefügt):

$$0\ 01111111\ 10011101111100111011011^2$$

Wenn wir diese 32 Bits nun wieder umwandeln in das dezimale Zahlensystem, so entspricht dieser fiktiven binären Zahl die dezimale ganze Zahl 1070528987 ! Diese muß auch in obigem Beispiel angezeigt werden, wenn die Annahmen über die Art der Speicherung richtig waren.

Es ist klar, daß es von Vorteil ist, dieses Verfahren zu automatisieren. Der Weg wäre der, daß man vorerst Annahmen über die Art der Speicherung von reellen Zahlen trifft und dann vom Programm verschiedene Kombinationen durchprobieren läßt - in derselben Weise, wie ich es eben in einzelnen Schritten erläutert habe. So kann man in einem Programm ein Modell der Datenspeicherung aufstellen, wobei folgende Parameter variiert werden können:

- 1) Anzahl der Bytes und somit der Bits des Datentyps
- 2) Anzahl der Bits in der Mantisse bzw. im Exponenten
- 3) Wert des additiven Gliedes im Exponent-Feld
- 4) Position von Vorzeichen, Exponent und Mantisse im Feld

Sollte trotz dieser Variationen keine Lösung gefunden werden, so kann man die Annahme, daß in der Mantisse eine mit dem Zweierkomplement normalisierte Zahl steht, ebenso überprüfen wie, daß umgekehrt im Exponenten kein Zweierkomplement, sondern eine Additive Zahl zur Anwendung kommt.

Wie unterschiedlich die Speicherung von reellen Zahlen in verschiedenen Computersystemen sein kann, wird durch folgende Beispiele belegt:

HP3000 (REAL - FORTRAN 66) : /F32.22/E256/NK2

	±	Exponent	.	Mantisse	
+1.234	=	0	100:0000:00		0
+1.235	=	0	100:0000:00		0
+2.47	=	0	100:0000:01		0
-1.234	=	1	100:0000:00		0

▲ Bit31
 ▲ Bit21
 ▲ Bit00

Feldgröße : 32 Bits

1<Mantisse<2 : Bit00-Bit21 = 22 Bits : +1 Bit('1') vor Komma

Dezimalpunkt : Bit21/Bit22

Exponent : Bit22-Bit30 = 9 Bits : 256 addiert

Vorzeichen : Bit31 = 1 Bit : 0 => + , 1 => -

Anz. Ziffern : 6

Wertebereich : ±1.999999*10^(±255)

HP3000-XL (REAL*4 - FORTRAN 77) : /F32.23/E127/NK2

	±	Exponent	.	Mantisse	
+1.234	=	0	1111:1	001:1101:1111:0011:1011:0110	0
+1.235	=	0	1111:1	001:1110:0001:0100:0111:1011	0
+2.47	=	0	0000:0	001:1110:0001:0100:0111:1011	0
-1.234	=	1	1111:1	001:1101:1111:0011:1011:0110	0

▲ Bit31
 ▲ Bit22
 ▲ Bit00

Feldgröße : 32 Bits

1<Mantisse<2 : Bit00-Bit22 = 23 Bits : +1 Bit('1') vor Komma

Dezimalpunkt : Bit22/Bit23

Exponent : Bit23-Bit30 = 8 Bits : 127 addiert

Vorzeichen : Bit31 = 1 Bit : 0 => + , 1 => -

Anz. Ziffern : 6

Wertebereich : ±1.999999*10^(±127)

HP9000 (REAL*4 - FORTRAN 77) : /F32.23/E127/NK2

	±	Exponent	.	Mantisse	
+1.234 =	0	011:1111:1		001:1101:1111:0011:1011:0110	⊞
+1.235 =	0	011:1111:1		001:1110:0001:0100:0111:1011	⊞
+2.47 =	0	100:0000:0		001:1110:0001:0100:0111:1011	⊞
-1.234 =	1	011:1111:1		001:1101:1111:0011:1011:0110	⊞

▲ Bit31 ▲ Bit22 ▲ Bit00

Feldgröße : 32 Bits

1 ≤ Mantisse < 2 : Bit00-Bit22 = 23 Bits : +1 Bit('1') vor Komma

Dezimalpunkt : Bit22/Bit23

Exponent : Bit23-Bit30 = 8 Bits : 127 addiert

Vorzeichen : Bit31 = 1 Bit : 0 => + , 1 => -

Anz. Ziffern : 6

Wertebereich : ±1.999999*10^(±127)

HP1000 (REAL*4 - FORTRAN 77) :

	.±	Mantisse	±Exponent	
+1.234 =	0	100:1110:1111:1001:1101:1011	0000:0010	⊞
+1.235 =	0	100:1111:0000:1010:0011:1101	0000:0010	⊞
+2.47 =	0	100:1111:0000:1010:0011:1101	0000:0100	⊞
-1.234 =	1	011:0001:0000:0110:0010:0101	0000:0010	⊞

▲ Bit31 ▲ Bit08 ▲ Bit00

Feldgröße : 32 Bits

0 ≤ Mantisse < 1 : Bit08-Bit30 = 23 Bits : +1 Bit('1') vor Komma

Dezimalpunkt : Bit31/Bit32 M < 0 => 2er-Komplement

Exponent : Bit00-Bit07 = 8 Bits : E < 0 => 2er-Komplement

Vorzeichen : Bit31 = 1 Bit : 0 => + , 1 => -

Anz. Ziffern : 6

Wertebereich : ±1.999999*10^(±127)

2.2.6 Bits bei Übertragungen

Schließlich muß noch auf die Bedeutung spezieller Bits bei der physikalischen Übertragung von Programmen oder Meßergebnissen hingewiesen werden (siehe auch Kapitel 'physikalische Übertragung'). Hier treten Bits auf, die Kontroll-Funktionen haben; zum Beispiel gibt es bei der seriellen Übertragung im allgemeinen Start-, Stop- und Paritäts-Bits. Dies sind bereits die wesentlichsten Elemente eines Hardware-Protokolles:

Ein Start-Bit zeigt den Beginn einer Übertragung eines Zeichens über eine serielle, asynchrone Verbindung an, das Stop-Bit beendet diese Übertragung und das Paritäts-Bit zeigt an, ob die Summe der jeweils gesetzten Bits eines einzelnen Zeichens gerade oder ungerade sein sollte. Dies wäre die einfachste Möglichkeit, um zu kontrollieren, ob die Sendung von der Quelle auch gut beim Ziel angekommen ist.

2.3 Zeichen

2.3.1 Bitmuster

Ein Zeichen ist die Repräsentation einer Folge von binären Informationen (Bits), die zu einer Gruppe zusammengefaßt werden (Byte). Meist umfaßt ein Byte eine Folge von acht Bits. Beispiel: '01000001' ist die duale Darstellung (die der tatsächlichen physikalischen Speicherung in einem Computer entspricht) der dezimalen Zahl '65' und der Zahl '41' im hexadezimalen Zahlensystem.

Bei 8 Bits sind 2^{*8} (Zwei hoch acht) = 256 Kombinationen pro Byte möglich; d.h. der dezimale Wert eines Bytes mit 8 Bits kann zwischen 0 und 255 liegen. Jeder dieser Kombinationen von Bits (Bitmuster) kann nun ein Zeichen zugeordnet werden. Die Gesetze einer solchen Abbildung stellen ein Codesystem dar. Eines der bekanntesten sind die ASCT. obigen Beispiel würde der dezimalen Zahl '65' der Buchstabe 'A' als einzelnes Zeichen zugeordnet werden.

2.3.2 Sonderzeichen

Leider stellen diese Code-Systeme keine Norm dar. Im Rahmen einer Umstellung der Programme müssen daher die beiden Code-Systeme auf das genaueste verglichen werden. Oft stellen sich dann Unterschiede heraus, obwohl von den Computerherstellern dasselbe Code-System zitiert wird. Nur bei wenigen Zeichen kann man sich halbwegs darauf verlassen, daß die Darstellung einheitlich gehandhabt wird: Das betrifft die Buchstaben 'A' bis 'Z', 'a' bis 'z' und die Ziffern '0' bis '9'. Alle anderen Zeichen müssen als Sonderzeichen klassifiziert werden.

Die ursprünglichen Programmtexte können nun Zeichen enthalten, die im Zielsystem nicht vorhanden oder anders dargestellt werden; speziell dann, wenn es sich um zwei verschiedene Computerhersteller handelt. Besonders zu erwähnen sind hier im Deutschen die Umlaute und 'ß'.

Bei einer Übertragung ohne besondere Konvertierung der Texte werden letztlich binäre Daten übertragen. Es kann somit vorkommen, daß zum Beispiel ein Bitmuster übertragen wird, welchem im einen System ein 'ä' zugeordnet wird, im anderen aber einem '|'. Somit würden im übertragenen Text überall dort, wo 'ä' stehen sollten, '|' angezeigt.

2.4 Zeilen

2.4.1 Logische Definition

Eine Zeile (Ein Record in einer Text-Datei) ist eine Kette von hintereinanderstehenden Zeichen, die auf irgendeine Weise beendet wird. Dies kann bei sequentiellen Text-Dateien ein spezielles Zeichen sein oder ein Eintrag in eine Verzeichnis-Tabelle.

2.4.2 Zeilen bei Umstellungen

Die Hauptschwierigkeit bezüglich Zeilen liegt bei Umstellungen in den jeweiligen Definitionen der Dateien, besonders wenn diese sehr verschieden ausgefallen sind.

Als Beispiel möchte ich nun anführen, daß es grundsätzlich 2 Arten von Records in Dateien gibt:

- 1) Solche mit Zeilen, die verschieden lang sein können und
- 2) solche mit Zeilen, die alle gleich lang sein müssen.

Bei einer Transformation ist nun Sorge zu tragen, daß die verschiedenen Datei-Typen kompatibel bleiben. Es kann dabei notwendig werden, daß man zum Beispiel eine sequentielle Text-Datei durch Anfügen von Leerzeichen so verändert, daß alle Zeilen gleich lang und somit geeignet für eine Übertragung in eine Datei mit fixen Zeilen-Längen werden.

Darüber hinaus könnte es vorkommen (z.B. von HP3000 unter MPE nach HP1000 unter RTE), daß man an alle die Zeilen, die kürzer als die längste Zeile der Datei sind, solange Leerzeichen anhängt, bis alle Zeilen gleich lang sind. Allerdings schneidet das Programm, welches die Datei von der HP3000 zur HP1000 überträgt, nachfolgende Leerzeichen ab. Hier kann man nur mehr so vorgehen, daß man das letzte Zeichen einer Zeile mit einem definierten Zeilenende zusätzlich durch ein spezielles, durch den Benutzer definiertes Zeichen ersetzt.

Natürlich müssen Programme, die diese Datei dann abarbeiten werden, von diesem speziellen Zeichen wissen und es nicht zu Berechnungen verwenden, sondern als eine Art Kommentarzeichen ignorieren.

2.5 Dateien

2.5.1 Logische Definition

Eine Datei (Ein File) ist eine Kette von aufeinanderfolgenden Zeilen (Records). Die Zeilen werden dabei durch ein spezielles Zeichen voneinander getrennt oder durch einen Eintrag in einem Datei-Verzeichnis definiert.

2.5.2 Dateien bei Transformationen

Was Dateien als ganzes betrifft, so führen meist - abgesehen von anscheinend immer unerwarteten Problemen mit dem Speicherplatz - die Namen bzw. die verschiedenen Pfade und Verzeichnisstrukturen der Betriebssysteme zu Problemen.

Besonders die Verwendung von Sonderzeichen in Datei-Namen ist abzuraten, da die Weiterverwendung solcher Dateinamen in anderen Umgebungen meist unmöglich ist. Dann müssen neue Namen gemäß der neuen Konventionen gefunden werden, die aber dennoch an die alten erinnern und somit eine Umstellung erleichtern.

2.5.3 Erweiterungen (Extensions) der Datei-Namen

Ein weiteres Problem besteht darin, daß oft die Erweiterungen von Datei-Namen mit bestimmten Datei-Typen korrespondieren müssen.

Beispielsweise sind die Erweiterungen im Datei-Namen (Extensions) unter dem Betriebssystem MS-DOS von größter Bedeutung: Erweiterungen als '.EXE' und '.COM' signalisieren die Ausführbarkeit eines Programmes, und da das Betriebssystem davon ausgeht, ist es unmöglich, ein anderes (nicht ausführbares) Programm als ein solches mit den genannten Erweiterungen im Datei-Namen zum Laufen zu bringen. Andere Betriebssysteme, wie CMS von IBM vergeben mehrere Arten der Erweiterungen in den Datei-Namen (Typ,Disk) ... allerdings meist automatisch und ohne eine Möglichkeit für den Anwender, in diese Logik einzugreifen.

2.5.4 Programme mit mehreren Dateien

Ein Programm ist für den Programmierer im einfachsten Fall vorerst eine Datei, die mit einem Text-System erstellt wird und die einen Text (Source-Code) enthält, der in einer Programmiersprache geschrieben ist und möglichst übersichtlich sein sollte. Erst ein Übersetzungsprogramm (Compiler) und ein Verbindungsprogramm (Linker) setzen im Prinzip diesen Text in ein ausführbares Programm, eine nicht mehr mit einem Textsystem anzeigbare Datei, um.

Große Programme bestehen dann im allgemeinen aus einer Sammlung von Text-Dateien, die alle Teile des Programmes beinhalten.

Eine Transformation solcher Programme ist dann am einfachsten, wenn jede Datei je ein Modul enthält. Dies wäre in FORTRAN je eine SUBROUTINE oder FUNCTION, in PASCAL je eine PROCEDURE oder FUNCTION u.s.w. Wenn man feststellen kann, daß eine Datei implizit immer identisch ist mit einem Programm oder wenigstens mit einem abgeschlossenen Modul des Programmes mit einer klaren Schnittstelle, so kann die explizite Behandlung von Modul-Grenzen und damit ein nicht unerheblicher Aufwand in einer Transformation entfallen.

Im folgenden sei daher bei allen Arten der Transformation angenommen, daß jeder Datei eine logische Einheit eines Programmes oder einer Registriereinheit entspricht.

3 Bitorientierte Transformationen

3.1 Bits ersetzen

3.1.1 Konvertierung

Das Austauschen von einzelnen Bits ist die einfachste Form einer Transformation. Wenn Zeichen in eine Text-Datei geschrieben und anschließend einzelne Bits verändert werden, so erhält man beim Lesen der so geänderten Datei andere Zeichen als die ursprünglich geschriebenen.

Beispiel : Wenn Bit 8 eines Zeichens = 1 ist, so ersetze alle 8 Bits des Zeichens durch 00100000.

Dies ist eine typische bitorientierte Transformation - der einfachste Fall einer Umformung. Angewendet auf alle Zeichen einer Datei bedeutet dies, daß alle Zeichen, die eine Bit-Sequenz repräsentieren, deren 8. Bit gesetzt ist und deren dezimaler Wert somit über 127 liegt, durch ein Leerzeichen ersetzt werden (in der ASCII-Konvention entspricht die binäre Zahl 00100000 [dezimal 32] einem Leerzeichen). So würden beispielsweise im IBM-PC alle Block-Graphik-Zeichen wie -| \bar{r} u.s.w durch Leerzeichen ersetzt werden.

Weiters ist es technisch einfacher, durch ein Programm Zeichen auszutauschen, als einzelne Bits in einem Speicher zu verändern. Daher entspricht der Austausch von einzelnen Bits dem Austausch von je einem Zeichen, sodaß die Anzahl der Zeichen insgesamt vor und nach einer bitorientierten Transformation von Programmen konstant bleibt.

Eine Zeichenorientierte Transformation hingegen (siehe später) bedeutet, daß Zeichen auch mehrfach ausgetauscht werden, sodaß die Anzahl der Zeichen insgesamt nicht konstant bleibt.

Es erscheint zweckmäßig, eine Tabelle für eine bitorientierte Transformation zu erstellen, die in jeder Zeile als Quelle und als Ziel je ein einzelnes Zeichen beinhaltet. Die Anzahl der Tabelleneinträge ist dann gleich der Anzahl der umzuformenden Bitkombinationen.

Eine zweite (schnellere) Möglichkeit besteht darin, eine Tabelle zu erstellen, die ebensoviele Einträge wie mögliche Zeichen enthält. Aus einem Zeichen kann dann ein Index berechnet werden, welcher wiederum Eingangsparameter in der Tabelle ist.

Diese Methoden sind natürlich nur dann sinnvoll, wenn das zu transformierende Zeichen isoliert betrachtet werden kann und nicht für die Transformation in Abhängigkeit von anderen Zeichen oder Parametern steht.

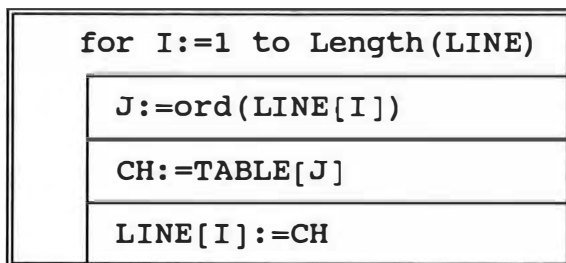
3.1.2 Prozedur NEWBIT

3.1.2.1 Einführung

Im folgenden soll demonstriert werden, wie eine bitorientierte Transformation allgemein, speziell aber von Programmen oder auch von übertragenen Meßergebnissen durchgeführt werden kann. Dabei soll davon ausgegangen werden, daß alle Zeichen isoliert betrachtet werden können. Die folgende Prozedur 'NEWBIT' erlaubt die Transformation von Bits innerhalb einer Zeile. Dies zeige ich mit der Programmiersprache 'TURBO-PASCAL 5.0':

3.1.2.2 Struktogramm von NEWBIT

NEWBIT



3.1.2.3 Source-Code von NEWBIT

```

{-----}
{ Die Prozedur NEWBIT verändert systematisch Bits in einer }
{ Zeile. }
{-----}

procedure NEWBIT
(
var Line : string ;{ Zu transformierende Zeile }
TABLE : string [255] );{ Transformations-Tabelle :
{ Eingangsparemeter ist Index i }
{ eines Zeichens in ASCII-Code- }
{ System => TABLE[i] ist das }
{ einzusetzende Zeichen. }

begin
for p:=1 to length(Line) do Line[p]:=TABLE(ord(Line[p]));
end;

```


3.1.3 Kompliziertere Formen

Schwieriger wird eine bitorientierte Transformation dann, wenn man die einzelnen Zeichen nicht mehr isoliert betrachten kann und sie mit anderen Bedingungen, zum Beispiel mit benachbarten Zeichen, verknüpfen muß.

Dies tritt dann auf, wenn nicht einfache Texte sondern binäre Daten transformiert werden müssen. Im Falle von Koordinaten müssen dann verschiedene Speicherkonventionen in Einklang gebracht werden.

Beispielsweise ist die Speicherung einer reellen Zahl in verschiedenen Systemen auch dann verschieden, wenn die gleiche Programmiersprache verwendet wurde. Einfache reelle Zahlen werden meist jeweils mit je 4 Zeichen (Bytes) abgespeichert. Wenn man nun annimmt, daß jedes Zeichen die Repräsentation von 8 Bits ist, so hat man eine Kette von 32 Bits vor sich. Darin enthalten sind je ein Bit für das Vorzeichen der Zahl selbst und des Exponenten, die Zahl selbst sowie ihr Exponent. Leider werden von verschiedenen Systemen diese Bits verschieden verteilt - manchmal ist das erste Bit das Vorzeichen der Zahl, manchmal steht dieses Bit an anderer Stelle.

3.2 Genereller Austausch von Bits

3.2.1 Konvertierung

Wenn man nun Bits über eine ganze Datei austauschen möchte und man überdies einzelne Zeichen immer isoliert verändert, so wird man nach Möglichkeit direkt Zeichen für Zeichen von der zu transformierenden Datei einlesen, Bits verändern und die resultierenden Zeichen wieder schreiben. Man sollte also nicht den Umweg über Zeilenstrukturen gehen, da diese oft die Veränderung von Sonderzeichen erschweren.

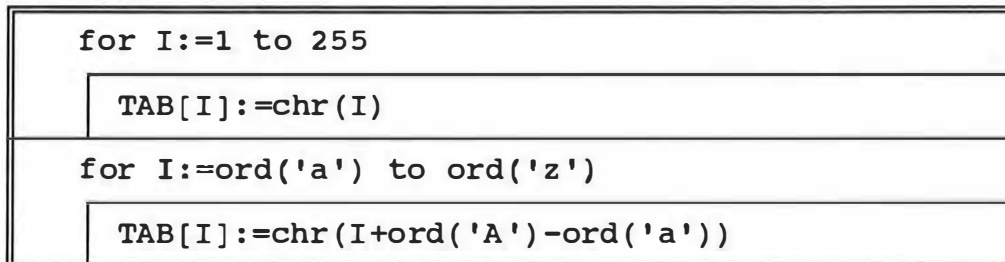
3.2.2 Prozedur CHANGE_BITS

3.2.2.1 Einführung

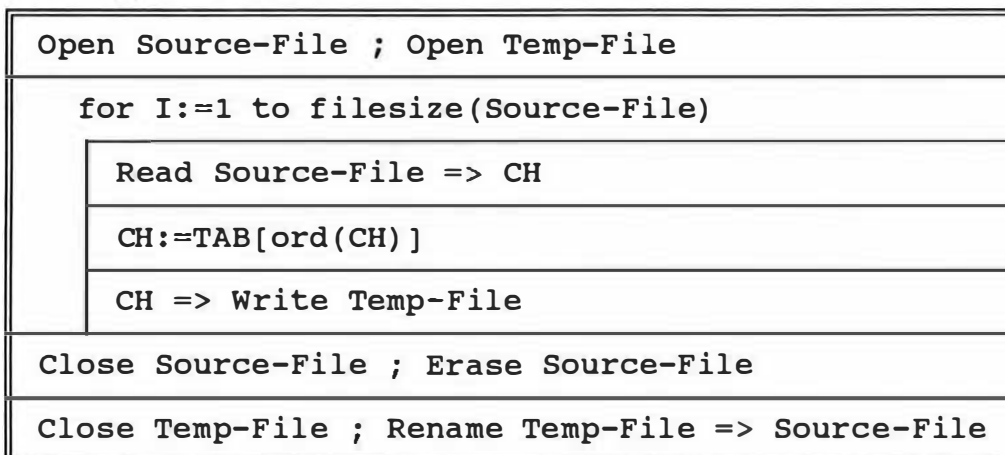
Im folgenden wird der Algorithmus CHANGE_BITS vorgestellt, der Bits über eine ganze Datei austauscht. Dabei sei angenommen, daß wenn der dezimale Wert eines Zeichens zwischen 97 und 122 liegt, das 6. Bit gelöscht wird. Damit würden (laut ASCII-Konvention) alle Kleinbuchstaben in Großbuchstaben umgewandelt werden (wieder mit der Sprache TURBO-PASCAL 5.0). Das Beispiel dient nur der Veranschaulichung der Art einer solchen Tabelle. In Turbo Pascal gibt es nämlich eine vordefinierte Funktion UPCASE - diese wird aber nicht verwendet:

3.2.2.2 Struktogramm von CHANGE_BITS

CHANGE_BITS (beim 1. Mal Tabelle initialisieren)



CHANGE_BITS (Transformation)



3.2.2.3 Source-Code von CHANGE_BITS

```

{-----}
{ Die Prozedur CHANGE_BITS macht aus allen Kleinbuchstaben }
{ einer Datei Großbuchstaben }
{-----}

procedure CHANGE_BITS
(
File_Name : string ); { Datei-Name der zu ändernden Datei }

var Source : file of char ; { Datei-Variable }
    Ch      : char          ; { Eingelesenes Zeichen }
    Temp    : file of char ; { Teporäre Hilfs-Datei }
    i       : longint      ; { Index des Zeichens }

begin
{-----}
{ Tabelle initialisieren }
{-----}
for i:=0 to 255 do Tab[i]:=chr(i);
for i:=ord('a') to ord('z') do
    Tab[i]:=chr(i+ord('A')-ord('a'));
{-----}
{ Dateien öffnen }
{-----}
assign(Source,File_Name); reset(Source);
assign(Temp,'CHANGE.TMP'); rewrite(Temp);
{-----}
{ Dateien ändern }
{-----}
for i:=1 to filesize(Source) do
begin
    read(Source,Ch); write(Temp,Tab[ord(Ch)]);
end;
{-----}
{ Dateien schließen }
{-----}
close(Temp); close(Source);
erase(Source); rename(Temp,File_Name);
end;

```

4 Zeichenorientierte Transformationen

4.1 Zeichen ersetzen

4.1.1 Konvertierung

Die zeichenorientierte Transformation ist nach der bit-orientierten Transformation die nächst-höhere Stufe einer Umformung und man versteht darunter das Austauschen von Zeichen in Text-Dateien, wobei sich die Gesamtanzahl von Zeichen in der Datei verändern kann.

Beispiel : Ersetze 'Autobahn' durch 'Radweg' in der Datei.

Dies ist ein Beispiel für eine typische zeichenorientierte Transformation. Wie man sieht, besteht die zu suchende Zeichenkette aus 8 Zeichen, die einzusetzende hingegen aus 6 Zeichen. Die doppelte Anzahl der Fundstellen in der Datei ergibt somit die Anzahl der Zeichen, um die die Datei kürzer wird.

Für zeichenorientierte Transformationen gilt, daß es am einfachsten ist, vorerst Zeilen aus einer Text-Datei zu lesen, darin Zeichenketten zu verändern und schließlich die geänderten Zeilen wieder auf die Datei zu schreiben, wobei die jeweilige Zeile als Puffer für eventuelle Änderungen der Zeilenlänge fungiert. Bei den meisten Systemen ist beispielsweise das Einfügen eines einzelnen Zeichens in eine Datei schwieriger als das Ersetzen einer Zeile durch eine veränderte, längere Zeile.

Die Ausnahme dieser Regel sind die zeichenorientierten Transformationen, wo Zeichenketten über mehrere Zeilen hinweg geändert werden müssen. Dieser Fall ist auch der Grund, weshalb man zeichenorientierte Transformationen nicht durch sequentiellles Abarbeiten der Quell-Datei lösen kann.

Vor der eigentlichen Konvertierung müssen die Texte außerdem derart verändert werden, daß sie 'lesbar' werden. Die Bitmuster sollen so verändert werden, daß ihre Darstellungen in beiden Systemen übereinstimmen oder daß wenigstens äquivalente Zeichen eingesetzt werden. Zum Beispiel könnte im ersten System das ASCII-Sonderzeichen '+' (Dezimal 197) vorkommen, welches das Zielsystem nicht kennt, sondern als '■' darstellen würde. In diesem Fall müßte man alle '■' ersetzen durch '+', um eine ähnliche Darstellung zu erhalten.

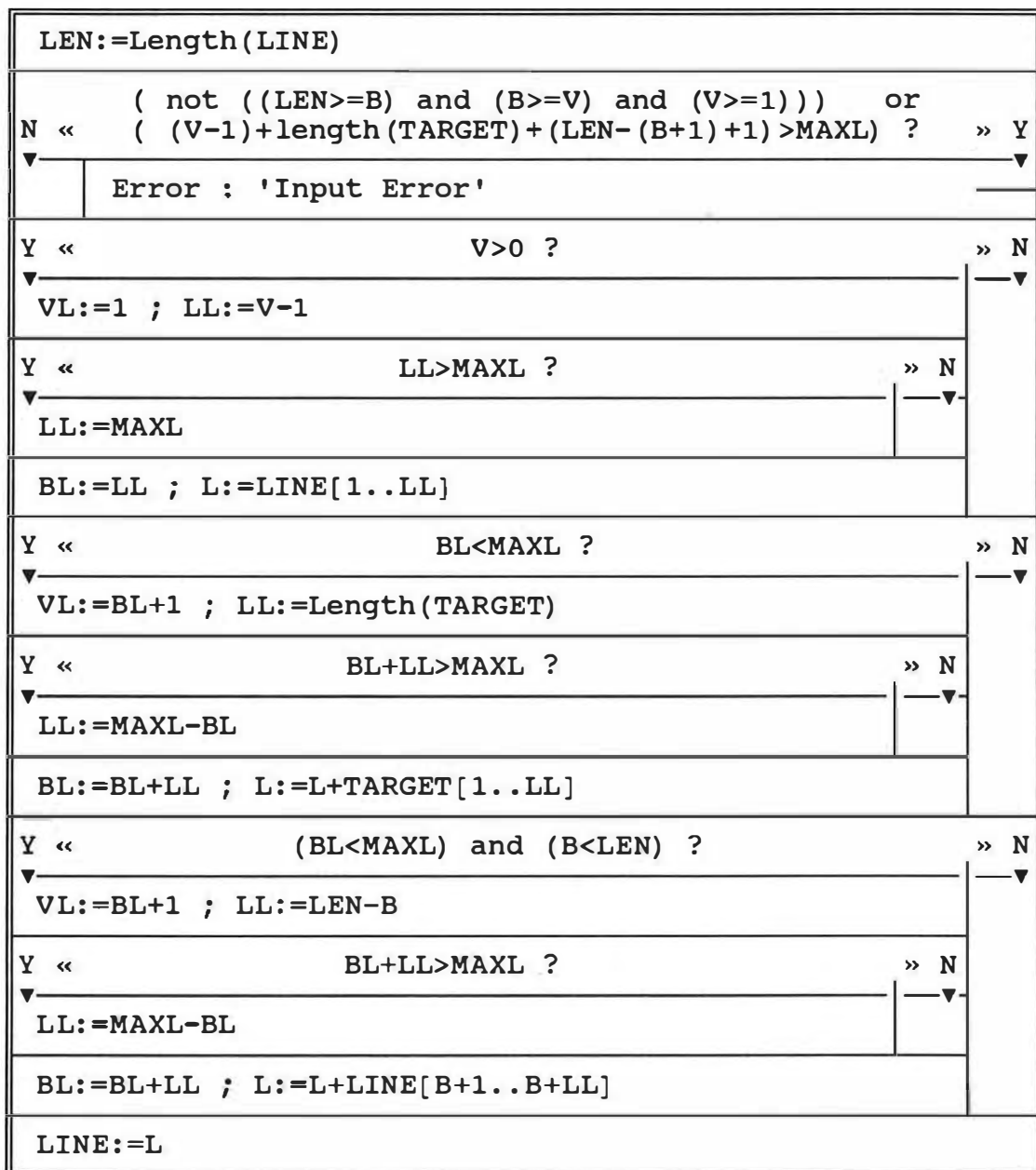
4.1.2 Prozedur CHANGE

4.1.2.1 Einführung

Um etwas allgemeinere Betrachtungen zu ermöglichen, definieren wir nun eine Prozedur 'CHANGE', die in einer Zeile nicht nur ein einzelnes Zeichen austauschen kann, sondern auch Zeichengruppen. Diese äußerst triviale Definition führe ich mit der Programmiersprache 'TURBO-PASCAL 5.0' durch :

4.1.2.2 Struktogramm von CHANGE

CHANGE



4.1.2.3 Source-Code von CHANGE

```

{
{ CHANGE : Setzt einen String TARGET in den String LINE }
{ von Position V bis Position B von LINE ein. }
{ ch80 ist als Datentyp string[80] definiert. }
{
}
procedure CHANGE (var LINE:ch80 ;{ Input/Output String }
                  V,B:integer;{ Positionen in LINE }
                  TARGET:ch80 ;{ einzusetzende Zeichen }

const MAXL = 80 ;{ Maximal erlaubte Länge eines Strings}
var VL : integer ;{ aktuelle Position 'von' in LINE }
    LL : integer ;{ aktuelle Länge des Substrings }
    BL : integer ;{ aktuelle Position 'bis' in LINE }
    L : ch80 ;{ Zwischenspeicher für LINE }
    LEN : integer ;{ Länge des Input-Parameters LINE }

begin
{
{ Eingangspartner prüfen }
{
LEN:=length(LINE);
if ( not ((LEN>=B) and (B>=V) and (V>=1)) ) or
( (V-1)+length(TARGET)+(LEN-(B+1)+1)>MAXL) then
begin
writeln('CHANGE: Input Error !');halt;
end;
{
{ Ersetzen }
{
if (V>0) then
begin
VL:=1;
LL:=V-1;
BL:=LL;
L:=copy(LINE,1,LL);
end;
if (BL<MAXL) then
begin
VL:=BL+1;
LL:=length(TARGET);
BL:=BL+LL;
L:=L+copy(TARGET,1,LL);
end;
if (BL<MAXL) and (B<LEN) then
begin
VL:=BL+1;
LL:=LEN-B;
BL:=BL+LL;
L:=L+copy(LINE,B+1,LL);
end;
LINE:=L;
end;

```

4.1.3 Grenzfälle

Beispiel : Die Zeile 'Dies ist ein Test' sei gegeben. In dieser Zeile soll nun 'ist' durch 'sei' ausgetauscht werden. Das Wort 'ist' belegt die Positionen 6 bis 8 in der Zeile, die der Variable LINE zugewiesen werden sollte. Die Variable LINE hat nach Durchführung von CHANGE(LINE,6,8,'sei') den Wert 'Dies sei ein Test'.

Ich habe obiges Beispiel deshalb so ausführlich dokumentiert, weil man damit bereits einige der wesentlichsten Probleme einer Umstellung aufzeigen kann. Schon im ersten Abschnitt der Prozedur werden nicht nur die Eingangsparameter hinsichtlich der Relevanz ihrer Werte überprüft, sondern auch festgestellt, ob die Länge des Ergebnisses die maximal erlaubte Länge MAXL übersteigen würde. Wenn das der Fall wäre, kann dieses Werkzeug nicht verwendet werden !

Daraus läßt sich folgern, daß es der Flexibilität wegen anzuraten ist, während der Umbildung genügend Reserven in der Länge der Zeilen vorzusehen und erst danach zu überprüfen, ob MAXL letztlich überschritten würde. Somit können Zwischenergebnisse der Umstellung auch längere Zeilen aufweisen. Das bedingt aber auch, daß Umstellungen allgemein wesentlich mehr Speicherplatz in den Computeranlagen benötigten als die Summe der ursprünglichen und der transformierten Programmtexte zusammen.

Die Befehle 'if (...>MAXL) ...' in dem Algorithmus CHANGE sind von der Logik her redundant - sie können nie durchgeführt werden, demonstrieren aber das Abschneiden der Zeichen, die über die Länge MAXL hinausragen würden. Würde der Abschnitt 'Eingangs-Parameter prüfen' in CHANGE weggelassen, könnte der Algorithmus trotz Überschreitens von MAXL durchgeführt werden, indem die überzähligen Zeichen einfach abgeschnitten werden.

Das Problem ist nun, daß zum Beispiel in der Programmiersprache TURBO-PASCAL 4.0 die vordefinierte Prozedur INSERT genau diese Art des Abschneidens ohne Warnung praktiziert, genau so wie manche Editoren; z. B. QUAD unter MPE (HP3000).

Das Beispiel zeigt einerseits, daß man ein Werkzeug zur generellen Änderung von Zeichenkombinationen nur nach sorgfältiger Prüfung der Grenzfälle verwenden darf, daß andererseits diese Art der Änderung sich an unabhängigen Zeichen orientiert und nicht an ihrer Bedeutung; es kann somit nicht unterschieden werden, ob es jeweils eine Änderung in einer Text-Konstanten oder einer Variablen ist.

Beispiel : System 1 und System 2 kennen gleichermaßen Umlaute, aber nur System 1 ließ diese in Variablennamen zu. Das Problem besteht dann darin, alle Umlaute durch Buchstabenkombinationen zu ersetzen; zum Beispiel 'ä' durch 'ae' aber nicht innerhalb von Texten ! 'Ä\$="Töpfer"' müßte dann 'Ae\$="Töpfer"' ergeben und nicht 'Ae\$="Toepfer"'. Mit anderen Worten: Es kommt darauf an, die Eingangsparameter V und B des Algorithmus CHANGE zu bestimmen.

4.2 Genereller Austausch von Zeichen

4.2.1 Tabelle CHANGE_TAB

Nun wird der oben beschriebene Algorithmus CHANGE für die zeichenorientierten Transformationen nun dazu verwendet, generelle Änderungen zu demonstrieren - nämlich den Austausch von Zeichenkombinationen über eine ganze Datei.

Zu diesem Zweck wird formal eine Tabelle angelegt, die die gewünschten Änderungen beinhalten soll. Diese Tabelle kann bei einer Realisierung in einem Programm ein Array sein oder noch besser eine Datei, welche auf einfache Weise durch ein Textsystem verändert werden kann.

Die Tabelle wird CHANGE_TAB genannt und darin 2 Spalten mit den Namen CHANGE_FROM und CHANGE_TO definiert, die beide Zeichenkombinationen beinhalten. Die Anzahl der Zeichen pro Spalte sei in CHANGE_FROM mindestens 1, in CHANGE_TO mindestens 0 und in beiden durch maximal 80 begrenzt. Die Anzahl der Zeilen sei beliebig groß und mit CHANGE_LEN symbolisiert. CHANGE_FROM enthalte die Zeichenkombinationen, nach denen in der Datei gesucht werden soll und CHANGE_TO beinhalte die Zeichen, die die korrespondierenden in derselben Zeile von CHANGE_FROM ersetzen sollen.

Beispiel: Nehmen wir an, wir wollen in einer Datei jedes Vorkommen von 'SUBROUTINE' durch 'procedure' ersetzen, weiters wollen wir 'CALL' durch '' austauschen, also es löschen, wo es vorkommt. Dann würde die Tabelle so aussehen:

Tabelle CHANGE_TAB	CHANGE_FROM	CHANGE_TO
Zeile 1	'SUBROUTINE'	'procedure'
Zeile 2	'CALL'	''

CHANGE_LEN = 2

In TURBO-PASCAL 5.0 hätte diese Tabelle folgende Darstellung:

```
CHANGE_FROM[1]:='SUBROUTINE';
CHANGE_TO [1]:='procedure' ;
CHANGE_FROM[2]:='CALL'      ;
CHANGE_TO  [2]:=''          ;
CHANGE_LEN:=2;
```

4.2.2 Tabelle CHANGE_FILES

Um Änderungen von Programmen bequemer durchführen zu können, muß man annehmen, daß nicht nur eine einzelne Datei von den Änderungen betroffen sein könnte, sondern daß unter Umständen mehrere Dateien von derselben Gruppe von Änderungen betroffen sein könnten. Daher definiere ich nun eine zweite Tabelle CHANGE_FILES, welche in ihrer einzigen Spalte mit dem Namen CHANGE_FILES alle die Datei-Namen der Dateien beinhaltet, die geändert werden sollen. Pro Zeile seien 1 bis 80 Zeichen erlaubt. Die Anzahl der Zeilen sei CHANGE_DIR.

Wenn auch die Datei-Namen in einer Tabelle stehen und nicht bei einer Realisierung des Programmes vom Benutzer angefordert und von diesem dann in den Computer eingetippt werden müssen, hat das den Vorteil, daß durch die Tabelle ständig eine übersichtliche Aufstellung über das Ausmaß der Änderungen besteht und außerdem die Tabelle für andere Änderungen derselben Dateien mehrfach herangezogen werden kann.

Beispiel: Nehmen wir nun an, die betroffenen Datei-Namen (MS-DOS) wären 'MAIN.FOR', 'TEST1.FOR' und 'TEST2.FOR'. Dann sähe die Darstellung der Tabelle in TURBO-PASCAL 5.0 so aus:

```
CHANGE_NAMEN[1]:='MAIN.FOR';
CHANGE_NAMEN[2]:='TEST1.FOR';
CHANGE_NAMEN[3]:='TEST2.FOR';
CHANGE_DIR:= 3;
```

4.2.3 Tabellenprogrammierung

Diese Technik ermöglicht es, ein Programm für verschiedene generelle Änderungen verwenden zu können, ohne es selbst zu verändern; sondern es werden nur die entsprechenden Einträge in der Tabelle zu dem Programm (in einer Datei) mit einem Textsystem durchgeführt; so braucht nichts wiederholt übersetzt (compiliert) zu werden. Der eigentliche Programmablauf steht damit in einer Tabelle und nicht eigentlich im Programm selbst, welches nur mehr die Möglichkeit beinhaltet, eine bestimmte Tabelle abzuarbeiten. Aus Änderungen im Programmtext werden so Änderungen in einer Tabelle.

Dies ist besonders bei großen Mehrplatzsystemen von Vorteil, wo Programme aus ökonomischen Gründen nicht einfach vorübergehend zwecks Neuinstallation gesperrt werden können. Hier werden somit nur Tabellen ausgetauscht, was nicht nur in der Betriebszeit geschehen kann, sondern sogar während das jeweilige Programm in aktiver Verwendung steht !

Wenn viele Anwender mit demselben Programm arbeiten, müßten vor einer Neuinstallation normalerweise alle ihre momentane Arbeit beenden und danach wieder aufnehmen. Das Programm kann ebenso wie die gewohnten Kommandos, die das Programm starten, normalerweise nicht ausgetauscht werden, solange auch nur ein einziger Anwender damit arbeitet. Außerdem verursachen gemeinsam benützte Datenbanksysteme und Netzwerkverbindungen zu anderen Computeranlagen bei einer Neuinstallation während der Betriebszeit große Schwierigkeiten. An dieser Stelle sei noch ein Gedanke vorweggenommen: Es ist klar, daß bei konsequenter Anwendung von Tabellen dort nicht mehr einfache Einträge im Sinne von Zeilen und Spalten genügen werden, sondern es werden Kennworte mit Parametern zur Verwendung kommen. So kommt man zwangsläufig zur Definition von eigenen Makro-Sprachen für jede Tabelle, welche aus Kennworten (Befehlen) und verschiedenen Parametern besteht und letztlich vom Programm interpretierend gelesen wird.

Der Gedanke sei nun vorgreifend weitergesponnen: Wenn der eigentliche Programmablauf in Tabellen steht und ein Programm diese interpretiert, kann doch dasselbe Programm während des Ablaufes diese Tabelle verändern. Damit aber verändert ein Programm während der Durchführung seinen Ablauf und damit sich selbst! Was wir dann vor uns haben, ist eine Variante von einem lernenden System.

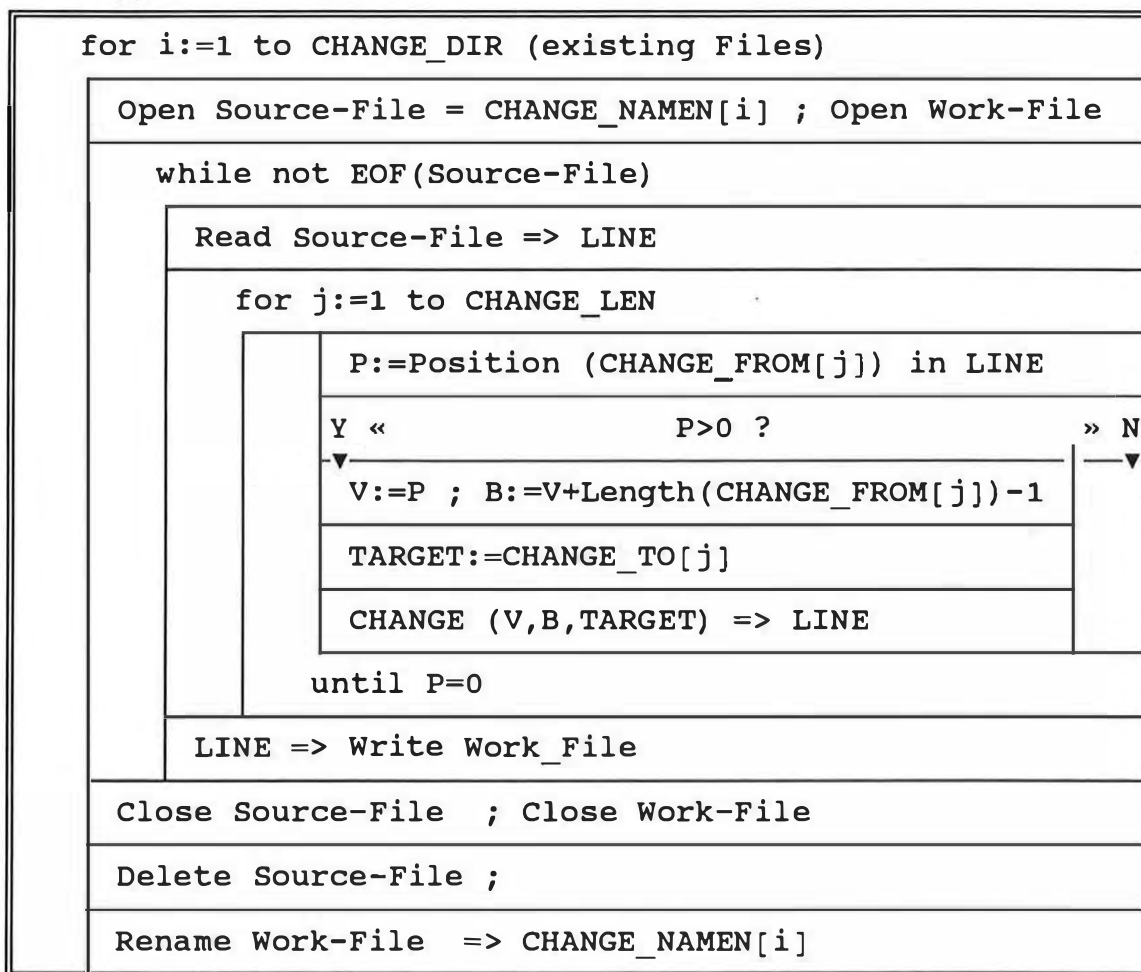
4.2.4 Prozedur CHANGE_ALL

4.2.4.1 Einführung

Doch zurück zu unserem einfachen Algorithmus : Ich möchte jetzt vorführen, wie in einer Gruppe von Files systematisch Änderungen durchgeführt werden können; und zwar wieder formal in der Form der Programmiersprache TURBO-PASCAL 5.0:

4.2.4.2 Struktogramm von CHANGE_ALL

CHANGE_ALL



4.2.4.3 Source-Code von CHANGE_ALL

```

{
{ CHANGE_ALL : Ändert die Files in Tabelle CHANGE_FILES
{           (Spalte CHANGE_NAMEN) laut der Tabelle
{           CHANGE_TAB (Spalten CHANGE_FROM und
{           CHANGE_TO); Anzahl Files sei CHANGE_DIR,
{           Anzahl Zeilen in _TAB sei CHANGE_LEN.
{
}
}
procedure CHANGE_ALL; const Work_File_Name = 'CHANGE.WRK';

var
    i,j,p,v,b : integer;
    Fil,Wrk    : text;
    Line       : string 255 ;

begin
{
{ Beginn der File-Schleife
{
for i:=1 to CHANGE_DIR do
begin
{
{ Files öffnen
{
assign(Fil,CHANGE_NAMEN[i]);reset (Fil);
assign(Wrk,Work_File_Name );rewrite(Wrk);
while not eof(Fil) do
begin
{
{ Nächste Zeile lesen
{
readln(Fil,Line);
{
{ Schleife der Änderungen
{
for j:=1 to CHANGE_LEN do
begin
repeat
p:=pos(CHANGE_FROM[j],Line);
if p>0 then
begin
v:=p;b:=v+length(CHANGE_FROM[j])-1;
Target:=CHANGE_TO[j];
CHANGE (Line,v,b,Target);
end;
until p=0;
end;
{
{ Nächste Zeile schreiben
{
writeln(Wrk,Line);
{
{ Files schließen
{
end;
close(Fil); close(Wrk); Del(CHANGE_NAMEN[i]);
Rename(Work_File_Name,CHANGE_NAMEN[i]);
end;
end;

```

4.2.5 Text-Veränderungen

Obwohl der vorliegende Algorithmus so einfach ist, läßt sich doch schon einiges ablesen : Es handelt sich um ein Werkzeug, welches Gruppen von Files systematisch verändern kann, und zwar nur hinsichtlich bestimmter Zeichenkombinationen und nicht z.B. von Worten. Es ist aber auch ein sehr primitives Mittel, Umstellungen zu bewerkstelligen, und hat im Prinzip nur die Intelligenz eines durchschnittlichen Textsystems (z.B. QUAD unter MPE).

Eines der Probleme dieser Art der Änderung ist folgende: Alle Änderungen betreffen den ganzen Text ; unabhängig von Prozedur-Ebenen und ohne Rücksicht darauf, ob die Änderung im Programmtext selbst, in Text-Konstanten, in Kommentaren oder in Compiler-Befehlen durchgeführt wird.

4.2.6 Hierarchie in Tabellen

Das zweite Problem liegt in der versteckten Hierarchie der CHANGE_TAB-Tabelle. Zum ersten werden je Zeile im File des ursprünglichen Programm-Textes Zeile für Zeile der Änderungstabelle durchgearbeitet; diese Zeilen werden außerdem solange wiederholt angesprochen, bis keine Änderung mehr durchgeführt wird. Dies habe ich deshalb so strukturiert, da ich annehme, daß die Files der Programme im allgemeinen größer sind als die Tabellen für die Änderungen und so die Anzahl der Input/Output-Operationen minimiert werden kann, wenn alle Files und Tabellen auf externen Massenspeichereinheiten installiert sind.

Diese Hierarchie bedingt, daß die Reihenfolge der Einträge in der Änderungstabelle sehr wesentlich wird. Bei ihrem Entwurf darf als jeweiliger status quo nicht der ursprüngliche Text genommen werden, sondern jener Text mit den Änderungen der bisher abgearbeiteten Einträge der Änderungstabelle.

Beispiel: Im ursprünglichen System sei das Zeichen " der Begrenzer für Text-Konstanten, im Ziel-System das Zeichen '. Die Tabelle könnte so definiert sein (TURBO-PASCAL 5.0) :

```
CHANGE_FROM 1 :=#39; CHANGE_TO 1 :=#96;  
CHANGE_FROM 2 :=#34; CHANGE_TO 2 :=#39;  
CHANGE_LEN:=2;
```

Die Notation #34 stellt das Zeichen ", #39 das Zeichen ' und #96 das Zeichen ` dar. Diese Darstellung wurde deshalb gewählt, da in PASCAL selbst der Begrenzer von Texten das Zeichen ' ist und deshalb nicht einfach auch innerhalb von Text-Konstanten verwendet werden kann (das wäre ''). Der Einheitlichkeit halber wurde dieselbe Darstellung bei den anderen Zeichen auch verwendet.

Daher tritt das Problem der Textbegrenzer konsequenterweise auch in Tabellen und Makro-Sprachen auf. Man kann es lösen, indem man einen Befehl erfindet, der das Zeichen für die Begrenzung von Text-Konstanten dynamisch umdefiniert. Als solches Zeichen verwendet man dann sinnvollerweise eines, welches innerhalb des Textes sicher nicht vorkommt.

In obigem Beispiel würde "Login:" in 'Login:' umgewandelt werden; das ist trivial. Weiter würde "Bitte geben Sie 'RETURN' ein : " in 'Bitte geben Sie `RETURN` ein :' richtig umgewandelt werden. Hier sieht man deutlich die Wichtigkeit der Reihenfolge der Tabelleneinträge - ja erst Kombinationen von Einträgen ergeben eine sinnvolle Änderung. Bei der umgekehrten Reihenfolge der Einträge würde folgendes herauskommen : `Bitte geben Sie `RETURN` ein :`. Das wäre aber sicher nicht das beabsichtigte Ergebnis.

4.2.7 Rekursion in Tabellen

Das zweite Problem bei Tabellen dieser Art ist die Rekursion, die innerhalb eines Eintrages und auch zwischen Einträgen eintreten kann und fast immer zu einem nicht oder nur von außen zu erzwingenden Ende kommt oder wenigstens zu irgend einer Art von Überlauf führt.

Beispiel: Der einfachste Fall eines einzelnen rekursiven Tabelleneintrages könnte so definiert sein (TURBO-PASCAL 5.0) :

```
CHANGE_FROM 1 := '+'; CHANGE_TO 1 := '++';  
CHANGE_LEN:=1;
```

Enthält eine Zeile ein '+', so explodiert sie sozusagen; ohne auf andere Einträge einzugehen, wird die Zeile Schritt für Schritt größer, bis die Länge der Zeile das erlaubte Maximum überschreitet und der Algorithmus 'abstürzt' wenn CHANGE verwendet wird. Es gibt natürlich auch kompliziertere Formen dieser Art von Rekursion.

Diese Effekte können natürlich ganz einfach vermieden werden, indem wir den Algorithmus CHANGE etwas verfeinern: Solange wir in einer Zeile der Tabelle verweilen, merken wir uns eine Position Q, ab der in der Zeile ein Eintrag aus CHANGE_FROM gefunden werden darf. Zu Beginn sei sie 1, dann ist sie jeweils gleich B+length(Target) der letzten Ersetzung:

```
{
{   Schleife der Änderungen
{
for j:=1 to CHANGE_LEN do
begin
  Q:=1;           { Start für Suchen initialisieren }
  repeat
    p:=pos(CHANGE_FROM j ,Line,Q); { ab Pos. Q }
    if p>0 then
      begin
        v:=p;
        b:=v+length(CHANGE_FROM j )-1;
        Target:=CHANGE_TO j ;
        CHANGE (Line,v,b,Target);
        Q:=v+length(Target);           { Q erhöhen }
      end;
    until p=0;
  end;
end;
```



4.2.8 Weitere Möglichkeiten

Es ist klar, daß für den praktischen Einsatz eines Algorithmus wie dem oben beschriebenen CHANGE_ALL noch einige Möglichkeiten hinzukommen müßten, von denen ich nur die zwei erwähnen möchte, die in einfachen Textsystemen (z.B. WordStar, Turbo_Edit,...) enthalten sind:

- 1) Beim Start die Möglichkeit zu entscheiden, ob jede einzelne Änderung vom Benutzer jeweils bestätigt werden soll. In dem Algorithmus CHANGE_ALL müßte eine solche Bestätigung so erfolgen, daß vorerst CHANGE aufgerufen wird, aber mit einer Hilfs-Variablen, auf die zuvor die aktuelle Zeile gespeichert wurde; dann müßten im Falle der Notwendigkeit einer Bestätigung beide Versionen am Bildschirm angezeigt und erst nach positiver Antwort des Benutzers dürfte das Ergebnis auf die ursprüngliche Zeile abgebildet werden:

```
Line2:=Line;
CHANGE(Line2,v,b,Target);
if Bestaetigung_holen then
begin
  writeln;
  writeln(Line);
  writeln(Line2);
  write('OK (J/N) ');
  readln(CH);
  if CH<>'J' then Line2:=Line;
end;
Line:=Line2;
```

- 2) Ebenso sollte beim Start entschieden werden können, ob bei der Suche nach dem Tabelleneintrag in der aktuellen Zeile zwischen Groß- bzw. Kleinschreibung unterschieden werden soll. Das löst man am einfachsten, indem man die Tabelleneinträge einerseits und vor Aufruf von POS die Zeile andererseits so umwandelt, daß eventuell auftretende Kleinbuchstaben in Großbuchstaben umgewandelt werden, falls dies gewünscht wird:

```
begin
  if Nur_Grossbuchstaben then
    begin
      for i:=1 to CHANGE_LEN do
        begin
          Line:=CHANGE_FROM[i];
          for j:=1 to length(Line) do
            Line[j]:=upcase(Line[j]);
          CHANGE_FROM[i]:=Line;
          Line:=CHANGE_TO[i];
          for j:=1 to length(Line) do
            Line[j]:=upcase(Line[j]);
          CHANGE_TO[i]:=Line;
        end;
      end;
      
      Line1:=Line;
      if Nur_Grossbuchstaben then
        for j:=Q to length(Line1) do
          Line1[j]:=upcase(Line1[j]);
        p:=pos(CHANGE_FROM[j],Line1,Q);

```

5 Maskenorientierte Transformationen

5.1 Einführung

Die logisch nächsthöhere Stufe einer Transformation von Programmen ist die maskenorientierte Transformation, die als Erweiterung der zeichenorientierten Transformation zu sehen ist und die bereits von einfachen Textsystemen beherrscht wird.

Beispiel : Ersetze '(*'..'*)' durch '{'..'}' in der Datei.

In Worten ausgedrückt: Suche in der Datei jeweils nach einer öffnenden runden Klammer und einem Stern, wobei dem Stern danach noch ein zweiter Stern gefolgt von einer schließenden runden Klammer folgen muß, wobei zwischen den beiden Sternen irgendetwas stehen kann. Wenn eine solche Maske gefunden wird, ersetze jeweils '(' durch '{' und '*' durch '}' und verändere die Zeichen dazwischen nicht.

Obige Regel angewendet auf die Zeile

Dies ist ein (* Kommentar *) !

ergibt

Dies ist ein { Kommentar } !

Im Prinzip wird dabei eine Maske definiert, die aus Zeichen besteht und die sozusagen auf eine Datei gelegt und verschoben wird, bis sie das erste Mal paßt. Darauf wird der unter der Maske liegende Teil der Datei herausgenommen und durch die Ziel-Maske ersetzt.

Wie bei der zeichenorientierten Transformation wird man bei der maskenorientierten Transformation zweckmäßigerweise von Zeilenstrukturen ausgehen. Gehen Masken über mehrere Zeilen, so ist eine Verkettung der Zeilen, Anwendung der Transformation und anschließende gemeinsame Ausgabe der geänderten Zeilen auf die Datei anzuraten.

Die Ausweitung der Masken über mehrere Zeilen führt automatisch zu den strukturorientierten Transformationen (siehe später), wenn man Zeilen als eine besonders einfache Art einer Struktur betrachtet.

5.2 Konstante und Variable

Eine Maske besteht grundsätzlich aus zwei Arten von Elementen : Konstanten und Variablen. Die Variablen werden vor dem Herausnehmen der Quell-Maske aus der Datei definiert und beim Ersetzen der Maske verwendet.

Ein einfaches Beispiel soll nun diesen Sachverhalt erläutern: Nehmen wir an, es ginge darum, Teile aus Zeichenketten in zwei verschiedenen Programmiersprachen zu definieren und eine in die andere überzuführen. Nehmen wir weiter an, daß diese Zeichenkette den Namen LINE bekäme und als Inhalt den Text 'Dies ist ein Test' aufzuweisen habe. Wenn man nun aus dieser Zeichenkette das 'ist' herausnehmen will und diesen Vorgang allgemein formulieren soll, so kann das mit der Programmiersprache FORTRAN 66 folgendermaßen geschehen : LINE[6:3]. Dies bedeutet, daß aus der Variablen LINE ab dem 6-ten Zeichen 3 aufeinanderfolgende Zeichen herausgenommen werden sollen, die dann als neue Zeichenkette das Ergebnis der beschriebenen Funktion darstellen.

Dasselbe Beispiel in der Programmiersprache FORTRAN 77 formuliert sieht folgendermaßen aus : LINE(6:8). Hier kann auch schon der Unterschied festgestellt werden. Bei FORTRAN 66 kennzeichnen eckige Klammern den Teil einer Zeichenkette, gefolgt von der Startposition, einem Doppelpunkt und der Anzahl der herauszunehmenden Zeichen innerhalb des Teiles der Zeichenkette. Hingegen in FORTRAN 77 sind runde Klammern vorgeschrieben, gefolgt ebenfalls von der Startposition, von einem Doppelpunkt und der Endposition des Teiles innerhalb der ursprünglichen Zeichenkette.

Nun wird versucht, das eben mit Worten beschriebene Beispiel mit einer Transformationsdefinition mit Hilfe von zeichenorientierten Masken allgemein zu veranschaulichen:

```
"['A':'B']" => " ('A':'A'+ 'B'-1) "
```

Das doppelte Anführungszeichen soll in unserem Beispiel je eine Definition begrenzen. Die Quell-Definition soll von der Ziel-Definition durch einen Implikationspfeil getrennt werden.

Innerhalb der einzelnen Masken seien die Konstanten mit einfachen Apostrophen begrenzt; die Namen dazwischen seien die Namen der Variablen.

Obiges Beispiel könnte so gelesen werden: Eine öffnende eckige Klammer gefolgt von irgendetwas, dem wir den Namen A geben, ein Doppelpunkt, irgendetwas anderes, dem wir den Namen B geben, und eine schließende eckige Klammer sollen durch folgendes ersetzt werden: Vorerst eine runde Klammer, dann der Inhalt der Variablen A - das, was in der Quelle zwischen eckiger Klammer und Doppelpunkt steht - , dann ein Doppelpunkt, nochmals der Inhalt der Variablen A, gefolgt von einem '-', dem Inhalt der Variablen B sowie dem '+1' und schließlich die schließende runde Klammer.

Somit beinhalten Masken zwei verschiedene Elemente: Konstante Texte und Variablen, denen ein Textteil dann zugewiesen wird, wenn sie das erste Mal auftauchen.

6 Strukturorientierte Transformationen

6.1 Einführung

Der letzte Schritt, die qualitativ vorläufig höchste Stufe einer Umformung ist schließlich eine strukturorientierte Transformation von Programmen oder auch die strukturorientierte Normierung von den verschiedenen geodätischen Registrierformaten. Dies bedeutet, daß nicht mehr nur einzelne Bits oder Zeichen oder Masken aus Zeichen geändert werden, sondern daß Strukturen geändert werden, die mit Zeilen nichts mehr zu tun haben müssen.

Beispiel : PROGRAM TEST
 WRITE(*,*) 'Dies ist ein Test'
 END

Hier haben wir ein kurzes Programm mit dem Namen TEST, geschrieben in der Programmiersprache FORTRAN 77. Es gibt am Bildschirm die Zeile Dies ist ein Test aus. Es sei WRITE(*,*) durch DISPLAY auszutauschen.

Lexikalische Analyse:

Symbol-Nummer	Symbol	Symbol-Typ
1	PROGRAM	Bezeichner
2	TEST	Bezeichner
2a	<CR>	Zeilenschaltung
3	WRITE	Bezeichner
4	(Klammer auf
5	*	Operator
6	,	Trennzeichen
7	*	Operator
8)	Klammer zu
9	'Dies ist ein Test'	Text
9a	<CR>	Zeilenschaltung
10	END	Bezeichner

Strukturelle Analyse:

Symbol- Nummer	Symbol/Struktur
1	PROGRAM
2	TEST
2a	<CR>
3	WRITE _____
4	(_____
5	* _____ Einheit
6	,
7	* _____ Format
8) _____
9	'Dies ist ein Test'-Text _____
9a	<CR>
10	END

Änderung der Strukturen:

Symbol- Nummer	Symbol/Struktur
1	PROGRAM
2	TEST
2a	<CR>
3	DISPLAY _____
9	'Dies ist ein Test'-Text _____
9a	<CR>
10	END

Ergebnis ausgeben:

```
PROGRAM TEST
DISPLAY 'Dies ist ein Test'
END
```

6.2 Lexikalische Analyse

Der erste Schritt einer derartigen Transformation ist die lexikalische Analyse. Hier wird aus einer Datei - einem Strom von Zeichen - eine Kette von Worten (Symbolen) mit Attributen. Damit können später Strukturen aus Symbolen erkannt werden. Diese Strukturen definieren den Zusammenhang der Symbole untereinander und ihre Änderung ist der Kern einer strukturorientierten Transformation.

Dieser erste Schritt ist nicht so einfach. Das liegt daran, daß die Worte nicht immer auf die gleiche Weise aus einer Zeichenkette erzeugt werden können. Die Regeln für diesen Vorgang variieren von Problem zu Problem. Es wird daher notwendig, eine eigene Sprache zu entwickeln, die diese Regeln beschreibt, oder mit anderen Worten, eine Tabelle, in der festgeschrieben wird, was im jeweiligen Fall ein Wort überhaupt ist.

Man erwartet beispielsweise, daß solche Regeln verallgemeinert werden und somit für alle oder zumindest für die meisten Transformationen konstant bleiben. Ein beliebtes Beispiel ist das Erkennen eines Bezeichners (Symbolen). Für unser Beispiel soll er aus 1 bis 8 Buchstaben und Ziffern bestehen können, wobei das erste Zeichen ein Buchstabe sein muß. Unerheblich ist hier, ob zwischen Groß- und Kleinschreibung unterschieden wird oder nicht.

Dieser Bezeichner scheint leicht aus einem Strom von Zeichen isoliert werden zu können. Ein Zeilenanfang oder ein Zeichen, welches kein Buchstabe ist, begrenzt automatisch jeden solchen Bezeichner sozusagen von links. Ebenso wird jeder Bezeichner rechts durch ein Zeichen, welches weder ein Buchstabe noch eine Ziffer ist, beziehungsweise durch ein Zeilenende begrenzt.

Das einfachste Gegenbeispiel wäre hier die Anwendung in der Programmiersprache FORTRAN. Hier sind Leerzeichen nicht signifikant. Das bedeutet, daß beispielsweise eine Schleife normalerweise als "DO 1 I=1,2,1" angeschrieben wird. Es werden alle Befehle zwischen diesem Befehl und der Marke (dem Label) mit der Nummer 1 solange wiederholt, bis die Variable I den Wert 2 erreicht hat. Am Beginn wird I gleich 1 gesetzt und nach jedem Durchgang um jeweils 1 erhöht. Denselben Befehl könnte man mit "DO1I=1,2,1" korrekt anschreiben. Hier würde unsere vorhin definierte Regel bereits kläglich versagen, da sie einen Bezeichner "DO1I" erkennen müßte, was natürlich in diesem Fall völlig sinnlos ist.

Ein zweites Gegenbeispiel und ebenfalls als typisches Problem bei Umstellungen sei wieder in FORTRAN die fehlende Signifikanz von Leerzeichen genannt. Als spezielle Anwendung sei nun nicht die fehlenden Leerzeichen außerhalb von Bezeichnern aufgeführt, sondern umgekehrt überzählige Leerzeichen in Bezeichnern. Zum Beispiel könnte man ein Unterprogramm in FORTRAN so beginnen : "SUBROUTINE GETNEW P" u.s.w. ... Da aber Leerzeichen nicht signifikant sind, wird oftmals der besseren Lesbarkeit wegen geschrieben: "SUBROUTINE GET NEW P" u.s.w. Obige Regel würde als Bezeichner des Namens des Unterprogrammes lediglich GET erkennen können.

Natürlich kann man Leerzeichen ignorieren. Dies wird schließlich von den FORTRAN-Übersetzungsprogrammen ja auch getan. Allerdings wäre es dann keine allgemeine Beschreibung einer lexikalischen Analyse mehr, da ja die reservierten Worte sowie ihre semantische Bedeutung als Spezialfälle programmiert werden mußten ! Dann könnte man wohl nicht mehr von einer allgemeinen Lösung sprechen, wenn man für jede Sprache irgendetwas Spezialanwendungen vorsehen müßte.

Ein zweiter sehr wichtiger Aspekt sei hier vorweggenommen: Wenn für das Erkennen eines Wortes im Zeichenstrom sein Zusammenhang mit bestimmten reservierten Worten oder/und mit benachbarten Worten für die lexikalische Analyse wesentlich wird, würde es schwer sein, immer auch unvollständige Programm-Bäume aufbauen zu können. Diese Möglichkeit, auch nur Teile von Strukturen in einen Strom von Symbolen verändern zu können, erscheint mir die Basis für eine effiziente Umstellung von Programmen zu sein.

Des Weiteren sei an dieser Stelle daran erinnert, daß es in dieser Arbeit ja auch um eine Normierung von Daten von Registriereinheiten von geodätischen Instrumenten geht. Dort können noch viel weniger Verallgemeinerungen bezüglich einer lexikalischen Analyse getroffen werden, da davon ausgegangen wird, daß die betroffenen Formate nur internen Zwecken dienen, normalerweise dem Geodäten verborgen bleiben, daher meist maschinennahe programmiert sind und somit am wenigsten irgendeiner Sprache ähnlich sind.

6.3 Strukturelle Analyse

Der zweite Schritt der Transformation ist die Struktur-Analyse. Dabei wird die Struktur des Stromes von Symbolen untersucht - aber nur nach den Kriterien, die für die jeweilige Umstellung interessant sind. Aus diesen Untersuchungen heraus wird schließlich ein unvollständiger Programmbaum von unten nach oben (Bottom-Up) erzeugt und wie die Symbole gespeichert. Manche Übersetzungsprogramme gehen von oben noch unten vor (Top-Down) - dann ist die erste Struktur das Programm als ganzes. Im folgenden wird immer von einer Bottom-Up-Analyse ausgegangen, da damit das Erstellen unvollständiger Programm-Bäume wesentlich leichter ist.

Bespielsweise könnte die Aufgabe darin bestehen, eckige Klammern in runde Klammern umzuwandeln und dabei anzuzeigen, welche öffnenden mit welchen schließenden Klammern korrespondieren:

Beispiel : $A := B[C[2], 3] + D[1]$

Symbol- Nummer	Symbol/Struktur
1	A
2	:=
3	B
4	[
5	C
6	[
7	2
8]
9	,
10	3
11]
12	+
13	D
14	[
15	1
16]

Wie man sieht, entstehen dabei nur einzelne Elemente eines Programm-Baumes, die jeweils die zunächst inneren beinhalten. Die Strukturen, die sozusagen in derselben Ebene liegen, berühren einander überhaupt nicht. Der andere Extremfall wäre der, daß alle Strukturen einer Quell-Datei in Form eines Symbol-Stromes umzuwandeln wären. Damit würde zwangsläufig ein vollständiger Programm-Baum entstehen.

Natürlich ist es notwendig, ebenso wie für die lexikalische Analyse eine allgemeine Sprache zu entwickeln, die die aufzubauenden Strukturen definiert. Diese Sprache muß unabhängig sein von der Art der verwendeten Programmiersprachen und möglichst weitgehend offen für zukünftige Entwicklungen. Nur so wird man eine Struktur allgemein beschreiben können.

6.4 Programmbaum ändern

Der dritte logische Schritt ist die systematische Veränderung des gespeicherten Programm-Baumes, sei er nun unvollständig oder nicht. Hier werden die gespeicherten Wurzeln und Äste nach festgelegten Regeln verändert. Dies bewirkt nicht nur eine Veränderung jeweils der Reihenfolge der Äste, sondern folgerichtig auch Veränderungen bis zu den Wurzeln.

6.5 Ergebnis ausgeben

Der letzte Schritt ist die Ausgabe des veränderten Programm-Baumes mit den Symbolen an den Enden der Wurzeln. Dabei muß man vorsichtig vorgehen. Da zwar die gespeicherten Strukturen verändert wurden, aber nicht der Ort, an dem sie gespeichert waren, besteht kein unmittelbar ablesbarer Zusammenhang mehr zwischen den Wurzeln des Programmbaumes und der auszugebenden Datei.

Somit muß der Programm-Baum vor dem Auslesen der Symbole entsprechend der neuen Ordnung umsortiert werden, sodaß die Symbole einfach nacheinander ausgegeben werden können, wie sie hintereinander in den Wurzeln des Programm-Baumes stehen. Würde man das nicht so machen, müßte für jedes Symbol seine neue Position in der Ziel-Datei berechnet werden.

7 Existierende Werkzeuge

7.1 BNF

Das 'Vokabular' einer Programmiersprache ist jene Menge an Zeichen und Wörtern, woraus Programme konstruiert werden. Die Elemente des Vokabulars einer Sprache werden Symbole (tokens) genannt. Diese Symbole werden üblicherweise in Klassen (classes) entsprechend ihrer Rolle in der Programmiersprache unterteilt.

Zum Beispiel gibt es in PASCAL die Klasse der Bezeichner (identifier); Zu diesen gehören 'writeln', 'Y', 'pos', u.s.w. Die Menge der Symbole einer Programmiersprache wird so definiert, daß es einerseits der Programmierer leicht hat, sie sich zu merken und sie zu überblicken und daß andererseits ein Übersetzungsprogramm (compiler) in der Lage ist, diese Symbole effizient zu erkennen und zu interpretieren. Den entsprechenden Modul des Übersetzungsprogrammes nennt man Scanner und den Vorgang nennt man lexikalische Analyse.

Bei einer rationellen Analyse sollte das Übersetzungsprogramm mit einem einzigen Lauf (pass) einen Programm-Text in die entsprechenden Symbole zerlegen und diese klassifizieren können.

Einfach ist dies noch bei einfachen Symbolen wie '+'. Schwieriger wird die Beschreibung der Klassen bei Bezeichnern, Konstanten u.s.w., da diese komplizierter definiert werden. Eine Möglichkeit, Klassen von Symbolen einer Programmiersprache zu beschreiben, ist eine Meta-Sprache, zum Beispiel eine kontextfreie Grammatik.

Dies ist eine gebräuchliche Methode, die Syntax einer Programmiersprache zu beschreiben, und wurde ursprünglich von NOAM CHOMSKY zwischen 1950 und 1960 entwickelt, um die syntaktische Struktur von Englisch zu definieren. Die kontextfreie Grammatik wurde später durch BACKUS und NAUR adaptiert und 1962 zur Definition der syntaktischen Struktur der Programmiersprache ALGOL verwendet. Somit wurde diese adaptierte Form 'BACKUS-NAUR-FORM' oder kurz 'BNF' genannt ([6],[15],[16],[17],[21]). Seither ist die BNF oder eine ihrer zahlreichen Varianten ein Standard-Werkzeug, um die Syntax von verschiedenen Programmiersprachen zu definieren.

In der BNF werden die folgenden Meta-Symbole verwendet:

```
::= heißt 'ist definiert als'  
{x} heißt 'eine Folge von 0 oder mehr Elementen aus x'  
[x] heißt 'eine Folge von 0 oder einem Element aus x'  
| heißt 'oder (ausschließend)'
```

Innerhalb der BNF-Regeln können entweder Namen von Symbol-Klassen (z.B. Bezeichner) oder einfache Symbole vorkommen. Auf der linken Seite steht immer der Name einer Symbol-Klasse gefolgt von einem ::= . Rechts davon steht dann die Definition der Klasse.

Um den Gebrauch der BNF zu demonstrieren, wird nun als Beispiel eine Klasse von Symbolen, nämlich der Bezeichner, definiert. Dieser Bezeichner sei durch einen Buchstaben, gefolgt von einer beliebigen Anzahl von Buchstaben und/oder Ziffern definiert. So wären die folgenden Symbole gültige Bezeichner: TEST, A30, Alpha, Summe, u.s.w.

Um die Klasse der Bezeichner wie oben beschrieben definieren zu können, müssen wir vorerst eine Klasse BUCHSTABE und eine Klasse ZIFFER definieren:

```
BUCHSTABE ::= a | b | c | ... | z | A | B | C | ... | Z
```

```
ZIFFER ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Mit dieser Basis können wir nun die Klasse BEZEICHNER beschreiben:

```
BEZEICHNER ::= BUCHSTABE { BUCHSTABE | ZIFFER }
```

Doch nicht nur Symbol-Klassen können mit der BNF beschrieben werden - auch die Syntax einer Programmiersprache kann definiert werden. Als Beispiel sei ein einfacher arithmetischer Ausdruck genannt, der aus ganzen Zahlen, Bezeichnern sowie den Symbolen, Operatoren der Grundrechnungsarten bestehen kann:

```
BUCHSTABE ::= a | b | c | ... | z | A | B | C | ... | Z
```

```
ZIFFER ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

```
MULOP ::= * | /
```

```
ADDOP ::= + | -
```

```
ZAHL ::= { ZIFFER }
```

```
BEZEICHNER ::= BUCHSTABE { BUCHSTABE | ZIFFER }
```

```
WERT ::= ZAHL | BEZEICHNER
```

```
AUSDRUCK ::= WERT MULOP WERT | WERT ADDOP WERT | WERT
```

Mit diesen Mitteln kann auch die vollständige Syntax einer Programmiersprache beschrieben werden. Beispielsweise werden 108 BNF-Regeln benötigt, um die Syntax von PASCAL zu beschreiben.

7.2 Attributierte Grammatik

Die Erstellung von Übersetzungsprogrammen (compilers) wird durch Übersetzer-Generatoren erleichtert, welche den Programmierer bei der Erstellung von Übersetzungsprogrammen für beliebige Quell- und Zielsprachen von Routinearbeit befreien. Aus der eingegebenen formalen Definition der Übersetzung Quellsprache->Zielsprache erstellt der Übersetzer-Generator dann das entsprechende lauffähige Übersetzungsprogramm. Diese Definition wird dabei in einer Generatorsprache formuliert. Als Generatorsprache wird meistens eine attributierte Grammatik verwendet.

Die attributierte Grammatik ist eine Mischung aus Grammatik und Programmiersprache. Die Syntax der Quellsprache wird dabei durch eine Grammatik definiert. Die Übersetzung von der Quellsprache in die Zielsprache wird beschrieben, indem den Grammatiksymbolen Attribute zugeordnet und semantische Regeln für diese Attribute formuliert werden. Somit wird die Übersetzung durch die Berechnung der Attribute definiert.

Die Berechnung der Attribute ist an die jeweilige Grammatik und somit an die Syntax der Quellsprache gebunden. Daher wird diese Art der Übersetzung allgemein syntaxgesteuerte Übersetzung genannt.

Die semantischen Regeln werden in einer Semantiksprache formuliert, welche nicht allgemein festgelegt ist. Meist jedoch wird eine existierende Programmiersprache verwendet.

Besonders bekannte Übersetzer-Generatoren sind LEX und YACC (siehe später). Sie verwenden die Programmiersprache C als Semantiksprache.

Die attributierte Grammatik ist theoretischer Hintergrund zahlreicher praktischer Anwendungen. Die konkrete Schreibweise ist allerdings nicht festgelegt, und so findet man in der Literatur auch verschiedene Formen.

7.3 LEX

Im folgenden wird eine Programmiersprache zur Spezifikation von Scannern vorgestellt. Damit kann eine Grundlage für einen Modul definiert werden, welcher dann aus einem Strom von Zeichen eine Folge von Symbolen liefert.

Es handelt sich um ein spezielles Werkzeug für Computer mit dem Betriebssystem UNIX, welches breite Verwendung bei der Entwicklung von Scannern von zahlreichen Programmiersprachen gefunden hat. Das Werkzeug wird LEX-Compiler, die dazugehörige Eingabebeschriftung LEX-Sprache genannt ([6],[21]).

Bei der Anwendung wird zunächst eine Spezifikation eines Scanners nach den Regeln der LEX-Sprache erstellt. Dieser Text wird vom LEX-Compiler in einen Programmtext umgewandelt, der in der Programmiersprache C geschrieben ist. Dieser Text muß wiederum von einem (vorauszusetzenden) C-Compiler in ein ausführbares Programm übersetzt werden, welches erst Symbole im Strom von Zeichen erkennt.

Ein LEX-Programm besteht aus drei Teilen:

Deklarationen

%%

Übersetzungsregeln

%%

Hilfsprozeduren

Der erste Teil (Deklarationen) dient der Deklaration von Variablen, symbolischen Konstanten sowie (regulären) Definitionen.

Der zweite Teil (Übersetzungsregeln) besteht aus regulären Ausdrücken mit Aktionen in geschweiften Klammern dahinter. Diese Aktionen sind in der Programmiersprache C geschrieben.

Der dritte Teil (Hilfsprozeduren) schließlich enthält alle jene Prozeduren, die in den Aktionen angesprochen worden sind oder werden können und ist ebenfalls in der Programmiersprache C geschrieben.

Normalerweise arbeitet ein Scanner mit einem interpretierenden Programm (parser) zusammen bzw. ist ein Teil davon. Beim Start liest der Scanner solange Zeichen ein, bis dieses 'längste Präfix' mit einem der regulären Ausdrücke im zweiten Teil übereinstimmt. Dann wird die entsprechende Aktion durchgeführt. Gibt diese Aktion die Kontrolle nicht an den Parser weiter, so bedeutet dies, daß das jeweilige Symbol übersprungen wird. Anderenfalls übergibt der Scanner als einzigen Wert das Symbol an den Parser. Allerdings gibt es noch die Möglichkeit, dieses Symbol mit Attributen zu versehen, auf welche die eingefügten Module in der Programmiersprache C (über die Variable YYLVAL) zugreifen können.

Folgende Meta-Symbole werden von LEX verwendet:

```
%% trennt die Abschnitte des LEX-Programmes voneinander
\t ist das Symbol für ein Leer- oder ein Tabulatorzeichen
\n ist das Symbol für eine Zeilenschaltung
\. ist das Symbol für einen Dezimalpunkt
\ - ist das Symbol für ein Minus-Zeichen
. ist das Symbol für ein beliebiges Zeichen außer \n
- wird (Infix-Notation) für Bereiche verwendet
? heißt 'höchstens einmaliges Auftreten von'
* heißt 'beliebiges Auftreten von'
/ ist der Vorausschau-Operator (Lookahead-Operator)
[ ] schließen Zeichenklassen ein
{ } schließen verwendete Namen von Symbolklassen ein
( ) können zur Zusammenfassung von Teilen verwendet werden
```

Um den Gebrauch von LEX zu demonstrieren, wird nun wieder als Beispiel eine Klasse von Symbolen, nämlich der Bezeichner, definiert. Dieser Bezeichner sei durch einen Buchstaben, gefolgt von einer beliebigen Anzahl von Buchstaben und/oder Ziffern definiert. So wären die folgenden Symbole gültige Bezeichner: TEST, A30, Alpha, Summe, u.s.w.

Um die Klasse der Bezeichner wie oben beschrieben definieren zu können, müssen wir (analog dem vorigen Kapitel) vorerst eine Klasse BUCHSTABE und eine Klasse ZIFFER im ersten Abschnitt (Reguläre Definitionen) beschreiben:

```
BUCHSTABE      [A-Za-z]
```

```
ZIFFER        [0-9]
```

Mit dieser Basis können wir nun die Klasse BEZEICHNER beschreiben:

```
BEZEICHNER     {BUCHSTABE}({BUCHSTABE}|{ZIFFER})*
```

Im zweiten Abschnitt (Übersetzungsregeln) könnte dann folgendes stehen:

```
{BEZEICHNER}  {YYLVAL=install_Bez();return(1);}
```

Hier stehen auf der rechten Seite in den geschweiften Klammern Anweisungen, die in der Programmiersprache C geschrieben sind. Die Variable YYLVAL wird dabei mit dem Wert des Funktionsergebnisses der Funktion install_Bez belegt. Diese Funktion könnte zum Beispiel das Symbol jeweils speichern und als Funktionsergebnis die Nummer des Symbolen in der Symboltabelle liefern. Diese Nummer wäre dann in diesem Fall das Attribut. Als Symbolklasse würde in unserem Beispiel der Wert 1 geliefert. Die Funktion install_Bez muß im dritten Abschnitt des LEX-Programmes in Form einer Prozedur, die in der Programmiersprache C geschrieben ist, stehen.

Abschließend sei noch eine Spezialität von LEX beschrieben: Es handelt sich um den Lookahead-Operator /. Dieser dient dazu, daß ein bestimmtes Symbol nur dann als solches erkannt wird, wenn dahinter bestimmte Zeichen stehen. Sowohl die Zeichen im Symbol selbst als auch die, welche dahinter stehen müssen, sind in der LEX-Sprache beschrieben und werden durch den Lookahead-Operator getrennt.

Die praktische Anwendung ist beispielsweise die Lösung eines bekannten Problems mit FORTRAN, wo Leerzeichen außerhalb von Kommentaren und Formaten keinerlei Bedeutung haben (daher ignoriert werden).

Die Schleifensteuerung in FORTRAN sieht z.B. so aus:

```
DO 1 I=2,3
```

Das Symbol DO ist hier ein Schlüsselwort von FORTRAN. Diese Anweisung bedeutet: Initialisiere die Variable I mit dem Wert 2 und führe die bis zur Marke mit der Nummer 1 stehenden Anweisungen solange aus, bis der Wert der Variablen I größer oder gleich 3 wird, wobei der Wert von I jeweils am Ende jedes Durchganges um 1 erhöht wird.

Da aber Leerzeichen ignoriert werden, könnte man obige Anweisung auch so schreiben: DO1I=2,3 . Hier wird das Problem offensichtlich: Nur dann, wenn nach DO eine Zahl, dann ein Bezeichner, ein Gleichheitszeichen, eine weitere Zahl und dann ein Beistrich steht, handelt es sich um den Beginn einer Anweisung für eine Schleife und nur dann sind die beiden ersten Zeichen wirklich das Schlüsselwort DO. In allen anderen Fällen handelt es sich um einen Bezeichner mit dem Namen DO1I.

Das Problem kann in LEX wie folgt gelöst werden:

```
DO/({LETTER}|{DIGIT})*=({LETTER}|{DIGIT})*,{return(LOOPKEY)}
```

7.4 YACC

Im nun folgenden Abschnitt wird ein Parser-Generator beschrieben. Dabei handelt es sich um ein Werkzeug, welches es erleichtern soll, die Benutzeroberfläche (front-end) eines neu zu schaffenden Übersetzungsprogrammes zu entwickeln.

Der Name des Parser-Generators ist YACC (engl: yet another compiler-compiler / deutsch: noch ein weiteres Übersetzungsprogramm für Übersetzungsprogramme). Die erste Version von YACC wurde 1970 und in den folgenden Jahren von S. C. JOHNSON entwickelt. Es ist ein spezielles Werkzeug für Computer mit dem Betriebssystem UNIX, ist dort sehr verbreitet und hat bereits bei der Entwicklung zahlreicher Übersetzungsprogramme geholfen. Das Werkzeug nennen wir YACC-Compiler und die Eingabebeschriftung YACC-Sprache ([6],[21]).

Der Vorgang bei der praktischen Anwendung ist folgender: Vorerst wird die Benutzeroberfläche des zu entwickelnden Übersetzungsprogrammes definiert; d.h. die Regeln für die Transformation der dann zu lesenden Texte werden in der YACC-Sprache formuliert. Darauf wird der YACC-Compiler aufgerufen, der aus dem Text ein Programm macht, welches in der Programmiersprache C geschrieben ist. Dieses Programm muß dann noch von einem C-Compiler - welcher vorhanden sein muß - übersetzt werden. Das Resultat ist schließlich ein Modul, welches erst einen Strom von Zeichen einlesen, interpretieren und daraufhin weitere Aktionen durchführen kann.

Ein Programm-Text, welcher in der YACC-Sprache geschrieben ist, besteht aus drei Teilen (analog LEX - vergl. Kapitel 'LEX'):

```
Deklarationen
%%
Übersetzungsregeln
%%
unterstützende C-Module
```

Im ersten Abschnitt (Deklarationen) werden Variablen, symbolische Konstante und andere Deklarationen definiert. Dies muß nach der Syntax der Programmiersprache C und eingeschlossen in spezielle Klammern %{ und %} durchgeführt werden. Außerdem werden in diesem Abschnitt YACC-Grammatikzeichen definiert. Diese werden durch %TOKEN und dem entsprechenden Namen dahinter festgelegt (z.B.: %TOKEN ZIFFER). Diese Symbole können dann im zweiten und dritten Abschnitt verwendet werden.

Der Befehl `%LEFT '+' '-'` legt fest, daß die Operatoren + und - die gleiche Priorität haben und linksassoziativ (Auswertung von links nach rechts) sind. Beispiel: Beim Ausdruck `1+2-3` wird zuerst `1+2` und dann das Ergebnis-3 berechnet.

Der Befehl `%RIGHT '^'` legt fest, daß der Operator `^` rechtsassoziativ (Auswertung von rechts nach links) ist. Beispiel: Beim Ausdruck `2^3^4` wird zuerst `3^4` und dann erst `2^Ergebnis` berechnet.

Der Befehl `%NONASSOC '<'` schließlich verbietet die sequentielle Verwendung des Operators `<`. Zum Beispiel sind dann Ausdrücke wie `3<4<5` nicht erlaubt.

Der zweite Abschnitt (Übersetzungsregeln) besteht aus Regeln. Diese Regeln bestehen wiederum jeweils aus einer Definition der Grammatik (Grammatikproduktion) sowie der dazugehörigen Aktion. Diese Aktionen stehen in geschwungenen Klammern und sind in der Programmiersprache C geschrieben. Das Symbol `$$` steht dabei für den aktuellen Attribut-Wert, das Symbol `$1` für die Zeichen, die mit dem ersten Symbol der jeweiligen Definition verbunden sind, `$2` für die Zeichen, die mit dem zweiten Symbol verbunden sind u.s.w. Fehlt die Aktion, wird `{ $$ = $1 ; }` eingesetzt.

Damit können Strukturen und ihre Änderungen beschrieben werden. Typischerweise wird allerdings davon ausgegangen, daß diese Strukturen vollständig definiert werden.

Der dritte Abschnitt (unterstützende C-Prozeduren) schließlich enthält alle die Prozeduren, die in den Aktionen angesprochen werden. Diese müssen ebenfalls in der Programmiersprache C geschrieben sein. Dazu gehört auch ein Scanner, welcher den Namen `YYLEX` (siehe Kapitel 'LEX') haben muß.

Folgende Meta-Symbole werden von YACC verwendet:

```
%% trennt die Abschnitte des YACC-Programmes voneinander
% steht den YACC-Kommandos voran
: heißt 'besteht aus'
| ist das Symbol für das (ausschließende) ODER
; beendet jede YACC-Definition (Produktion)
$$ ist das Symbol für den aktuellen Attribut-Wert
$i steht für das i-te Symbol der Definition in der Quelle
```

Als Anwendungsbeispiel einer Transformation eines Textes in einen anderen Text mit YACC nehme ich zwei verschiedene Substring-Darstellungen (Teil einer Zeichen-Kette) von FORTRAN:

```
[<Startposition>:<Länge>] -> [<Startposition>:<Endposition>]
```

Wenn man nun ausschließlich den zweiten Abschnitt des dazugehörigen YACC-Programmes betrachtet, so müßte dort unter anderem stehen:

```
SUBSTRING : '[' AUSDRUCK ':' AUSDRUCK ']'
           { printf($1,$2,$3,$2,"+", $4,"-1", $5); } ;
```

7.5 SOURCE-TO-SOURCE TRANSLATION

Diese Technik ([3]) basiert auf der Anwendung von attribuierten Grammatiken. Wenn eine attribuierte Grammatik vorliegt, die eine Übersetzung von der Programmiersprache A in die Programmiersprache B beschreibt, so kann damit automatisch eine inverse attribuierte Grammatik gewonnen werden. Diese beschreibt dann die Übersetzung von der Programmiersprache B in die Programmiersprache A.

Um dies zu erreichen, wird eine neue Methode vorgestellt, um Übersetzungen mit einer Grammatik zu definieren, nämlich die 'Restricted Inverse Form Grammar (RIF)'. Dies ist eine Mischung aus einem syntax-gesteuerten Übersetzungs-Schema und einer attribuierten Grammatik. Dies bedeutet zwar prinzipiell eine Einschränkung der Möglichkeiten einer attribuierten Grammatik, andererseits wird in [3] gezeigt, daß RIFs beim Aufbau von inversen Grammatiken mächtiger sein können als attribuierte Grammatiken.

Es wird weiter davon ausgegangen, daß die (nur) auf diese Weise zu beschreibende Struktur des Quell-Programmes sowie die durchzuführenden Änderungen vollständig (!) definiert sein müssen.

Allerdings wird bei einer strengen Anwendung dieser Methode eine inverse Grammatik erzeugt, die besonders unübersichtlich (auch mehrdeutig) und ineffizient ist, selbst wenn die ursprüngliche Grammatik eindeutig, übersichtlich und effizient war.

Daher wird in der beschriebenen Arbeit hauptsächlich die Eignung und Effizienz von RIFs sowie der entsprechenden Inversions-Algorithmen untersucht. Das Ergebnis ist schließlich eine Reihe von Zusatzalgorithmen, welche die Grammatiken und ihre Inversen normalisieren und effizienter machen sollen.

Die Methoden wurden in Form eines Übersetzer-Generators realisiert, welcher nach den oben beschriebenen Definitionen zwei Übersetzer (einer für die Übersetzung A->B und einer für die Übersetzung B->A) erzeugt, und zwar als Programm-Texte, welche nochmals übersetzt werden müssen.

Eine Übersetzung von der Programmiersprache A in die Programmiersprache B (oder umgekehrt) erfolgt dann nicht direkt, sondern über eine kanonische Form zwischen Quellsprache und Zielsprache, welche für jede Problemstellung neu entworfen werden sollte, um die Effizienz zu erhöhen.

7.6 INVERSE TRANSLATION

Das Ziel der Arbeit [10] war, die Spezifikation von Übersetzern invertieren, um das automatische Programmieren zu erleichtern. Dabei wird von einer syntax-gesteuerten Übersetzung ausgegangen, welche wesentlich weniger Möglichkeiten bietet als etwa eine attribuierte Grammatik.

7.7 A TWO-WAY LANGUAGE INTERFACE

In [5] wurde eine Grammatik entworfen, welche sowohl englische (natürlichsprachliche) Sätze in die Elemente einer für den Computer leicht zu interpretierenden Kommandosprache übersetzt und umgekehrt.

7.8 CIRCULAR ATTRIBUTE GRAMMARS

Das Werk [8] enthält einen besonders interessanten Ansatz, Übersetzungen und die dazugehörigen Inversionen durchzuführen. Es wird die Möglichkeit untersucht, die Übersetzung in der Programmiersprache PROLOG zu beschreiben, da Prozeduren in Prolog leicht invertiert werden können. Leider kann nicht jede Übersetzung auf diese Weise definiert werden und außerdem muß immer ein PROLOG-Übersetzungsprogramm vorhanden sein.

7.9 VAX SCAN

VAX SCAN ([34]) ist ein Werkzeug, Programme zur Veränderung von Texten zu erzeugen. VAX SCAN ist auf Computern der Type VAX von DIGITAL EQUIPMENT mit dem Betriebssystem VMS verfügbar. Das Übersetzungsprogramm heißt SCAN-Übersetzer und die Eingabe wird in der SCAN-Sprache formuliert, welche eine eigenständige Programmiersprache ist. Damit ist VAX-SCAN ein maskenorientierter Übersetzer-Generator. Das Ergebnis ist ein lauffähiges Programm, welches einen Textstrom prüft (SCANnt) und dabei einen veränderten Textstrom ausgibt.

Die SCAN-Sprache enthält im wesentlichen folgende Elemente:

- 1) Definition von Symbolklassen, Variablen und Konstanten
- 2) Definition von Textmasken
- 3) Definition der Konstruktion des einzusetzenden Textes
- 4) Definition der Ausgabe der Dokumentation der Änderungen

Durch MACROS bzw. PROCEDURES können auch verschachtelte und komplizierte Strukturen relativ einfach behandelt werden. Innerhalb eines SCAN-Programmes kann außerdem auf Funktionen der Laufzeit-Bibliothek des VMS-Betriebssystems zugegriffen werden.

VAX SCAN kann als Filter verwendet werden, um aus einem Datenstrom bestimmte Elemente zu löschen. Ebenso kann man mit SCAN Ersetzungen in Programm-Texten durchführen, um verschiedene Dialekte einer Programmiersprache auszugleichen, einen Programm-Text zu analysieren oder den Programm-Text für die eigentliche Übersetzung aufzubereiten (precompiler).

7.10 Zusammenfassung

Die bestehenden Werkzeuge sind nicht sehr zahlreich (das bekannteste ist wohl LEX/YACC). Ein Grund dafür kann darin gesehen werden, daß Programmtexte üblicherweise in eine von einem Computer ausführbare Form und nicht in einen anderen Programmtext übersetzt werden sollen.

Nun werden zusammenfassend einige Gründe aufgeführt, warum die existierenden Werkzeuge nicht für die Transformation von geodätischen Programmen und die Normierung von Vermessungsdaten verwendet werden können:

- 1) Die bestehenden Werkzeuge sind konstruiert worden, um Programmtexte zu verändern und nicht, um Daten zu normieren. Daten repräsentieren keine auszuführenden Anweisungen, sondern allgemein Informationen.
- 2) Für den rationellen Einsatz muß ein einzelner Aufruf eines Übersetzungsprogrammes zur Durchführung von Änderungen genügen. Die bestehenden Werkzeuge sind hingegen meist Übersetzer-Generatoren, die eine weitere Übersetzung durch ein vorauszusetzendes Übersetzungsprogramm notwendig machen.
- 3) Ebenso muß ein allgemeines Werkzeug auch leicht auf anderen Systemen zu implementieren sein. Das wird von Übersetzer-Generatoren prinzipiell nicht erfüllt, da sie die Existenz eines bestimmten Übersetzungsprogrammes voraussetzen.
- 4) Die bestehenden Werkzeuge gehen meist davon aus, daß die Strukturen des Quell- sowie des Zieltextes vollständig bekannt sind. In der Praxis stellt sich hingegen oft die Aufgabe, nur bestimmte Elemente zu verändern und den Text dazwischen zu ignorieren.
- 5) Eine typische Schwäche ist die, daß meist davon ausgegangen wird, daß die Quell- und die Zielsprache gut strukturiert sind. Somit wird die Behandlung schlecht strukturierter Elemente einer Sprache nur sehr kompliziert oder gar unmöglich. Beispiele: GOSUB in BASIC, FORMAT in FORTRAN u.s.w.
- 6) Die Syntax für die Definition der lexikalischen Analyse, der Strukturen und ihrer Änderungen sollte gleich sein. Dies wird von den existierenden Werkzeugen nicht erfüllt.
- 7) Das Prinzip "Null-Eins-Unendlich" genügt in schlecht strukturierten geodätischen Programmen bzw. Daten nicht.
- 8) In Vermessungsprogrammen und besonders bei Vermessungsdaten ist es notwendig, für bestimmte Symbole zulässige Spaltenbereiche zu definieren.

- 9) Bei der Definition von Symbolklassen genügt es nicht, die zulässige Anzahl von Zeichen festzulegen, damit das jeweilige Symbol erkannt wird; vielmehr muß es möglich sein, eine weitere Bedingung einzuführen, damit in einer zweiten Stufe der Prüfung das erkannte Symbol einer speziellen Behandlung unterzogen werden kann, wenn die Bedingung nicht erfüllt ist.
- 10) Ebenso müssen auf einfache Weise numerische Symbolklassen definiert werden können, wo als zweite Bedingung ein zulässiger Wertebereich als Plausibilitätsprüfung des erkannten Wertes festgelegt wird.

8 TRANS - Kurzbeschreibung

Es wird nun ein Beispiel für eine maskenorientierte Transformation von Programmen oder Messungen präsentiert.

Es handelt sich dabei um eine primitive Programmiersprache mit dem Namen TRANS, die ich in einem kurzen, sehr einfachen Programm in der Programmiersprache MS-BASIC für die Personalcomputer von IBM unter dem Betriebssystem MS-DOS realisiert habe. Dieses Programm steht schon seit einiger Zeit im Einsatz und hat sich bereits als einfaches, aber effizientes Werkzeug bei der Übertragung von geodätischen Programmen bewährt. Damit soll weiter demonstriert werden, daß es auch mit sehr einfachen Mitteln möglich ist, taugliche Werkzeuge zur Umstellung von Programmen zu schaffen.

```
Beispiel : INPUT "TEST.FOR"
           OUTPUT "TEST.FTN"
           CS "TAN(_)" TO "DTAN(_)"
```

Dies ist ein besonders kurzes TRANS-Programm. Es liest eine Datei mit dem Namen TEST.FOR ein, ändert diese und schreibt das Ergebnis auf eine Datei mit dem Namen TEST.FTN. Die Regel für die Änderung steht in der dritten Zeile. Sie besagt: Suche in der ganzen Datei nach den 4 Zeichen 'TAN(', dem Zeichen ')' und dazwischen irgendwelche andere Zeichen. Ersetze dann jede so gefundene Maske durch die 5 Zeichen 'DTAN(', dem Zeichen ')' und setze dazwischen diejenigen Zeichen ein, die sich auch ursprünglich zwischen den runden Klammern befanden.

```
So würde aus der Zeile
TAN(3+0.1416)
das folgende entstehen:
DTAN(3+0.1416)
```

In der Anwendung existiert also ein interpretierendes Programm mit dem Namen TRANS. Dieses liest von einer einfachen Datei die auszuführenden Befehle, die in der Programmiersprache TRANS geschrieben sind, ein und verarbeitet sie. Diese Datei stellt somit eine Art Tabelle dar, in welcher nicht nur steht, welche Dateien geändert werden sollen und welche Dateien daraus entstehen sollen, sondern es stehen auch die Regeln für die Transformation selbst darin. Es werden daher alle Dateien, die umzuwandeln sind, Stück für Stück eingelesen, nach den in der Sprache TRANS formulierten Regeln umgewandelt und das Ergebnis wird schließlich jeweils auf eine neue Datei ausgegeben.

Die Entwicklung der Tabelle und somit des in TRANS geschriebenen Programmes kann in zwei Stufen erfolgen: Erst wird das Programm nach den in den jeweiligen Handbüchern enthaltenen Informationen geschrieben. Dann nimmt man eine der umzustellenden Dateien her, die typischerweise möglichst alle Aspekte der konkreten Umstellung enthält und somit geeignet ist, beispielhaft das TRANS-Programm zu überprüfen. Diese Datei wird nun transformiert und ihre Funktion im neuen System getestet - meist dadurch, daß der Programm-Text des neu entstandenen Programmes neu übersetzt (compiliert) wird. Dann werden eventuelle Fehler Schritt für Schritt beseitigt, bis dieser eine Modul halbwegs sicher richtig übersetzt wurde. Dann wird die Tabelle nicht mehr verändert, sondern es werden alle anderen Dateien nach demselben Programm umgeformt.

Es sei hier an dieser Stelle betont, daß diese Art der Automatisierung nur einen Teil der gesamten Umstellung von Programmen von einem System in ein anderes darstellen kann. Es darf nicht erwartet werden, daß ein so bearbeitetes Programm sofort eingesetzt werden kann. Es wird im allgemeinen selbstverständlich nachbearbeitet werden. Die Programmiersprache TRANS ist so wie alle in dieser Arbeit vorgestellten Methoden nur ein Werkzeug und muß als solches sinnvoll eingesetzt werden. Es muß in der Praxis immer ein Kompromiß gefunden werden, wieviel von den notwendigen Umstellungen automatisiert werden kann und welche speziellen Anwendungen dann einzeln vom Bearbeiter geändert werden müssen; denn je spezieller eine notwendige Änderung ist, desto weniger paßt sie in einen allgemein zu formulierenden Algorithmus und desto mehr Aufwand kostet ihre Implementierung in jeder Art von weitgehend automatisch arbeitenden Systemen.

Die detaillierte Definition der Programmiersprache TRANS liegt an der Technischen Universität Wien am Institut für Landesvermessung und Ingenieurgeodäsie, Abteilung Landesvermessung, auf.

9 TRAFO - Kurzbeschreibung

9.1 Einführung

In diesem Kapitel wird der Kern, die Essenz der Arbeit vorgestellt: Es handelt sich um ein Beispiel für eine strukturorientierte Transformation von Programmen oder Messungen in Form einer kontextfreien Programmiersprache mit dem Namen TRAFO.

Dabei wird vorausgesetzt, daß das Quell-Programm bzw. die ursprünglichen Messungen keine lexikalischen oder syntaktischen Fehler enthalten und daß eventuell vorkommende Operatoren linksassoziativ sind.

Diese Programmiersprache wurde mit Hilfe der Programmiersprache TURBO-PASCAL 5.0 für die Personal-Computer von IBM unter dem Betriebssystem MS-DOS realisiert.

Beispiel : IMPLICIT_WORDTYPE

```

INTEGER ::= [0..1] of ['+'..'-' ] +
           [1..9] of ['0'..'9' ] ;
ADD_CHAR ::= ['+' ] ;
SUB_CHAR ::= ['-'] ;
MUL_CHAR ::= ['*'] ;
DIV_CHAR ::= ['/'] ;
BRACKET1 ::= ['(' ] ;
BRACKET2 ::= [')' ] ;
ISDEF    ::= [':' ] + ['=' ] ;
IS       ::= ['=' ] ;
NAME     ::= [1..8] of ['A'..'Z', 'a'..'z' ] ;

```

IMPLICIT_TREE

```

EXPRESSION ::= <INTEGER> ;
EXPRESSION ::= <BRACKET1>+EXPRESSION+<BRACKET2>
EXPRESSION ::= EXPRESSION+<MUL_CHAR>+EXPRESSION
EXPRESSION ::= EXPRESSION+<DIV_CHAR>+EXPRESSION
EXPRESSION ::= EXPRESSION+<ADD_CHAR>+EXPRESSION
EXPRESSION ::= EXPRESSION+<SUB_CHAR>+EXPRESSION
EXPRESSION ::= EXPRESSION+EXPRESSION

```

```

DEF ::= 'LET'      = .LET.  +
        <NAME>     = .VAR.  +
        <IS>       = .IS.   +
        EXPRESSION = .EXPR. => .VAR.  &
                               ':'=' &
                               .EXPR. ;

```

In diesem Beispiel wird im ersten Abschnitt die lexikalische Analyse festgelegt. Demnach wird eine Symbol-Klasse INTEGER definiert, die ein Vorzeichen haben kann und aus 1 bis 9 Ziffern besteht. Danach werden den arithmetischen Operatoren Namen gegeben. Am Ende des ersten Abschnittes wird festgehalten, daß der NAME eines Bezeichners aus 1 bis 8 Groß- und/oder Kleinbuchstaben bestehen kann.

Im zweiten Abschnitt werden die Strukturen und ihre Änderungen definiert. Hier werden die Regeln von einfachen arithmetischen Ausdrücken festgelegt. Aus der Struktur DEF, die aus dem Symbol 'LET', einem Symbol des Typs NAME, einem Symbol des Typs IS und einem Ausdruck (EXPRESSION) besteht, soll folgendes gemacht werden: Zuerst der Bezeichner, der hinter dem 'LET' stand, dann das neue Symbol ':=' gefolgt von dem ursprünglichen Ausdruck.

So würde aus der Zeile
LET A=3+4*(5/8)
folgende Zeile gemacht werden:
A:=3+4*(5/8)

Die Programmiersprache TRAF0 enthält im wesentlichen drei Elemente :

- 1) Erstens kann die lexikalische Analyse der zu transformierenden Dateien definiert werden. Es wird damit festgelegt, nach welchen Regeln ein Strom von Zeichen in einen Strom von Symbolen umgewandelt werden soll.
- 2) Das zweite Element ist die Definition der strukturellen Analyse oder mit anderen Worten das Aufstellen von Regeln, nach denen die Programm-Bäume oder allgemein die Strukturen in den umzuformenden Dateien erkannt werden sollen.
- 3) Das dritte Element ist schließlich die Transformation selbst. Diese besteht wieder aus Regeln, nach welchen die gespeicherten Programm-Bäume systematisch umgeformt werden sollen.
- 4) Das vierte Element bleibt verborgen, da es automatisch durchgeführt werden kann; nämlich die Umwandlung der veränderten Strukturen in Ketten von Symbolen und von dort wieder in einen Strom von Zeichen, welcher die veränderte Datei ergibt.

Es existiert nun ein Programm, welches ebenfalls den Namen TRAF0 trägt und welches Transformationen von Programmen nach den in einer Tabelle stehenden Regeln durchführen kann. Diese Tabelle bleibt nach der Erzeugung immer dieselbe. Sie wird bei jedem Start neu interpretiert und ist andererseits nichts anderes als das vom Anwender in der Programmiersprache TRAF0 geschriebene Programm.

Der erste Schritt besteht am Anfang aus 3 lexikalischen Analysen, die mit demselben Programm durchgeführt werden:

- ▶ Beim ersten Mal wird (innerhalb des Programmes) festgelegt, wie die Symbole eines TRAF0-Programmes auszusehen haben,
- ▶ dann wird (als zweite Anwendung) die Tabelle (das TRAF0-Programm) nach den eben definierten Regeln interpretiert und die darin enthaltenen Regeln in dieselben Daten-Strukturen von TRAF0 gespeichert. Diese ersten beiden Anwendungen brauchen natürlich nur einmal durchgeführt werden.
- ▶ Die dritte Anwendung ist dann die lexikalische Analyse der umzustellenden Programme auf Grund der im TRAF0-Programm enthaltenen Regeln.

Die lexikalische Analyse liefert eine Kette von Symbolen oder Worten mit Attributen. Daraus versucht das Programm, die in der TRAF0-Sprache definierten Strukturen zu erkennen, und speichert sie anschließend.

Diese gespeicherten Strukturen werden dann systematisch nach den im TRAFO-Programm festgelegten Regeln verändert, in eine neue Kette von Symbolen umgewandelt und letztendlich wieder als ein Strom von Zeichen auf eine Datei ausgegeben.

Die detaillierte Definition der Programmiersprache TRANS liegt an der Technischen Universität Wien am Institut für Landesvermessung und Ingenieurgeodäsie, Abteilung Landesvermessung, auf.

9.2 Anforderungen/Design von TRAFO

Im folgenden wird eine Liste von in der Praxis notwendigen Eigenschaften (also das 'Pflichtenheft') von TRAFO vorgestellt (Diese Forderungen werden von TRAFO erfüllt - siehe Kapitel 9.3 'Definition von TRAFO'):

I) Allgemeines:

Entwicklung einer Programmiersprache und eines dazugehörigen Übersetzungsprogrammes mit folgenden Aufgaben:

- 1) Definition der lexikalischen Elemente eines Quelltextes und der dazugehörigen Änderungen.
- 2) Definition der strukturellen Elemente eines Quelltextes und der dazugehörigen Änderungen.
- 3) Diese beiden Definitionen sollen mit Hilfe einer kontextfreien attributierten Grammatik formuliert werden, wobei für beide Teile dieselben syntaktischen Elemente zur Anwendung kommen.
 - Anmerkung: Diese Bedingung (gleiche Syntax für lexikalische und strukturelle Analyse) wird von LEX/YACC (siehe Kapitel 'Existierende Werkzeuge') nicht erfüllt.
- 4) Die Grenze zwischen lexikalischen und strukturellen Elementen muß eine fließende sein, damit man Probleme, wie z.B. kontextabhängige lexikalische Definitionen, bewältigen kann.
 - Anmerkung: Diese Bedingung kann durch LEX/YACC (siehe Kapitel 'Existierende Werkzeuge') nur durch Eingliederung von zusätzlichen Prozeduren, die in der Programmiersprache C geschrieben sind, erfüllt werden.

Diese Forderung wird von TRAFO aus folgenden Gründen erfüllt:

- a) Die Beschreibung der Symbole und der Strukturen erfolgt nach derselben Syntax.
- b) Die Interpretation dieser Beschreibung wird in einem Durchgang durchgeführt.
- c) Die einfachste Struktur besteht aus einem einzelnen Symbol.
- d) Eine Struktur kann durch Referenz auf eine andere von dieser abhängig gemacht werden.
- e) Die Beschreibung der Strukturen kann rekursiv sein.

- 5) Die Programmiersprache muß die Möglichkeit enthalten, auch schlecht oder gar nicht strukturierte Elemente (und ihre Änderungen) eines Textes zu beschreiben. Beispiel: GOSUB in BASIC / FORMAT in FORTRAN u.s.w.

► Anmerkung: Dies ist wohl eine typische Schwäche der Werkzeuge LEX/YACC (siehe Kapitel 'Existierende Werkzeuge'). Eine (für die Praxis zu umständliche) Methode wäre folgende: Verwendung von Zeilennummern als Attribute und Erzeugung zweier (!) Compiler(-teile) durch LEX/YACC und einen C-Compiler, wobei der zweite beispielsweise die so gefundenen Prozeduren für PASCAL an den Programmbeginn setzt, 'überflüssige' Zeilennummern eliminiert u.s.w.

In TRAF0 wird das Problem auf folgende Weise gelöst: Durch eine Referenz auf eine Struktur, die irgendwo im Quelltext vorkommen und auch Teil von anderen Strukturen sein kann, wird auch die Behandlung der oben genannten Elemente möglich. Beispiel: Eine Zeilennummer ist in BASIC dann der Kopf einer Prozedur, wenn irgendwo im Text ein GOSUB-Befehl mit derselben Zeilennummer dahinter steht. Durch nur 4 kurze TRAF0-Befehle (siehe Beispiel im Kapitel 10) wird die Prozedur als solche erkannt, ebenso wie jeder ihrer Aufrufe umgeformt und an den Programm-Beginn gestellt wird.

- 6) Erstellung eines Übersetzungsprogrammes (Compilers), welches obige Definitionen interpretiert und anschließend einen oder mehrere Texte transformiert. Der Vorgang der Übersetzung geschieht somit mit einem einzigen Aufruf eines Compilers, was die Kosten von Personal und Betriebsmitteln minimiert.

► Anmerkung: Diese Bedingung wird von LEX/YACC (siehe Kapitel 'Existierende Werkzeuge') nicht erfüllt. LEX/YACC stellen nur Werkzeuge dar, einen Compiler zu entwickeln und machen somit eine zusätzliche Übersetzung (in diesem Fall durch einen C-Compiler) notwendig.

Der Compiler TRAF0 hingegen wurde mit Hilfe der Programmiersprache TURBO-PASCAL für die IBM-Personalcomputer unter dem Betriebssystem MS-DOS entwickelt und führt in einem einzigen Aufruf die oben genannten Aktionen durch.

- 7) Diese neue Programmiersprache muß so beschaffen sein, daß sie leicht auf verschiedenen Computersystemen mit verschiedenen Betriebssystemen zu implementieren ist.

► Anmerkung: Diese Bedingung wird von LEX/YACC (siehe Kapitel 'Existierende Werkzeuge') nicht erfüllt, da sie typische Werkzeuge für Computer mit dem Betriebssystem UNIX sind. Außerdem wird bei der Verwendung von LEX/YACC zusätzlich ein C-Compiler benötigt.

Der entwickelte Compiler TRAF0 kann unabhängig von weiteren Hilfsprogrammen oder Programmiersprachen betrieben werden und ist deshalb leicht auf anderen Computersystemen zu installieren, indem der Programm-Text von TRAF0 durch TRAF0 selbst transformiert wird.

II) Lexikalische Analyse:

- 1) Definition der Mengen von Zeichen, um bei der Umwandlung des Stromes von Zeichen in einen Strom von Symbolen (Token) Teile von Worten (Lexemen) erkennen zu können.

► Anmerkung: In TRAF0 wird der Datentyp SET OF CHAR aus PASCAL verwendet.

a) Beispiel: Programmiersprache HP85-BASIC:
Bezeichner für Variable: ['A'..'Z', '0'..'9', '\$'].

Dies ist die Menge aller Zeichen, die in einem Bezeichner für eine Variable vorkommen können. Diese Menge kann durch mehrere Bereiche und einzelne Zeichen angegeben werden.

b) Beispiel: Registriereinheit WILD-GRE4:
Horizontaler Winkel: ['+', '-', '0'..'9'].

Bei dieser Registriereinheit wird jeder Wert ohne Dezimalpunkt gespeichert. Daher kann auch ein horizontaler Winkel nur aus Zeichen obenstehender Menge bestehen.

2) Definition der Anzahl von Zeichen aus einer Menge von Zeichen für einen Teil eines Wortes, wobei die Anzahl eine Menge aus ganzen positiven Zahlen ist.

- Anmerkung: Diese einfache Möglichkeit existiert in LEX (siehe Kapitel 'Existierende Werkzeuge') nicht, da dort als Anzahl nur 0, 1 oder ∞ möglich und üblich ist.

In TRAF0 wird der Datentyp SET OF BYTE aus PASCAL verwendet, gefolgt von dem TRAF0-Schlüsselwort 'of' und einer Zahlenmenge. Fehlt die Anzahl, wird [1] angenommen. Da TRAF0 auf einem Personalcomputer entwickelt wurde, mußte die Einschränkung getroffen werden, daß in der Menge nur Zahlen vorkommen dürfen, deren Werte zwischen 0 und 254 liegen. Der Wert 255 steht dann stellvertretend für $+\infty$.

- a) Beispiel: Programmiersprache HP85-BASIC:
Präfix für Bezeichner: [1,3..5] of ['A'..'Z'].

In HP85-BASIC kann ein Bezeichner für eine Variable aus einem Buchstaben, gefolgt von eventuell einer Ziffer und eventuell einem Dollarzeichen (bei Zeichenketten-Variablen) bestehen. Andererseits kann ein Bezeichner für einen BASIC-Befehl bzw. eine BASIC-Funktion aus 3 bis 5 Buchstaben bestehen. Es gibt aber keinen programmierbaren Befehl und keine Funktion, die aus 2 Buchstaben bestehen. Will man daher in HP85-BASIC Bezeichner allgemein als Symbolklasse zusammenfassen, so muß festgehalten werden, daß ein Bereich für die Anzahl der Zeichen nicht genügt!

Somit ist die einzige Struktur, die allen diesen Anforderungen gerecht werden kann, die Menge. Außerdem wird TRAF0 einfacher und besser lesbar, wenn nur wenige Strukturen vorkommen - schließlich ist der einzige Unterschied zwischen der Angabe von Zeichen und einer Anzahl, daß in der Zeichenmenge Zeichen stehen und in der Menge für die Anzahl ganze positive Zahlen.

- b) Beispiel: Registriereinheit WILD-GRE4:
Füllzeichen: [1..2,4] of ['.'].

Im Datenformat dieser Registriereinheit sind Blöcke mit jeweils genau 15 Zeichen vorgesehen. Die ersten beiden Zeichen geben an, worum es sich jeweils handelt (Richtung, Entfernung etc.) und die letzten 8 Zeichen repräsentieren den eigentlichen Wert. Dazwischen stehen Zusatzinformationen und einzelne Punkte als Füllzeichen. Es dürfen aber nur 1 bis 2 oder 4 Punkte hintereinander stehen. Daher würde auch hier ein einfacher Bereich für die Anzahl nicht genügen.

- 3) Definition eines Spaltenbereiches (= Position eines Wortes in der jeweiligen Zeile) für einen Teil eines Wortes, wobei dieser Spaltenbereich eine Menge von ganzen Zahlen ist.

► Anmerkung: Diese Bedingung wird von LEX (siehe Kapitel 'Existierende Werkzeuge') nicht erfüllt, da dort Lexeme nicht in Abhängigkeit von Spaltenbereichen gebracht werden können. Das Problem könnte eventuell mit zusätzlichen C-Modulen und den Spalten als Attribute gelöst werden.

In TRAFO wurde der maximale Spaltenbereich auf 1 bis 255 eingeschränkt, da Zeilen in Text-Dateien auf dem Personalcomputer maximal 255 Zeichen lang sein können.

a) Beispiel: Programmiersprache FORTRAN:

Label: [1..5] of ['0'..'9'] incol [1..5].

b) Beispiel: Registriereinheit WILD-GLE1:

Code: [1..3] of ['0'..'9'] incol [20..22,24..26].

- 4) Definition einer Transformation von Zeichen eines Teiles eines Wortes, die jeweils während der lexikalischen Analyse durchgeführt wird, bevor (!) überprüft wird, ob die Zeichen des gefundenen Wort-Teiles in der angegebenen Zeichenmenge enthalten sind, ob die Anzahl der Zeichen Element der Anzahl-Menge ist und ob die Zeichen innerhalb des eventuell angegebenen Spaltenbereiches stehen.

► Anmerkung: In TRAFO kann die Tabelle für die Transformation an die Definition des Teiles eines Wort-Typs angehängt werden. Der 'linke' Teil der Tabelle besteht aus dem TRAFO-Schlüsselwort 'fromchar' und einer Folge von Zeichen, die der Syntax der Datenstruktur SET OF CHAR von Pascal entsprechen. Analog besteht die 'rechte' Seite aus dem TRAFO-Schlüsselwort 'tochar' und einer Folge von Zeichen, die wieder der Syntax der Datenstruktur SET OF CHAR von Pascal entsprechen. Die Anzahl der Zeichen in den beiden Zeichenmengen muß gleich sein. Wird dann bei der lexikalischen Analyse im zu testenden Bereich ein Zeichen gefunden, welches identisch mit dem ersten Zeichen der 'linken' Spalte der Tabelle ist, so wird dieses Zeichen durch das erste Zeichen der 'rechten' Spalte ersetzt. Analog wird mit dem zweiten Zeichen verfahren u.s.w. Zusätzlich kann das TRAFO-Schlüsselwort UPCASE angegeben werden (auch ohne Tabelle), welches Kleinbuchstaben durch Großbuchstaben ersetzt.

- a) Beispiel: Programmiersprache HP85-BASIC->PASCAL:
Inhalt von Strings: [...] of [#0..#38,#40..#255]
fromchar [#39, '"," '\`']
tochar ['\`', #39, '"'] .

In diesem Beispiel wird beschrieben, wie der Inhalt von Strings aussehen muß. In HP85-BASIC wird der Inhalt von Strings durch zwei doppelte Apostrophe " begrenzt, innerhalb eines Strings kann der einfache Apostroph ' vorkommen. In PASCAL ist es genau umgekehrt.

Daher ist die Menge der Zeichen definiert durch alle ASCII-Zeichen (Ordnung 0 bis 255) außer dem Zeichen mit der Ordnung 39 (das ist der einfache Apostroph '). Die Anzahl ist beliebig. Vor dem Test des Wort-Teiles wird folgende Transformation durchgeführt:

- 1.) ' => `
- 2.) " => '
- 3.) ` => "

Mit anderen Worten: jeder einfache Apostroph ' wird durch einen doppelten " ersetzt und umgekehrt. Der umgekehrte einfache Apostroph ` kann dabei als Zwischenspeicher dienen, da er in HP85-BASIC niemals vorkommen darf. Damit wird gewährleistet, daß das gespeicherte Wort bereits die PASCAL-gemäßen Zeichen enthält.

- b) Beispiel: Registriereinheit AGA-GEODAT:
Entfernung: [1..9] of ['0'..'9', '.']
fromchar ['.','] tochar ['.'] .

Hier kann auf einfache Weise ein beliebiger Tippfehler behoben werden: Wenn der numerische Wert einer Entfernung nicht automatisch registriert, sondern 'händisch' eingetippt wird (z.B. weil eine besonders kurze Entfernung mit einem Maßband und nicht mit einem Tachymeter gemessen wurde), so wird oft ein Beistrich statt des Dezimalpunktes eingetippt, wenn dies die Registriereinheit zuläßt.

Im obenstehenden Beispiel wird vor dem Test automatisch aus jedem Beistrich ein Dezimalpunkt gemacht, da angenommen werden kann, daß ein Beistrich nur durch die Verwechslung mit dem Punkt, anlässlich der Eingabe einer Entfernung, gespeichert wurde.

- 5) Vollständige Definition von Symbolen (Token) innerhalb der lexikalischen Analyse.

► Anmerkung: In TRAFO wird jedem Wort eindeutig eine Symbol-Klasse zugeordnet, welche durch einen Namen symbolisiert wird. Jeder Symbol-Klasse wird in TRAFO eine Definition zugewiesen, die nach dem Symbol '::=' aus den einzelnen Definitionsteilen, getrennt durch Additionszeichen und abgeschlossen durch einen Strichpunkt, besteht.

- a) Beispiel: Programmiersprache HP85-BASIC:

```
Bezeichner: BEZEICHNER ::= [1,3..5] of ['A'..'Z'] +
                [0..1]   of ['0'..'9'] +
                [0..1]   of ['$']      ;
```

Hier besteht ein Bezeichner aus drei Teilen: Aus 1 oder 3 bis 5 Großbuchstaben, eventuell aus einer Ziffer und eventuell noch aus einem Dollar-Zeichen. So können Teile von Symbolen beliebig kombiniert werden.

- b) Beispiel: Registriereinheit AGA-GEODAT:

```
Kennsatz für Entfernung: DIST ::= ['9'] + ['='] ;
```

Die Zeichenfolge '9=' ist in dieser Registriereinheit das Präfix einer Entfernung.

- 6) Angabe einer Gesamtanzahl von Zeichen innerhalb eines Wortes. Diese Gesamtanzahl soll analog einer Anzahl von Zeichen durch eine Menge von ganzen positiven Zahlen repräsentiert werden.

► Anmerkung: Diese Möglichkeit könnte mit LEX/YACC (siehe Kapitel 'existierende Werkzeuge') indirekt mit Hilfe der YTEXT-Funktion oder wieder nur mit neu zu entwickelnden C-Routinen gelöst werden, die der lexikalischen Analyse nachgeschaltet werden.

In TRAFO ist diese Funktion bereits implementiert. Es wird der Datentyp SET OF BYTE aus PASCAL und davor das TRAFO-Schlüsselwort 'length' verwendet. Fehlt diese Angabe, wird keine Überprüfung durchgeführt. Wurde eine derartige Einschränkung gegeben, und ist aber bei einem Wort die gefundene Gesamtzahl von Zeichen nicht innerhalb der Menge, so erfolgt eine Warnung. Diese ganzen positiven Zahlen müssen allerdings im numerischen Bereich zwischen 0 und 255 liegen. Der Wert 255 steht dann stellvertretend für $+\infty$.

a) Beispiel: Programmiersprache FORTRAN=>FORTRAN.

Länge von Bezeichnern für Variable:

```

VARNAME length [1..6] ::=
    [1] of ['A'..'Z']uppercase +
    [0..7] of ['A'..'Z','0'..'9']uppercase;

```

Manche FORTRAN-Compiler akzeptieren Bezeichner für Variable mit einer Länge bis zu 8 Zeichen, manche erlauben nur eine Länge von 6 Zeichen. Wenn nun die Aufgabe besteht, einen Programmtext zu konvertieren, der 8 Zeichen erlaubt, und das Zielsystem nur 6 Zeichen gestattet, so müssen außerhalb der automatischen Umformung gesonderte Arbeiten durchgeführt werden (umgekehrt wäre es ja kein Problem).

In obigem Beispiel wird ein Bezeichner für eine Variable dann erkannt, wenn er aus bis zu 8 Buchstaben oder Ziffern besteht, wobei das erste Zeichen ein Buchstabe sein muß und vor dem Test alle Kleinbuchstaben in Großbuchstaben umgewandelt worden sind, da in FORTRAN im allgemeinen nicht zwischen Groß- und Kleinschreibung unterschieden wird. Wenn aber der Bezeichner insgesamt mehr als 6 Zeichen lang ist, erfolgt eine Warnung - zur Dokumentation.

b) Beispiel: Registriereinheit AGA-GEODAT:

Entfernung:

```

ENTFERNUNG length [1..12] ::=
    [0..9] of ['0'..'9'] +
    [1] of ['.'] +
    [0..9] of ['0'..'9'];

```

In diesem Beispiel wird eine Entfernung definiert, die aus einer Dezimalzahl besteht, wobei vor und hinter dem Dezimalpunkt jeweils bis zu 9 Ziffern stehen können, womit dieser Wert als Dezimalzahl erkannt wird. Wenn die Anzahl der Zeichen insgesamt 12 überschreitet, erfolgt eine Warnung. Dies wäre beispielsweise dann der Fall, wenn die Zahl zwar richtig und lesbar gespeichert ist, die Gesamtzahl an signifikanten Stellen aber das übliche Maß von 12 überschreitet (z.B. 6-Byte-REAL unter MS-DOS). Eine solche Zahl könnte zwar gelesen, die überschüssigen Ziffern aber nicht verwendet werden.

7) Definition von numerischen Symbol-Klassen.

- Anmerkung: Diese Möglichkeit könnte mit LEX/YACC (siehe Kapitel 'existierende Werkzeuge') analog dem vorherigen Abschnitt nur mit neu zu programmierenden C-Routinen gelöst werden, die der lexikalischen Analyse nachgeschaltet werden.

In TRAF0 ist diese Möglichkeit bereits implementiert: Vor und nach dem Namen einer Symbol-Klasse kann jeweils eine numerische Unter- bzw. Obergrenze angegeben werden, jeweils vom Bezeichner durch das Symbol '<=' getrennt. Wird für eine Symbol-Klasse eine solche Grenze angegeben, wird nach dem Test der Zeichenmengen versucht, das Symbol als numerischen Wert zu lesen und darauf mit den angegebenen Grenzen zu vergleichen. Ist es dann bei einem Symbol des entsprechenden Typs nicht möglich, einen Wert zu lesen, oder liegt der gelesene Wert nicht innerhalb der angegebenen Grenzen, wird eine Warnung ausgegeben.

a) Beispiel: Programmiersprache FORTRAN:

INTEGER-Konstante:

```
-32768<=INTEGER<=32767 ::= [1..9] of ['0'..'9']
                                incol [7..72] ;
```

In diesem Beispiel wird die Symbol-Klasse einer INTEGER-Konstanten in FORTRAN definiert. Sie kann aus bis zu 9 Ziffern bestehen und muß zwischen Spalte 7 und 72 gespeichert sein. Wenn sie allerdings -32768 unter- bzw. +32767 überschreitet, erfolgt eine Warnung (dies entspricht einer ganzen Zahl, die mit 2 Bytes gespeichert ist). Somit ist sie als INTEGER-Konstante erkannt, aber es folgt ein Hinweis auf einen möglicherweise fehlerhaften Typ.

b) Beispiel: Registriereinheit AGA-GEODAT:

Entfernung:

```
1.5 <= ENTFERNUNG <= 2000.0
    length [1..12] ::=
        [0..9] of ['0'..'9'] +
        [1] of ['.'] +
        [0..9] of ['0'..'9'] ;
```

Dieses Beispiel ist eine Erweiterung von 8b). Zusätzlich erfolgt nun auch dann eine Warnung, wenn der numerische Wert, also die Entfernung, weniger als 1.5m oder mehr als 2km beträgt. Dies ist nämlich ein typischer Meßbereich eines Tachymeters.

Da keine Entfernung mit weniger als 1.5m gemessen werden kann, erfolgt ebenso eine Warnung wie bei einer Entfernung über 2km, welche eine typische maximale Reichweite für Tachymeter ist.

III) Struktur-Analyse:

- 1) Definition von Strukturen, die aus mehreren Teilen bestehen können. Dabei dürfen die Namen für diese Strukturen mehrfach vergeben werden. Die Teile sind wiederum bestimmte Symbole, Typen von Symbolen oder Strukturen.

► Anmerkung: In TRAF0 können die Namen für Strukturen mehrfach vergeben werden. Die Definition kann aus mehreren Teilen bestehen. Jeder Teil kann wieder eines aus den oben genannten Elementen sein.

- a) Beispiel: Programmiersprache HP85-BASIC:
Additive Operatoren: $\text{ADD_OP} ::= '+' ;$
 $\text{ADD_OP} ::= '-' ;$

So wird gewährleistet, daß eine bestimmte Struktur die eine ODER andere Gestalt haben kann: ADD_OP kann ein '+' oder ein '-' sein.

- b) Beispiel: Registriereinheit WILD-GRE4:
Informationen: $\text{INFO} ::= \text{INFO1} ;$
 $\text{INFO} ::= \text{INFO2} ;$
 $\text{INFO} ::= \text{INFO3} ;$

Hier kann eine Struktur INFO aus einer Struktur INFO1 oder aus einer Struktur INFO2 oder aus einer Struktur INFO3 bestehen.

- 2) Hierarchische Definition von Strukturen.

► Anmerkung: In TRAF0 bestimmt einfach die Reihenfolge der Struktur-Definitionen die Hierarchie.

- a) Beispiel: Programmiersprache HP85-BASIC:
Arithmetische Ausdrücke:
 $\text{EXPR} ::= \text{WERT} + \text{MUL_OP} + \text{WERT} ;$
 $\text{EXPR} ::= \text{WERT} + \text{ADD_OP} + \text{WERT} ;$

Hier wird eine Struktur EXPR definiert, die jeweils aus zwei WERTen mit einem Operator dazwischen besteht. Dabei werden die MUL_OP (multiplikativen Operatoren) vor den ADD_OP (additiven Operatoren) ausgewertet. Im Ausdruck $3*8+5+6*8$ würde somit zuerst $3*8$ als EXPR identifiziert werden, dann $6*8$, dann erst $24+5+48$.

- b) Beispiel: Registriereinheit AGA-GEODAT:
 Ziffern: KEY ::= [1] of ['0'..'9'] + ['='] ;
 NUM ::= [1] of ['0'..'9'] ;

In diesem Beispiel wird zuerst eine Schlüssel-Zeichenfolge KEY definiert, die aus einer Ziffer und einem '=' besteht. Alle diese Zeichenkombinationen in der Quell-Datei werden dann zu einer Struktur mit dem Namen KEY zusammengefaßt. Dann erst wird jeder 'übriggebliebenen' Ziffer der Name NUM zugeordnet. Würden obige Definitionen in der umgekehrten Reihenfolge stehen, so würde konsequenterweise kein KEY erkannt.

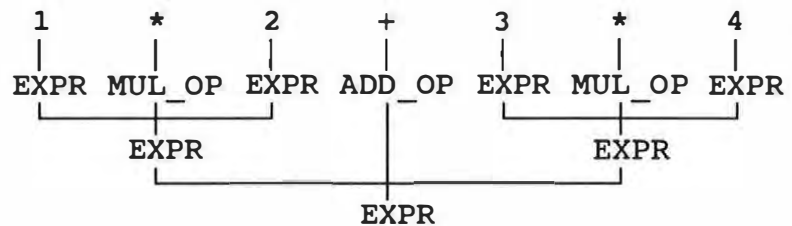
3) Rekursive Definition von Strukturen.

- Anmerkung: In TRAFO kann die Definition jeder Struktur direkt oder indirekt rekursiv erfolgen.

- a) Beispiel: Programmiersprache HP85-BASIC:
 Arithmetische Ausdrücke:
 EXPR ::= <INTEGER> ;
 EXPR ::= EXPR + MUL_OP + EXPR ;
 EXPR ::= EXPR + ADD_OP + EXPR ;

Hier wird eine Struktur (ein Ausdruck) mit dem Namen EXPR definiert. Sie kann aus Symbolen des Typs <INTEGER> (z.B. eine ganze Zahl) bestehen, oder aus multiplikativen oder additiven Operatoren zwischen Ausdrücken.

So würde der Ausdruck 1*2+3*4 in folgender Weise analysiert werden:



- b) Beispiel: Registriereinheit WILD-GRE4:
 Zeilenende/Dateiende: EOI ::= <EOL> ;
 EOI ::= ' ' ;
 EOI ::= EOI + '! ----' ;

Hier wird festgehalten, daß ein Zeilenende zu einer Struktur mit dem Namen EOI wird (end of information), ebenso ein Leerzeichen oder wiederum ein EOI und den Zeichen '! ----'.

- 4) Jede Struktur muß neben ihrer Definition auch eine Definition ihrer Änderung unter Verwendung von lokalen Namen und globalen Referenzen enthalten können.

► Anmerkung: In TRAFO wird an die Definition einer Struktur die Definition der entsprechenden Änderung angehängt. Es können Bezeichner für die Teile der Definitionen mit lokaler Gültigkeit ebenso verwendet werden, wie daran angehängte Verweise auf andere Strukturen mit globaler Gültigkeit.

a) Beispiel: Programmiersprache HP85-BASIC=>PASCAL:
 Kommentar: REM ::= <COMMENT> = .COM.
 + <EOL> = .EOR.
 => '{' & .COM. & '}' ;

Hier wird die Struktur REM (remark) definiert durch ein Symbol des Typs COMMENT und eines des Typs EOL (end of line). Der erste Teil bekommt den lokalen Namen .COM. und der zweite .EOR. . Dann wird folgende Änderung definiert: Mache aus Strukturen mit dem Namen REM vorerst ein '{', dann setze .COM. (also das im Quell-Text, was dem ersten Teil der Definition entspricht) ein und schließe mit '}' ab. Mit anderen Worten: Es werden die Kommentare aus einem BASIC-Programm in geschwungene Klammern eingeschlossen, wobei der darauffolgende Zeilenwechsel gelöscht wird.

b) Beispiel:Registriereinheit WILD-GRE4=>AGA-GEODAT:
 Entfernung: SEITE ::= <SE> = .COM.
 + [5] of <ZIFFER> = .VKS.
 + [3] of <ZIFFER> = .NKS.
 => '9' & '='
 & .VKS. & '.' & .NKS. ;

Hier wird die Struktur SEITE durch ein Symbol des Typs SE (Schlüsselzeichen für die Seite) und 5 Ziffern (vor dem Dezimalpunkt) und 3 Ziffern (hinter dem Dezimalpunkt) definiert; der Dezimalpunkt wird im Quell-System nicht gespeichert. Daraus sollen dann jeweils die Zeichen '9=', gefolgt von der Zahl gemacht werden, wobei der Dezimalpunkt eingefügt wird.

10 Beispiele

10.1 HP85-BASIC->PASCAL

BASIC (HP85)

(ohne Deklarationen)

```

10 ! Horizontale Entfernung
20 DISP "Eingabe Y1,X1,Y2,X2"; @ INPUT Y1,X1,Y2,X2
30 GOSUB 50 @ DISP "S=";S
40 GOTO 20
50 S=SQR((Y2-Y1)^2+(X2-X1)^2)
60 RETURN
70 END

```

T r a n s f o r m a t o r

TRAFO

```

IMPLICIT_WORDTYPE
REMARK ::= [1] of
IMPLICIT_TREE
KOMMENTAR ::= <REM

```

TURBO_PASCAL (IBM-PC)

(ohne Deklarationen)

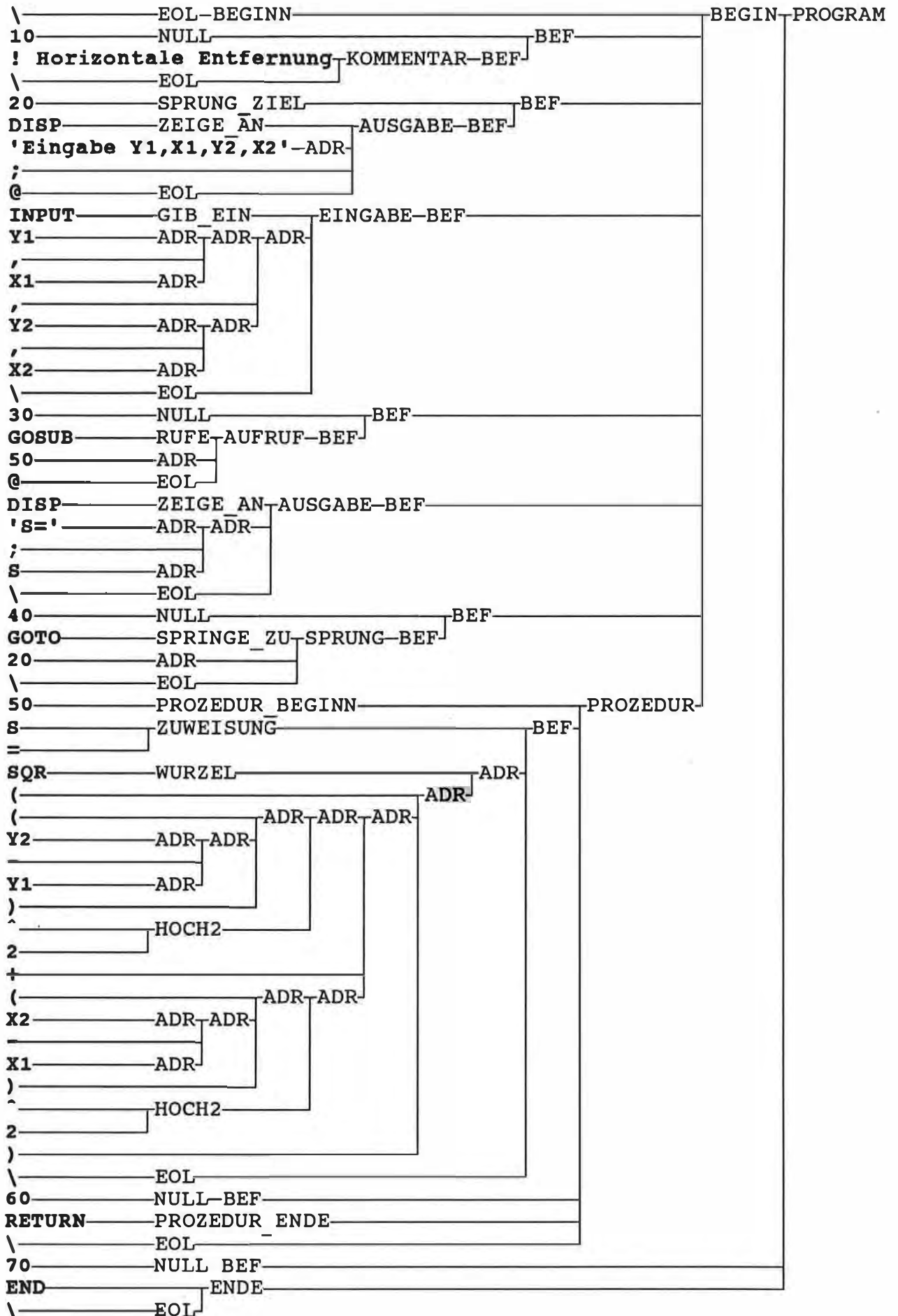
```

program DEMO;

procedure _50;
begin
S:=sqrt(sqr((Y2-Y1))+sqr((X2-X1)));
end;

begin
{! Horizontale Entfernung}
20:write('Eingabe Y1,X1,Y2,X2');readln(Y1,X1,Y2,X2);
_50;writeln('S=',S);
goto 20;
end.

```



```
{
{ Definition der lexikalischen Elemente
}
```

IMPLICIT_WORDTYPE

EOL ::= ['\'] ;

REMARK ::= ['!'] + [...] of [#..] ;

LINENUM ::= [1] of ['1'..'9'] incol [1] +
[0..5] of ['0'..'9'] incol [2..6] ;

STRING ::= [#39] fromchar [#39,#34,#96]
tochar [#96,#39,#34] +
[...] of [#0..#38,#40..#255] fromchar [#39,#34,#96]
tochar [#96,#39,#34] +
[#39] fromchar [#39,#34,#96]
tochar [#96,#39,#34] ;

SEPARATOR ::= ['@'] ;

NAME ::= [1] of ['A'..'Z','a'..'z'] +
[...] of ['A'..'Z','a'..'z','_','0'..'9']
fromchar ['\$'] tochar ['_'] ;

EXTENDED ::= [0..1] of ['+', '-'] +
[0..9] of ['0'..'9'] +
[1] of ['.'] +
[1..9] of ['0'..'9'] +
[1] of ['E', 'D'] +
[0..1] of ['+', '-'] +
[1..2] of ['0'..'9'] ;

REAL ::= [0..1] of ['+', '-'] +
[0..9] of ['0'..'9'] +
[1] of ['.'] +
[1..9] of ['0'..'9'] ;

INTEGER ::= [0..1] of ['+', '-'] incol [2..255] +
[1..6] of ['0'..'9'] incol [2..255] ;

GE ::= ['>'] + ['='] ;

LE ::= ['<'] + ['='] ;

NE ::= ['<'] + ['>'] ;

NE ::= ['#'] ;

EQ ::= ['='] ;

GT ::= ['>'] ;

LT ::= ['<'] ;

POT ::= ['^'] ;

MUL ::= ['*'] ; **DIV** ::= ['/'] ; **IDIV** ::= ['\'] ;

ADD ::= ['+'] ; **SUB** ::= ['-'] ;

KOM ::= [','] ; **ABS** ::= [';'] ;

SUBADD ::= ['&'] ;

SUBBEG ::= ['['] ; **SUBEND** ::= ['] ;

FBEG ::= ['('] ; **FEND** ::= [')'] ;

```
{
{ Definition des strukturellen Elemente
}
```

IMPLICIT_TREE

```
{
{ Zeilenschaltung/Verkettung von Befehlen
}
```

```
EOL ::= <SEPARATOR> = .EOR. => ';' ;
EOL ::= <EOL> = .EOR. => ';' & '\ ' ;
```

```
{
{ Kommentar
}
```

```
KOMMENTAR ::= <REMARK> = .REM. +
EOL = .EOR. => '{' & .REM. & '}' & '\ ' ;
```

```
BEF ::= KOMMENTAR ;
```

```
{
{ reservierte Worte
}
```

```
WURZEL ::= 'SQR' ;
SPRINGE_ZU ::= 'GOTO' ;
RUFEN ::= 'GOSUB' ;
PROZEDUR_ENDE ::= 'RETURN' ;
ZEIGE_AN ::= 'DISP' ;
GIB_EIN ::= 'INPUT' ;
ENDE ::= 'END' = .EOP. +
EOL = .EOR. => 'end.' ;
```

```
{
{ Zuweisung
}
```

```
ZUWEISUNG ::= <NAME> = .NAM. +
'=' = .DEF. => .NAM. & ':= ' ;
```

```
{
{ Arithmetik-/String-Ausdrücke
}
```

```
HOCH2 ::= '^' + '2' ;
ADR ::= <INTEGER> ;
ADR ::= <REAL> ;
ADR ::= <EXTENDED> ;
ADR ::= <NAME> ;
ADR ::= <STRING> ;
ADR ::= '(' + ADR + ')' ;

ADR ::= ADR = .NUM. +
HOCH2 = .OP. => 'sqr' & '(' & .NUM. & ')' ;

ADR ::= ADR + '*' + ADR ; ADR ::= ADR + '/' + ADR ;
ADR ::= ADR + '+' + ADR ; ADR ::= ADR + '-' + ADR ;
```

```

ADR ::= ADR + ',' + ADR ;
ADR ::= ADR = .EXP1. +
        ';' = .SEP. +
        ADR = .EXP2. => .EXP1. & ',' & .EXP2. ;

ADR ::= WURZEL = .FNC. + ADR = .EXP. => 'sqrt' & .EXP. ;

BEF ::= ZUWEISUNG + ADR + EOL ;

{
{ Ausgabe auf den Bildschirm
}
}

AUSGABE ::= ZEIGE_AN = .COM. +
            ADR      = .VAR. +
            ';'      = .SEP. +
            EOL      = .EOR.

            => 'write' & '(' & .VAR. & ')' & .EOR. ;

AUSGABE ::= ZEIGE_AN = .COM. +
            ADR      = .VAR. +
            EOL      = .EOR.

            => 'writeln' & '(' & .VAR. & ')' & .EOR. ;

BEF ::= AUSGABE ;

{
{ Eingabe von der Tastatur
}
}

EINGABE ::= GIB_EIN = .COM. +
            ADR      = .VAR. +
            EOL      = .EOR.

            => 'readln' & '(' & .VAR. & ')' & .EOR. ;

BEF ::= EINGABE ;

{
{ Sprung zu einer Zeile
}
}

SPRUNG ::= SPRINGE_ZU = .COM. +
            ADR      = .LAB. +
            EOL      = .EOR. => 'goto' & .LAB. & .EOR. ;

SPRUNG_ZIEL ::= <LINENUM> = .LAB.(SPRUNG)
                => .LAB.(SPRUNG) & ':' ;

BEF ::= SPRUNG ; BEF ::= SPRUNG_ZIEL + BEF ;

```

```
{
{   Aufruf eines Unterprogrammes
}
```

```
AUFRUF ::= RUFE = .COM. +
        ADR  = .LAB. +
        EOL  = .EOR. => '≡' & .LAB. & .EOR. ;
```

```
PROZEDUR_BEGINN ::= <LINENUM> = .LAB. (AUFRUF)
                  => .LAB. (AUFRUF) ;
```

```
BEF ::= AUFRUF ;
```

```
{
{   Unterprogramm
}
```

```
BEF ::= BEF + EOL ;
```

```
PROZEDUR ::= PROZEDUR_BEGINN = .LAB. +
             [..] of BEF      = .BLK. +
             PROZEDUR_ENDE   = .RET. +
             EOL              = .EOR.

             => '\ ' & 'procedure' & '≡' & .LAB. & ';' & '\ ' &
                'begin' & '\ ' & .BLK. & 'end' &
                .EOR. ;
```

```
{
{   Nicht benützte Zeilennummern
}
```

```
NULL ::= <LINENUM> = .LAB. ;
```

```
BEF ::= NULL + BEF ; BEF ::= NULL ;
```

```
{  
{ Programm }  
}
```

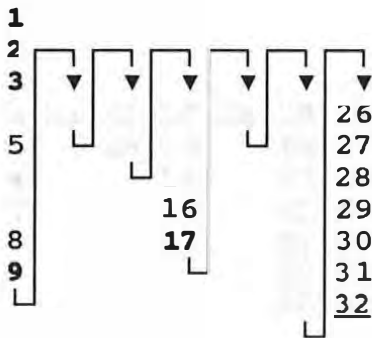
BEGINN ::= EOL = .BEG. => '\ ' & 'begin' & '\ ' ;

**BEGINN ::= BEGINN = .BEG. +
[..] of BEF = .BLK. +
[1..] of PROZEDUR = .PRC.
=> .PRC. & .BEG. & .BLK. ;**

**PROGRAMM ::= BEGINN = .BEG. +
[..] of BEF = .BLK. +
ENDE = .END.
=> 'program' & 'DEMO' & ';' & '\ ' &
.BEG. & .BLK. & .END. ;**

TRAFO

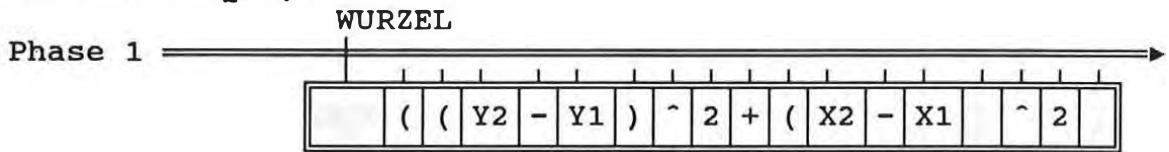
Phasen:



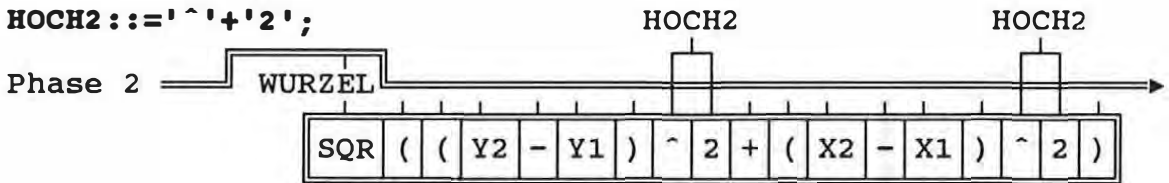
```

IMPLICIT _FREE
WURZEL ::= 'SQR' ;
HOCH2  ::= '^' + '2' ;
ADR    ::= <NAME> ;
ADR    ::= '(' + ADR + ')' ;
ADR    ::= ADR + HOCH2 ;
ADR    ::= ADR + '*' + ADR ;
ADR    ::= ADR + '/' + ADR ;
ADR    ::= ADR + '+' + ADR ;
ADR    ::= ADR + '-' + ADR ;
ADR    ::= WURZEL + ADR ;
    
```

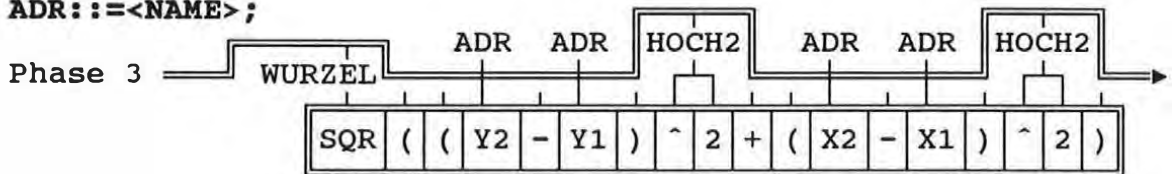
WURZEL ::= 'SQR' ;



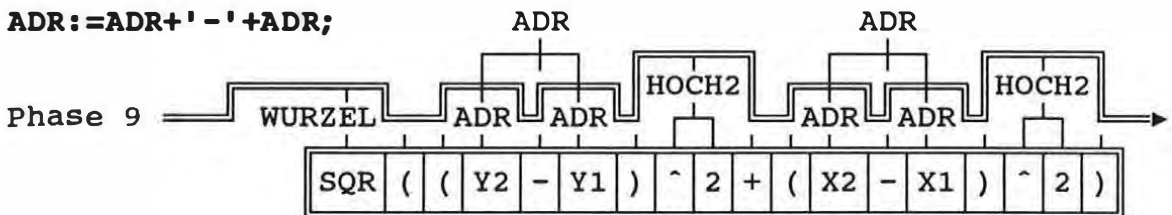
HOCH2 ::= '^'+ '2' ;



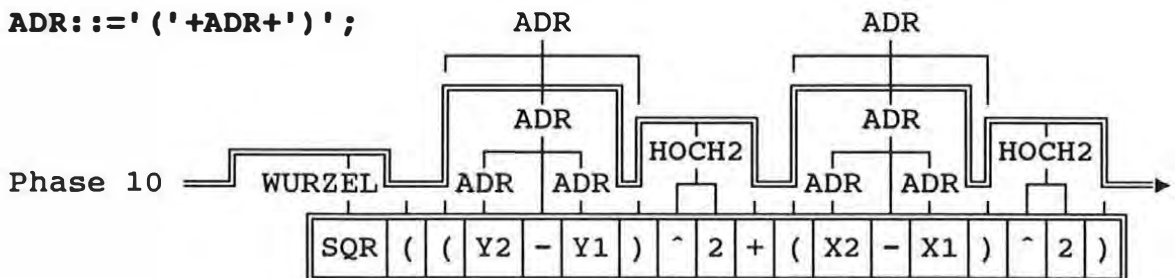
ADR ::= <NAME> ;



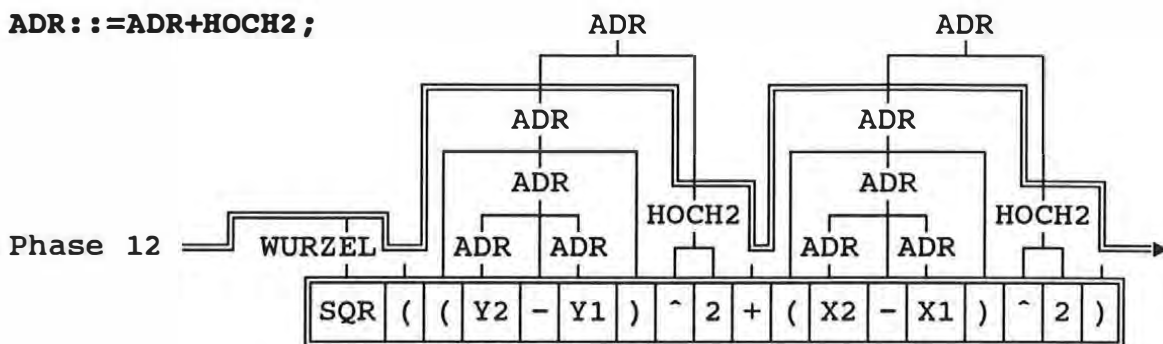
ADR ::= ADR + '-' + ADR ;



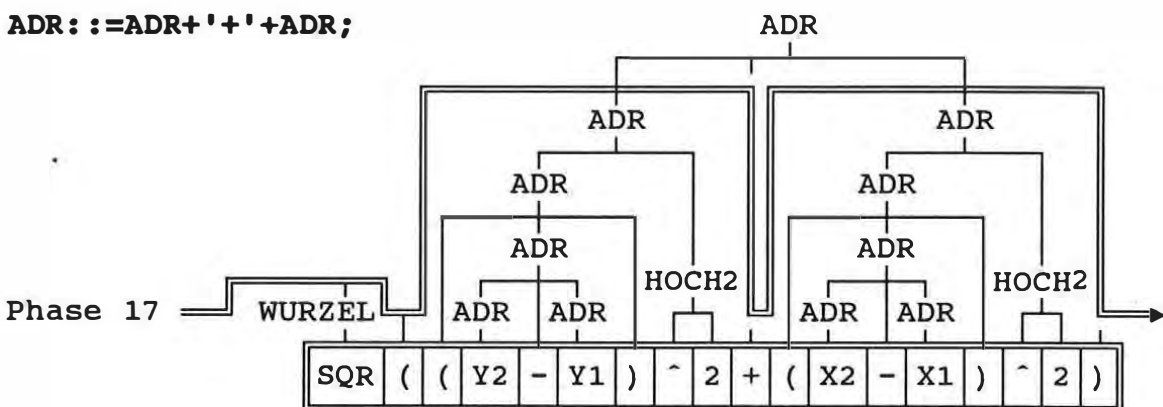
ADR ::= '(' + ADR + ')' ;



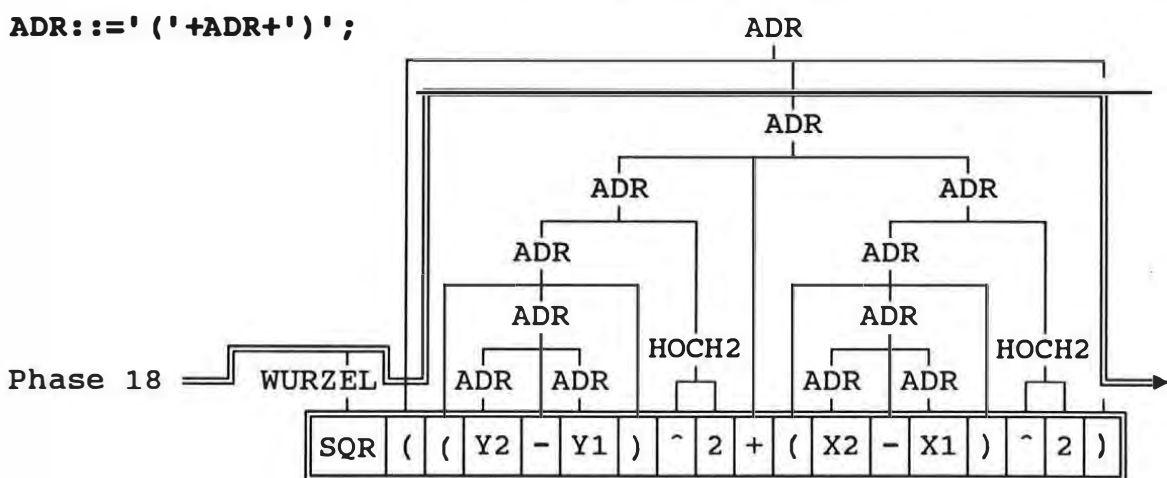
ADR ::= ADR + HOCH2 ;



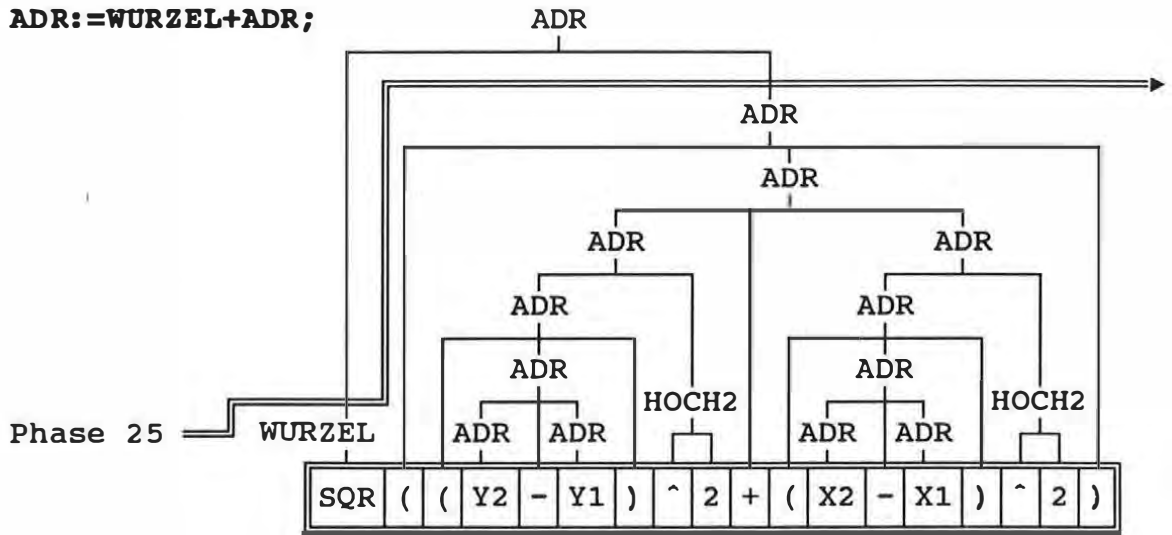
ADR ::= ADR + ' ' + ADR ;

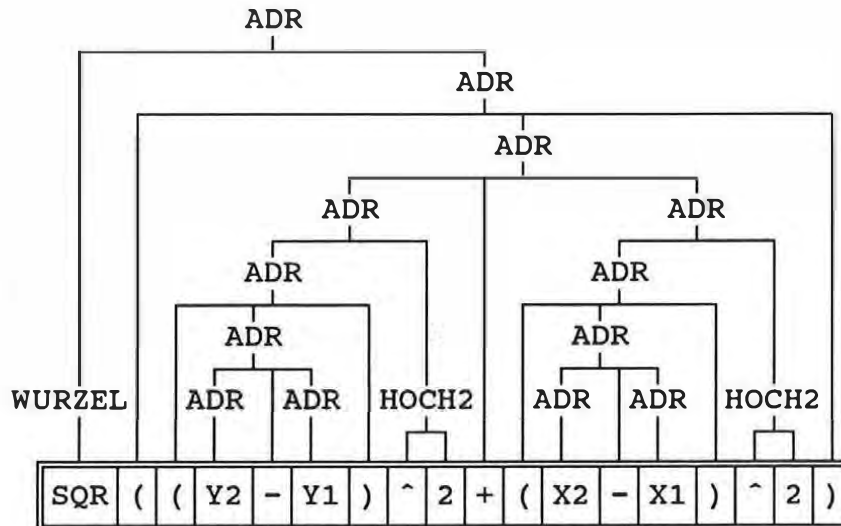


ADR ::= ' (' + ADR + ') ' ;



ADR:=WURZEL+ADR;





TRAFO

```

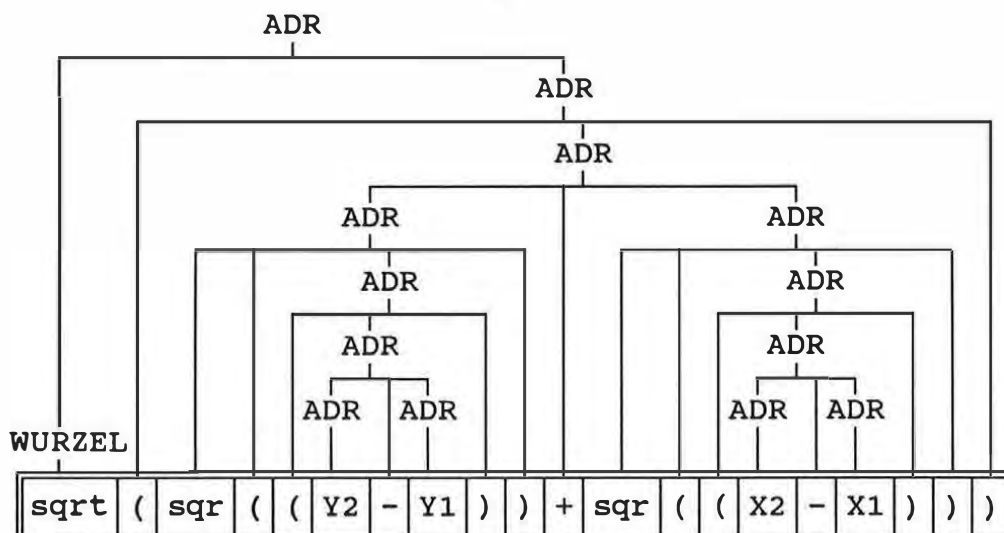
IMPLICIT_TREE

WURZEL ::= 'SQR' ;
HOCH2  ::= '^' + '2' ;
ADR    ::= <INTEGER> ;
ADR    ::= <REAL> ;
ADR    ::= <EXTENDED> ;
ADR    ::= <NAME> ;
ADR    ::= '(' + ADR + ')' ;

ADR ::= ADR = .NUM. + HOCH2 = .OP.
      => 'sqr' & '(' & .NUM. & ')' ;

ADR ::= ADR + '*' + ADR ;
ADR ::= ADR + '/' + ADR ;
ADR ::= ADR + '+' + ADR ;
ADR ::= ADR + '-' + ADR ;

ADR ::= WURZEL = .FNC. +
      ADR = .EXP. => 'sqrt' & .EXP. ;
    
```



10.2 WILD-GRE4->AGA-GEODAT

(GRE4)

```

410004+00000010 42....+01000001
410005+00000012
110006+02000001 21.102+00033243 22.102+00998219
31..00+00026564 51....+0000+000
410009+00000013 42....+00001500
110011+00000001 21.102+02064557 22.102+00976269
31..00+00027827 51....+0000+000
410018+00000019
110019+02000001 21.102+00033249 22.102+00998194
31..00+00026581 51....+0000+000
! ----

```

TRAFO

```

IMPLICIT_WORDTYPE
CD ::= [0..1] of [
IMPLICIT_TREE
CODE ::= <CD> + [8

```

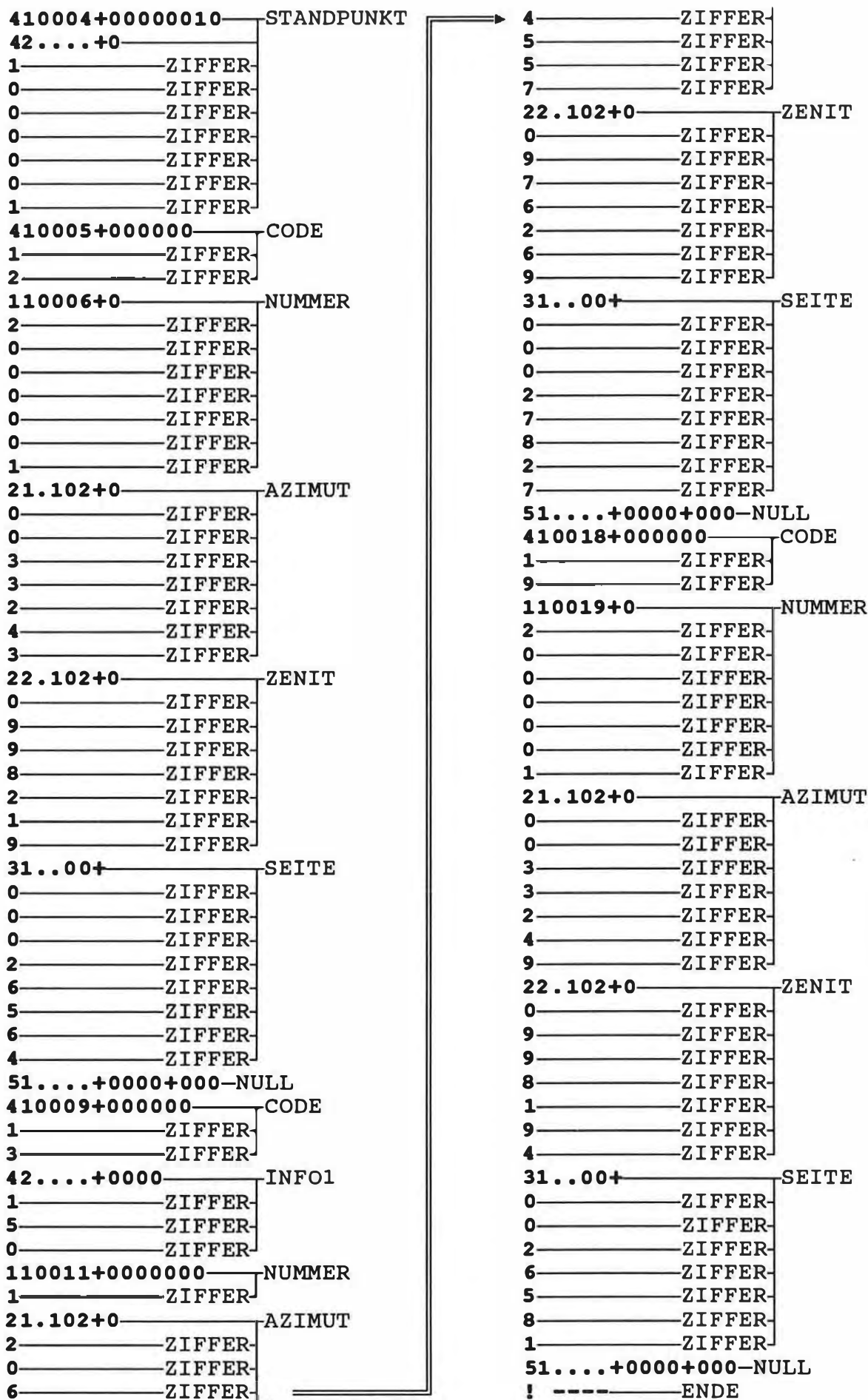
T r a n s f o r m a t o r

(GEODAT)

```

2=1000001
0=12
5=2000001 7=003.3243 8=099.8219 9=00026.564
0=13 1=1500
5=1 7=206.4557 8=097.6269 9=00027.827
0=19
5=2000001 7=003.3249 8=099.8194 9=00026.581

```



```
{
{
{ Definition der lexikalischen Elemente
}
}
}
```

IMPLICIT_WORDTYPE

```
ST ::= [0..1] of [' '] +
      ['4']+['1'] + [4] of ['0'..'9','.'] + ['+'] +
      [6] of ['0'] + ['1'] + ['0'] ;

CD ::= [0..1] of [' '] +
      ['4']+['1'] + [4] of ['0'..'9','.'] + ['+'] +
      [0..7] of ['0'] ;

I1 ::= [0..1] of [' '] +
      ['4']+['2'] + [4] of ['0'..'9','.'] + ['+'] +
      [0..7] of ['0'] ;

I2 ::= [0..1] of [' '] +
      ['4']+['3'] + [4] of ['0'..'9','.'] + ['+'] +
      [0..7] of ['0'] ;

I3 ::= [0..1] of [' '] +
      ['4']+['4'] + [4] of ['0'..'9','.'] + ['+'] +
      [0..7] of ['0'] ;

NR ::= [0..1] of [' '] +
      ['1']+['1'] + [4] of ['0'..'9','.'] + ['+'] +
      [0..7] of ['0'] ;

AZ ::= [0..1] of [' '] +
      ['2']+['1'] + [4] of ['0'..'9','.'] + ['+'] + ['0'] ;

ZE ::= [0..1] of [' '] +
      ['2']+['2'] + [4] of ['0'..'9','.'] + ['+'] + ['0'] ;

SS ::= [0..1] of [' '] +
      ['3'] + ['1'] + [4] of ['0'..'9','.'] + ['+'] ;

KN ::= [0..1] of [' '] +
      ['5'] + ['1'] + [4] of ['0'..'9','.'] +
      [1] of ['+', '-'] +
      [8] of ['0'..'9', '+', '-'] ;

NU ::= [1] of ['0'..'9'] ;

EN ::= ['!'] + [' '] + [4] of ['-'] ;
```

```
{
{ Definition des strukturellen Elemente
}
```

IMPLICIT_TREE

```
ZIFFER ::= <NU> = .ZIF. => '=' & .ZIF. ;
```

```
STANDPUNKT ::=
            <ST>   = .KEY.  +
            <I1>   = .INF.  +
            [1..8] of ZIFFER = .NUM.
            => '\ ' & '2' & '=' & .NUM. ;
```

```
CODE ::= <CD> = .KEY. + [1..8] of ZIFFER = .COD.
        => '\ ' & '0' & '=' & .COD. ;
```

```
INFO1 ::= <I1> = .KEY. + [1..8] of ZIFFER = .INF.
        => '1' & '=' & .INF. ;
```

```
INFO2 ::= <I2> = .KEY. + [1..8] of ZIFFER = .INF.
        => '1' & '=' & .INF. ;
```

```
INFO3 ::= <I3> = .KEY. + [1..8] of ZIFFER = .INF.
        => '1' & '=' & .INF. ;
```

```
NUMMER ::= <NR> = .KEY. + [1..8] of ZIFFER = .NUM.
        => '\ ' & '5' & '=' & .NUM. ;
```

```
AZIMUT ::= <AZ> = .KEY. + [3] of ZIFFER = .NUM1.
           + [4] of ZIFFER = .NUM2.
        => '7' & '=' & .NUM1. & '.' & .NUM2. ;
```

```
ZENIT ::= <ZE> = .KEY. + [3] of ZIFFER = .NUM1.
           + [4] of ZIFFER = .NUM2.
        => '8' & '=' & .NUM1. & '.' & .NUM2. ;
```

```
SEITE ::= <SS> = .KEY. + [5] of ZIFFER = .NUM1.
           + [3] of ZIFFER = .NUM2.
        => '9' & '=' & .NUM1. & '.' & .NUM2. ;
```

```
NULL ::= <KN> = .KEY. ;
```

```
ENDE ::= <EN> = .END. ;
```

11 Schlußwort

Diese Arbeit stellt auszugsweise die Erfahrungen dar, die ich in mehr als 10 Jahren kommerzieller Praxis in einem Bereich zwischen Geodäsie und Informatik gesammelt habe. Im praktischen Einsatz der Umstellung von Vermessungsprogrammen habe ich mich umfassend mit den dabei auftretenden Problemen beschäftigt. Aus den praktischen Anforderungen heraus ist auch die Programmiersprache TRAF0 und das dazugehörige Übersetzungsprogramm entstanden, welche hier zusammen mit alten und neuen Methoden der Übertragung vorgestellt werden.

Die Arbeit ist somit jenen Geodäten gewidmet, die sich entschlossen haben, bewährte Programme auf neue Systeme umzustellen oder sich zu informieren, welche Probleme dabei auftreten können.

Die vorgestellten Methoden aus Theorie und Praxis stellen insgesamt ein Nachschlagewerk dar, welches außerdem eine neue Programmiersprache als Werkzeug enthält. Außerdem sind konkrete Übersetzungen und die dazugehörigen TRAF0-Programme als Parameter dieses Werkzeuges verfügbar. Damit ist der Geodät in der Lage, selbst Transformationen von geodätischen Programmen weitgehend automatisiert durchzuführen.

Selbstverständlich wird TRAF0 weiterentwickelt und an neue Anforderungen angepaßt werden. Ich glaube nicht, daß in absehbarer Zeit eine Vereinheitlichung der Programmiersprachen, der Betriebssysteme oder gar der Computer selbst stattfinden wird - im Gegenteil. Je mehr die Computer in verschiedene Lebensbereiche eindringen, desto unterschiedlicher werden nicht nur die Aufgaben, sondern auch die Lösungen sein.

Daher muß man schon jetzt dafür Sorge tragen, daß der Computer endlich zu dem wird, was er sein sollte: Ein Werkzeug des Menschen; denn das Werk ist das Ziel, nicht das Werkzeug.

12 Literatur

12.1 Allgemein

- [1] Algorithmen und Datenstrukturen
Niklaus Wirth
B.G.Teubner, Stuttgart ; 3.Auflage, 1983
- [2] Assembly Language Interfacing in Turbo-Pascal
Sanjiva Nath
Management Information Source Inc., 1987
- [3] Attribute Grammar Inversion and Source-to-Source Translation
Daniel M. Yellin
Springer Verlag, Berlin Heidelberg 1988
- [4] An efficient context-free parsing algorithm
J. Earley
Comm. ACM 13:2,94-102
- [5] A Two-Way Language Interface
Boris Katz and Patrick H. Winston
P.Degano and E.Sandewall, Holland 1983
- [6] Compiler-Bau, Teil 1 und 2
Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullmann
Addison-Wesely Verlag GmbH, 1988
- [7] Compilers. Principles, techniques,
Alfred V. Aho
Addison-Wesely Verlag GmbH, 1986
- [8] Compiling Circular Attribute Grammars into Prolog
Bijan Arbab
Journal of IBM Research and Development, Vol 303, 1986
- [9] Dictionary for Computer Languages
Hans Breuer
Academic Press Inc., London ; 2.Edition, 1973
- [10] Inverse Translation:
The Theory of Practical Automatic Programming
Steven P. Reiss
PhD thesis, Yale University, December 1977
- [11] Schaltungen,
D.Steinbach
W.Hofacker, Holzkirchen, 1984
- [12] Konzept sprachverstehenden Systems
Harald Trost
Österr. Gesellschaft für Kybernetik, Wien ; 1980
- [13] Mapper, Programmieren Programmierer
Harry Katzan, Jr.
Verlag Paul Haupt Bern und Stuttgart, 1985

- [14] Pascal für Anfänger
Helmut Schauer
R.Oldebourg Verlag Wien München, 3. Auflage, 1979
- [15] Principles of Programming Languages:
Design, Evaluation and Implementation
Bruce J. MacLennan
CBS College Publishing, New York ; 2.Edition, 1987
- [16] Programmieren in Assembler
für die IBM Personal Computer
David J. Bradley
Verlage Carl Hanser/Prentice-Hall international, 1985
- [17] Programming Languages
Allen B. Tucker (Colgate Univ.)
McGraw-Hill, USA ; 2.Edition, 1987
- [18] Taschenwörterbuch der Programmiersprachen
ALGOL, FORTRAN, PL/1
Hans Breuer
Anton Hain Verlag, Meisenheim 1976
- [19] Theory of parsing, translation and compiling
Alfred V. Aho
Prentice-Hall Series in automatic computation, 1973
- [20] Schnittstellenhandbuch
J.Elsing A.Wiencek
IWT Verlag GmbH , Vaterstetten bei München;
2.Auflage, 1987
- [21] Übersetzerbau
Skriptum zur Vorlesung
M.Brockhaus A.Falkner
TU Wien, Inst. f. prakt. Informatik; 2.Auflage, SS 1989

12.2 Manuals

- [22] BASIC-Handbuch (Personal-Computer)
IBM-Deutschland; 1983
- [23] BASIC Handbuch 2.0
VU Literatur 0426, IBM Deutschland; 1983
- [24] Einführung in dBASE III PLUS
Robert Cowart
Sybex Verlag, Düsseldorf ; 2.Auflage, 1987
- [25] Framework für Einsteiger / Version I und II
Voss
Data Becker GmbH , Düsseldorf ; 1978
- [26] GW-BASIC Interpreter
Microsoft Corporation; 1987
- [27] Bedienungs- Programmierhandbuch
Hewlett Packard Company, USA; 1979
- [28] HP86/87 Bedienungs- Programmierhandbuch
Hewlett Packard Company, USA; 1982
- [29] IBM Personal Computer Professional FORTRAN
International Business Machines Corporation; 1984
- [30] Turbo C (IBM Version)
Borland International, USA; 1987
- [31] Turbo Pascal 3.0
Heimsoeth&Borland, München ; 1985
- [32] Turbo Pascal 4.0
Heimsoeth&Borland, München ; 1987
- [33] Turbo Pascal 5.0
Heimsoeth&Borland, München ; 1988
- [34] VAX SCAN
Digital Equipment Corporation, massachusetts ; 1985
- [35] VS FORTRAN Language and Library Reference
International Business Machines Corporation;
2.Auflage, 1985

A TWO-WAY LANGUAGE INTERFACE	76	112
Abschneiden der Zeichen		46
ACK/NAK-Protokoll		14
Algorithmen und Datenstrukturen		112
An efficient context-free parsing algorithm		112
Anforderungen/Design von TRAFO		84
Anzahl der Bits pro Zeichen		18
Assembly Language Interfacin		112
Asynchrones Format		12
ASCII-Codes		34
Attribute Grammar Inversion and Source-to-Source Tran»»»»		112
Attribute		69
Attributierte Grammatik		69
Ausgabe des veränderten ProgrammBaume		66
Auskreuzen		16
Austauschen von einzelnen Bi		38
Änderungen in einer Tabelle		49
binäre Daten	18,34	
BACKUS-NAUR-FORM		67
BASIC Handbuch 2.0		114
BASIC-Handbuch (Pe:		114
BASIC (HP85)		96
BASIC		3
Benutzeroberfläche (front-end)		73
Betriebsarten Senden/Empfänger		10
Betriebssysteme		2
Bezeichner		67
Bitmuster		34
Bitorientierte Transfc		38
Bits bei Übertragunger		33
Bits ersetzen		38
Bits in Datentypen		18
Bits über eine gan:		41
Bits		34
BNF		67
Bottom-Up		65
Büroorganisation		2
Bytes für die Speicherung		22
Byte		34
classes		67
compilers		69
compiler-compile		73
compiler		67
Centronics		15
CIRCULAR ATTRIBUTE GRAMMARS		76
Codeerweiterungszeichen		13
Codesystem		34
Compilers. Principles, techniques, and tools.		112
Compiler-Bau, Teil 1 und 2		112
Compiling Circular Attribute Grammars into Prolog		112
Dateien bei Transformationen		36
Dateien		36
Datenendeinrichtung DEE		9
Datenformat		12
Datenquelle		9
Datensenke		9
Datenübertragungseinrichtung DÜE		9
Definitionen der Dateien		35
Dictionary for Computer Languages		112

Dokumentation	4
einfachen Textsystemen	58
elektronische Realität	17
englische (natürlichsprachliche) Sätze	76
Einführung in dBASE III PLUS	114
Einführung	1,8,18,39,41,44,51,58,61,80
Eingangsparameter	46
Einträge in der Tabelle	49
Entwicklung und Produktion	2
Ergebnis ausgeben	66
Erweiterungen (Extensions) der Datei-Namen	36
Existierende Werkzeuge	67
Extensions	36
File	36
Filter	76
Flexibilität	46
Formatsteuerzeichen	13
Framework für Einsteiger / Version I und II	114
ganze Zahlen in der Binärarithmetik	20
Ganze positive Zahlen	20
Ganze Zahlen	22
Generatorsprache	69
Genereller Austausch von Bits	41
Genereller Austausch von Zeichen	48
Gerätsteuerzeichen	13
Gleitkommazahlen	24
Grenzfälle	46
Groß- bzw. Kleinschreibung unterschieden	57
Grundlagen	8
GW-BASIC Interpreter	114
hardware handshake	14
Halb-Duplex	10
Hardwareprotokoll	14
Hardware-Protokolles	33
Hardware	1
Hauptspeicher	2
Hierarchie in Tabellen	53
HP1000 (REAL*4 - FORTRAN 77)	32
HP3000 (REAL - FORTRAN 66)	31
HP3000-XL (REAL*4 - FORTRAN 77)	31
HP85 Bedienungs- und Programmierhandbuch	114
HP85-BASIC->PASCAL	96
HP86/87 Bedienungs- und BASIC Programmierhandbuch	114
HP9000 (REAL*4 - FORTRAN 77)	32
identifizier	67
IBM Personal Computer Professional FORTRAN	114
IC Schaltungen, TTL CMOS LINEAR	112
Inhaltsübersicht	5
Input/Output-Operationen minimiert	53
Inverse Translation: The Theory of Practical Automati»»»»	112
INVERSE TRANSLATION	75
kleinste Informationseinheit	17
kontextfreie Grammatik	67
Klassen	67
Konstante und Variable	59
Konvertierung	38,41,43
Konzept eines sprachverstehenden Systems	112
längste Präfix	70
lernenden System	50

linksassoziativ (Auswertung von links nach rechts)	74
logische Datenübertragung der Messungen	3,4
Landeskoordinaten	18
Lexikalische Analyse	63,67
LEX-Compiler	70
LEX-Sprache	70
LEX	70
Literatur	112
Logische Definition	17,35,36
Lookahead-Operator	71
maskenorientierte Transformation	78
maskenorientierter Übersetzer-Generator	76
mehr Speicherplatz	46
mehrere Dateien	49
Makro-Sprachen	50
Manuals	114
Mapper, Programmieren ohne Programmierer	112
Maskenorientierte Transformationen	58
Massenspeicher	3
Modem	9
Nachschlagwerk	3
Neuinstallation	49
Normierung von Daten	64
pass	67
physikalische Datenübertragung der Registriereinheiten ...	3
physikalische Datenübertragung von den Registriereinheiten	4
physikalische Übertragung der Vermessungsprogramme	1,4
precompiler	76
Parallele Übertragung	11
Parallele/Serielle Übertragung	11
Parser-Generator	73
Pascal für Anfänger	113
Physikalische Übertragung	8
Principles of Programming Languages: Design, Evaluatio»»»»	113
Problem der Textbegrenzer	54
Programmbaum ändern	66
Programme mit mehreren Dateien	36
Programmieren in Assembler für die IBM Personal Computer	113
Programmiersprache FORTRAN	3
Programmiersprachen	2
Programming Languages	113
Progr.-Sprachen	2
Protokolle	13
rechtsassoziativ (Auswertung von rechts nach links) ...	74
reelle Zahlen in der Binärarithmetik	24
Rechnen	19
Reihenfolge der Einträge	53
Rekursion in Tabellen	54
Restricted Inverse Form Grammar (RIF)	75
semantische Bedeutung als Spezialfälle	64
semantische Regeln	69
signifikante Stellen	22
software handshake	14
strukturorientierte Transformation	80
syntaxgesteuerte Übersetzung	69
Scanner	67
Schlußwort	111
Schnittstellenhandbuch	113
Schnittstellen	15

SCAN-Sprache	76
SCAN-Übersetzer	76
Semantiksprache	69
Serielle Übertragung	11
Simplex	10
Situation	1
Softwareprotokoll	14
Software	1
Sonderzeichen	34
SOURCE-TO-SOURCE TRANSLATION	75
Speicherung einer reellen Zahl	40
Spezielle Bit-Kombinationen	17
Spezifikation von Scannern	70
Start/Stop-Verfahren	12
StrukturAnalyse	65
Strukturelle Analyse	65
Strukturorientierte Transformationen	61
Symbole	67
Synchrones Format	12
tokens	67
typisch geodätisches Problem	24
Tabellenprogrammierung	49
Tabelle	48
Taschenwörterbuch der Programmiersprachen ALGOL, FORTR»»»»	113
Testprogramm	28
Text-Veränderungen	53
Theory of parsing, translation and compiling	113
Top-Down	65
Transformation in ein anderes System	17
Transformation von Bits innerhalb einer Zeile	39
TRAFO - Kurzbeschreibung	80
TRAFO	80, 96, 103, 106, 107
TRANS - Kurzbeschreibung	78
TRANS	78
Turbo C (IBM Version)	114
Turbo Pascal 3.0	114
Turbo Pascal 4.0	114
Turbo Pascal 5.0	114
TURBO PASCAL(IBM-PC)	96
unvollständige Programm-Bäume	64
übersichtliche Aufstellung	49
Umlaute	34
Umschulung	3
Umstellung der Programme	1
Umstellung der Vermessungsprogramme	4
Überlauf	54
Übersetzerbau	113
Übersetzer-Generatoren	69
Übersetzungsprogrammen	69
Übersetzungsprogramm	67
Übertragungsprotokoll	13
Übertragungssteuerzeichen	13
verschiedene Handhabung der Zeichen	17
vor der Übertragung speichern	18
VAX SCAN	76, 114
Veränderung des gespeicherten Programm-Baumes	66
Verzeichnisstrukturen	36
Voll-Duplex	10
Vorzeichen in der Binärarithmetik	22

VS FORTRAN Language and Library Reference	114
V24Box	16
V.24	15
Weite	56
WILD-GRE4-→AGA-GEODAT	107
XON/XOFF-Protokoll	14
YACC-Compiler	73
YACC-Sprache	73
YACC	73
Zeichen ersetze	43
Zeichengruppen	44
Zeichenkombinationen	53
Zeichenorientierte Transfo	43
Zeichenrahmen	12
Zeichen	34
Zeilen bei Umstellungen	35
Zeilen	35
Ziel	3
Zusammenfassung	77
Zweierkomplement	22

Bisher erschienen:

- Heft 1 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1970 - 1973, Dezember 1973.
- Heft 2 EGGER-PERDICH-PLACH-WAGENSOMMERER, Taschenrechner HP 45 und HP 65, Programme und Anwendungen im Vermessungswesen, 1. Auflage, März 1974, Special Edition in English, Juli 1974, 2. verbesserte Auflage, November 1974.
- Heft 3 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1973 - 1974, September 1974.
- Heft 4 EGGER-PALFINGER-PERDICH-PLACH-WAGENSOMMERER, Tektronix-Taschenrechner TEK 31, Programmbibliothek für den Einsatz im Vermessungswesen, November 1974.
- Heft 5 K. LEDERSTEGGER, Die horizontale Isostasie und das isostatische Geoid, Februar 1975.
- Heft 6 F. REINHART, Katalog von FK4 Horrebow-Paaren für Breiten von +30 bis +60, Oktober 1975.
- Heft 7 Arbeiten aus dem Institut für Höhere Geodäsie, Wien, Dezember 1975.
- Heft 8 Veröffentlichungen des Instituts für Photogrammetrie zum XIII. Internationalen Kongreß für Photogrammetrie in Helsinki 1976, Wien, Juli 1976.
- Heft 9 W. PILLEWIZER, Felsdarstellung aus Orthophotos, Wien, Juni 1976.
- Heft 10 PERDICH-PLACH-WAGENSOMMERER, Der Einsatz des programmierbaren Taschenrechners Texas Instruments SR-52 mit Drucker PC100 in der ingenieurgeodätischen Rechentechnik, Wien, Mai 1976.
- Heft 11 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1974 - 1976, November 1976.
- Heft 12 Kartographische Vorträge der Geodätischen Informationstage 1976, Wien, Mai 1977.
- Heft 13 Veröffentlichung des Instituts für Photogrammetrie anlässlich des 80. Geburtstages von Prof. Dr.h.c. K. Neumaier, Wien, Januar 1978.
- Heft 14 L. MOLNAR, Self Checking Analytical Relative Orientation and Strip Formation, Wien, Dezember 1978.
- Heft 15 Veröffentlichung des Instituts für Landesvermessung anlässlich des 80. Geburtstages von Prof. Dr. Alois Bavir, Wien, Januar 1979.
- Heft 16 Kolloquium der Assistenten der Studienrichtung Vermessungswesen 1976 - 1979, Wien, November 1979.
- Heft 17 E. VOZIKIS, Die photographische Differentialumbildung gekrümmter Flächen mit Beispielen aus der Architekturbildmessung, Wien, Dezember 1979.
- Heft 18 Veröffentlichung des Instituts für Allgemeine Geodäsie anlässlich des 75. Geburtstages von Prof. Dipl. Ing. Dr. F. Hauer, Die Höhe des Großglockners, Wien, 1981.
- Heft 19 H. KAGER, Bündeltriangulation mit indirekt beobachteten Kreiszentren, Wien, April 1981.

- Heft 20 Kartographische Vorträge der Geodätischen Informationstage 1980, Wien, Mai 1982.
- Heft 21 Veröffentlichung des Instituts für Kartographie anlässlich des 70. Geburtstages von Prof. Dr. Wolfgang Pillewizer: Glaziologie und Kartographie, Wien, Dezember 1982.
- Heft 22 K. TEMPFLI, Genauigkeitsschätzung digitaler Höhenmodelle mittels Spektralanalyse, Wien, Mai 1982.
- Heft 23 E. CSAPLOVICS, Interpretation von Farbinfrarotbildern, Wien, November 1982.
- Heft 24 J. JANSKA, Rektifizierung von Multispektral-Scanneraufnahmen - Entwicklung und Erprobung eines EDV-Programms, Wien, Mai 1983.
- Heft 25 Zusammenfassungen der Diplomarbeiten, Dissertationen und Habilitationen an den geodätischen Instituten der TU Wien, Wien, November 1984.
- Heft 26 T. WUNDERLICH, Die voraussetzungsfreie Bestimmung von Refraktionswinkeln, Wien, August 1985.
- Heft 27 G. GERSTBACH (Hrsg.), Geowissenschaftliche/geotechnische Daten in Landinformationssystemen - Bedarf und Möglichkeiten in Österreich, Wien, Juni 1986.
- Heft 28 K. NOVAK, Orientierung von Amateuraufnahmen ohne Paßpunkte, Wien, August 1986.
- Heft 29 Veröffentlichung des Instituts für Landesvermessung und Ingenieurgeodäsie, Abteilung Ingenieurgeodäsie anlässlich des 80. Geburtstages von Prof. Dipl. Ing. Dr. F. Hauer, Wien, Oktober 1986.
- Heft 30 K.-H. ROCH, Über die Bedeutung dynamisch ermittelter Parameter für die Bestimmung von Gesteins- und Gebirgseigenschaften, Wien, Februar 1987.
- Heft 31 G. HE, Bildverbesserung mittels digitaler Filterung, Wien, April 1989.
- Heft 32 F. SCHLÖGELHOFER, Qualitäts- und Wirtschaftlichkeitsmodelle für die Ingenieurphotogrammetrie, Wien, April 1989.
- Heft 33 G. GERSTBACH (Hrsg.), Geowissenschaftliche/geotechnische Daten in Landinformationssystemen - Datenbestände und Datenaustausch in Österreich, Wien, Juni 1989.
- Heft 34 F. HOCHSTÖGER, Ein Beitrag zur Anwendung und Visualisierung digitaler Geländemodelle, Wien, Dezember 1989.
- Heft 35 R. WEBER, Lokale Schwerefeldmodellierung unter Berücksichtigung spektraler Methoden zur Geländereduktion, Wien, April 1990.