Numerische Optimierung als Werkzeug zur Ausgleichungsrechnung?

eine praktische Anleitung für MS®-Excel

R. Staiger

Der unbestrittene Marktführer unter den Tabellenkalkulationsprogrammen ist MS-Excel. Neben mehr als 330 Funktionen bietet Excel auch die Möglichkeit der Matrizenrechnung und verfügt mit dem Add-In SOLVER über ein leistungsfähiges Werkzeug zur numerischen Optimierung. Leider haben diese Funktionalitäten unter Ingenieuren bisher wenig Beachtung gefunden.
In diesem Artikel wird die Leistungsfähigkeit von Excel als Werkzeug zur Ausgleichungsrechnung untersucht. Dazu wird zunächst die 2D-Helmert-Transformation detailliert auf zwei verschiedenen Wegen berechnet. Es folgen numerische Nährungslösungen für die wichtigsten Aufgaben der Formbestimmung (Zylinder, Kugel, Ebene, Kreis und Raumgerade). Abschließend werden die Vor- und Nachteile numerischer Optimierungsverfahren diskutiert.

1 Einleitung

Zur Lösung von Ausgleichungsaufgaben gibt es eine Reihe von Lehrbüchern, die erschöpfend Auskunft über Berechnungsmethoden und anzuwendende Algorithmen geben. Wenig ist dort über Software und Rechenhilfsmittel zu lesen, die sich zur konkreten Berechnung anbieten. Grundsätzlich gibt es für Berechnungen verschiedene Möglichkeiten: steht kein fertiges Programm zur Verfügung, implementiert man entweder die klassisch geschlossene Lösung oder eine numerische Nährungslösung. Unabhängig vom Lösungsweg gilt es dann die Formeln entweder in einer Hochsprache wie C++, PASCAL oder JAVA oder in einem Computeralgebrasystem (CAS) wie MATHEMATICA, MATLAB, MATHCAD oder MUPAD zu programmieren.
In [6] wurde erst neulich ein Nährungsverfahren für die ebene Koordinatentransformation vollständig abgeleitet. Bedient sich der Anwender eines numerischen Optimierungswerkzeugs, werden diese theoretischen Ableitungen überflüssig und es genügt, den funktionalen Zusammenhang zwischen den Beobachtungen und den Unbekannten zu formulieren. Das „Finden“ der gewünschten Lösung übernimmt die Software.
MS®-Excel, der unbestrittene Marktführer unter den Tabellenkalkulationsprogrammen (TBK), kann durchaus auch als CAS betrachtet werden. Dies gilt insbesondere, wenn die Matrizenfunktionen und das als „SOLVER“ bezeichnete Modul zur Lösung komplexer Optimierungsaufgaben genutzt werden.
Im Folgenden wird gezeigt, wie unterschiedliche Ausgleichungsaufgaben schnell und anschaulich in Excel gelöst werden können. Am Beispiel der ebenen Koordinatentransformationen werden zunächst zwei grundsätzlich verschiedene Berechnungswege beschrieben:

– Mit den eingebauten Matrizenfunktionen wird die geschlossene klassische Lösung der ebenen Koordinatentransformation direkt in einem Tabellenblatt umgesetzt.
– Das Optimierungswerkzeug SOLVER wird für eine numerische Lösung genutzt. Dazu ist die Ausgleichungsaufgabe als Optimierungsaufgabe zu formulieren und zu lösen.

Daran anschließend wird die Flexibilität der numerischen Methode an verschiedenen Aufgaben zur Bestimmung geometrischer Flächen und Körper aufgezeigt.
Gewidmet ist dieser Artikel Prof. Harald Schlemmer zu seinem 65. Geburtstag. Auch wenn Ausgleichungsrechnung als solches nicht zu seinen Spezialgebieten gehört, so sind ihm der Bezug zur Praxis und die konkrete Anwendung von Forschung und ihren Ergebnissen immer ein großes Anliegen gewesen. Ebenso hat er sich für zeitgemäße Werkzeuge und einen anschaulichen und lebendigen Unterricht engagiert. In diesem Sinne hofft der Verfasser auch auf eine künftig verstärkte Nutzung der Tabellenkalkulation für Auswertungen geodätischer Messungen, nicht zuletzt in der Ausbildung.

2 Optimierungsaufgaben

Das Gebiet der mathematischen Optimierung (engl.: *mathematical programming*) in der angewandten Mathematik beschäftigt sich damit, optimale Parameter eines – meist komplexen – Systems zu finden. „Optimal“ bedeutet, dass eine Zielfunktion einen Extremwert (Minimum oder Maximum) oder einen konkreten Wert unter gegebenen Bedingungen annimmt. Optimierungsprobleme stellen sich in der Wirtschaftsmathematik, Statistik, Operations Research und generell in allen ingenieur-

wissenschaftlichen Disziplinen, in denen mit unbekannten Parametern gearbeitet wird.
Die Lösung einer Optimierungsaufgabe besteht im Wesentlichen aus zwei Schritten:
1. Zunächst ist die Problemstellung in ein mathematisches Modell abzubilden, welches die Zielfunktion in Abhängigkeit der variablen Parameter (Unbekannte) beschreibt. Oft liegen zusätzlich für die Variablen Einschränkungen vor, die meist in Form von Ungleichungen formuliert werden.
2. Liegt das Modell vor, wird im Rahmen der mathematischen Optimierung eine Lösung gesucht. Je nach funktionalem Zusammenhang werden dabei unterschiedliche Lösungsstrategien und Algorithmen angewendet.

In der Literatur wird zwischen linearer und nichtlinearer Optimierung unterschieden. Bei der linearen Optimierung (engl.: *linear programming*) besteht ein linearer Zusammenhang zwischen den Unbekannten und der Zielfunktion und die Nebenbedingungen bestehen nur aus linearen Ungleichungen. Das bekannteste Lösungsverfahren ist der Simplex-Algorithmus. Nach einer endlichen Anzahl von Rechenschritten gibt es entweder ein globales Extremum oder die Aufgabe ist nicht lösbar.
Sobald eine Funktion der Nebenbedingungen oder die Zielfunktion nicht linear sind, wird von nicht-linearer Optimierung gesprochen. Hier sind andere Algorithmen erforderlich und die Existenz einer Lösung ist nicht pauschal nachweisbar. Außerdem können neben globalen auch lokale Extrema und Sattelpunkte auftreten (vgl. Abb. 8).
Eine weitere wichtige Kategorie innerhalb der skalaren Optimierung ist die ganzzahlige oder kombinatorische Optimierung. Darunter werden Aufgaben der linearen und der nicht-linearen Optimierung verstanden, bei denen einige oder alle Variable nur ganzzahlige Werte annehmen dürfen [1]. Bei vielen Aufgaben, z.B. aus der Produktionsplanung, sind nur ganzzahlige Lösungen sinnvoll, wie das folgende Beispiel zeigt [7].
Ein Produkt wird in festen Losgrößen gefertigt. Gesucht ist die optimale Losgröße unter folgenden Randbedingungen: es werden 100 000 Stück insgesamt im Jahr produziert. Die Initialkosten pro Los betragen 800 €, die Produktionskosten pro Stück 5 € und die Lagerhaltung kostet 1,25 €. Dazu wird angenommen, dass durchschnittlich die halbe Losgröße auf Lager liegt. Bezeichnen wir die Losgröße mit x, belaufen sich die Gesamtkosten auf

$$G(x) = \frac{100000}{x} \cdot 800 + 100000 \cdot 5 + \frac{x}{2} \cdot 1{,}25 \qquad (1)$$

In der graphischen Darstellung (Abb. 1) ist sofort zu erkennen, dass G(x) über ein globales Minimum verfügt. Das zugehörige Optimierungsverfahren ist nichtlinear, mit der Nebenbedingung dass nur ganzzahlige Lösungen zulässig sind. Bei einer Lösgröße von $x_{\min} = 11\,314$ entstehen minimale Gesamtkosten in Höhe von 514 142,14 €. Obwohl die Lösungsvielfalt bei der ganzzahligen Optimierung stark eingeschränkt ist, ist ihre Lösung nicht-trivial und sehr rechenaufwändig. Für obiges Beispiel ist es auch möglich, zunächst für die stetige und differenzierbare Zielfunktion (Gl. 1) eine reelle Lösung zu suchen,

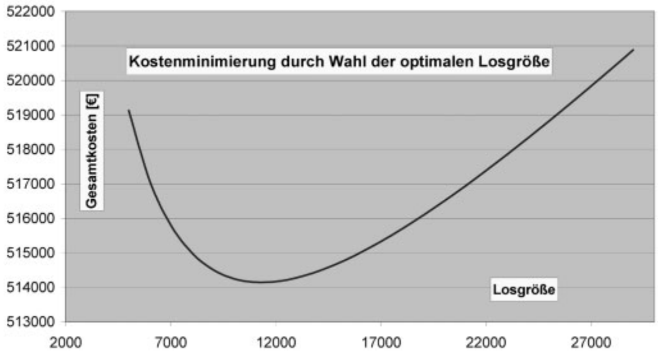


Abb. 1: Gesamtkosten in Abhängigkeit von der Losgröße

die anschließend zur nächsten ganzen Zahl gerundet wird. Diese Lösungsstrategie führt jedoch nicht immer zur korrekten Lösung [1, S. 366ff].

Ein Ausgleichungsproblem formuliert als Optimierungsaufgabe

Existiert für eine Ausgleichungsaufgabe, die z.B. nach der Methode der kleinsten Quadrate formuliert wurde, eine explizite Lösung, so verfügt zwangsläufig die Quadratsumme der Verbesserungen, ausgedrückt als Funktion der Beobachtungen und der Unbekannten, über ein globales Minimum, das auch mit numerischen Optimierungsverfahren gefunden werden kann. Aus diesem Grund wird hier die Frage nach der Existenz eines Minimums nicht weiter verfolgt. Für detaillierte Informationen hierzu wird auf die Literatur, z.B. [1], verwiesen.

3 Tabellenkalkulationsprogramme für ingenieurwissenschaftliche Anwendungen

Zwei Studenten der Harvard Business School, Dan Bricklin und Bob Frankston, gelten als Väter der Tabellenkalkulationsprogramme. 1978 entwickelten sie mit VISICALC ein Programm für den Apple II, das den Bildschirm in Zeilen und Spalten unterteilte und die Eingabe von Zahlen und Formeln in die Zellen ermöglichte. Erstmals wurden echte Tabellen generiert, die mit 254 Zeilen und 64 Spalten genügend Platz für Business-Kalkulationen aller Art boten. Die Grundfunktionalität eines Tabellenkalkulationsprogramms besteht aus dem „Rechnen in Tabellen“ und der graphischen Darstellungsmöglichkeiten der Mess- und Rechenergebnisse.
Excel ist aufgrund seiner Marktanteile ein Synonym für alle Tabellenkalkulationsprogramme geworden. Andere Produkte wie STARCALC, QUATTRO PRO oder OPENCALC spielen nur eine untergeordnete Rolle.
Neben 330 eingebauten Funktionen bietet Excel, die Möglichkeit der Matrizenrechnung (inkl. Matrizeninversion), einen Makrorekorder sowie eine komplette Programmierumgebung (Visual Basic for Applications). In den letzten 10 Jahren (seit Excel 97) ist der Funktionsumfang von Excel praktisch nicht verändert und nur unwesentlich erweitert worden. Mit EXCEL 2003 wurde die maximale Größe eines Tabellenblattes auf 256 Spalten

und 65 536 Zeilen erweitert; in der aktuellen Version (EXCEL 2007) setzt sich ein Tabellenblatt aus maximal 16 384 Spalten und 1 048 576 Zeilen zusammen. Ein Teil der Funktionen (ca. 220) werden dem Excel-Nutzer standardmäßig zur Verfügung gestellt, während der Rest in einer Vielzahl von Analyse-Funktionen (sog. Add-Ins) steckt, die zwar zum Lieferumfang gehören, jedoch extra geladen werden müssen (vgl. Kap. 4.2).
Viele Ingenieure und Naturwissenschaftler belächeln noch heute – wie in den Anfangstagen der Tabellenkalkulation – Excel und andere TBK-Programme und sehen sie vornehmlich als Werkzeug für Business-Berechnungen (Betriebswirtschaft, Controlling, Zeiterfassung, usw.). Die Verwendungsmöglichkeiten für ingenieurwissenschaftliche Fragestellungen werden dabei jedoch oft unterschätzt.

3.1 Matrizenfunktionen in Excel

In Excel gibt es vier eingebaute Funktionen für Matrizenoperationen (Tab. 1).
Praktischer Hinweis zum Gebrauch: die Matrizenfunktionen sind nicht mit der „EINGABE-Taste, sondern mit der Tastenkombination STRG + UMSCHALT + EINGABE zu übergeben; vorher ist die Ergebnismatrix exakt im Tabellenblatt zu markieren.

3.2 Die Add-In-Funktion „SOLVER“

Frontline Systems Inc. entwickelte das SOLVER-Programm für MS®-Excel. Nach eigenen Angaben wurden seit 1990 ca. 440 Millionen Lizenzen Excel einschließlich SOLVER verkauft. Daneben werden separat leistungsstärkere Versionen des SOLVER angeboten, die entweder mit Excel oder im Zusammenhang mit einer eigener Programmierung (SDK-Kit für VB.net, C++, JAVA oder MATLAB) genutzt werden [3]. Die technischen Merkmale der verschiedenen Versionen sind in Tab. 2 aufgeführt.
Die beiden Premium-Versionen sind schneller und bieten eine Plattform für umfangreichere Berechnungen. Zusätzlich sind die internen Lösungsstrategien vielfältiger und die Ergebnisse werden detaillierter beschrieben und dokumentiert. Die folgenden Betrachtungen beschränken sich auf die Standardversion, welche zur Berechnung der Beispiele völlig ausreicht.

Funktionalität und Schätzmethoden von SOLVER

Grundsätzlich gilt es ein Optimierungsproblem zu formulieren, indem ein mathematisches Modell über Excel-Formeln und Zellbezüge definiert wird. Das Modell setzt sich dabei aus verschiedenen Komponenten zusammen:

das Optimierungsziel: eine Zelle, deren Inhalt es zu optimieren gilt, d.h. die Zelle soll einen maximalen, minimalen oder vorgegebenen Wert annehmen.

die Variablen: variable Größen zum Erreichen des Optimierungszieles.

die Bedingungen: beschreiben funktionale Abhängigkeiten zwischen den Variablen oder einschränkende Bedingungen für die Variablen.

Tab. 1: Matrixfunktionen von MS®-Excel

Funktion	Beschreibung	Aufruf
MDET	Determinante einer quadratischen Matrix	{= MDET(*A1:D4*)}
MINV	Inverse einer quadratischen Matrix	{= MINV(*A1:D4*)}
MMULT	Multiplikation zweier Matrizen	{= MMULT(*A1:D4;B10:C13*)}
MTRANS	Transponierte einer Matrix	{= MTRANS(*A1:A4*)}

Tab. 2: Technische Merkmale der verschiedenen Produkte

Funktionalität und Leistungsfähigkeit	Standard Excel Solver	Premium Solver	Premium Solver Platform
Implementierte Lösungsstrategien			
Simplex Linear Solver	Nur LP[1]	Nur LP	LP/Quadratisch[2]
GRG[3] Nonlinear Solver	Ja	Ja	Ja (schneller)
Mixed-Integer Solver	Ja	Ja (schnell)	Ja (schneller)
Problemgröße			
Lineare Variable × Bedingungen	200 × 200	2000 × 1000	8000 × 8000
Nichtlineare Variable x Bedingungen	200 × 100	400 × 200	500 × 250
Ganzzahlige Einschränkungen für Variable	200	1000	2000

[1]LP = Lineare Optimierung
[2]Quadratisch nur hinsichtlich der Zielfunktion
[3]GRG2 = General Reduced Gradient-Algorithmus

In SOLVER sind unterschiedliche Lösungsstrategien für lineare, nicht-lineare und Aufgaben mit ganzzahligen Einschränkungen implementiert. Hinter der Lösung linearer Aufgabenstellungen steht ein SIMPLEX-Algorithmus, während die nicht-linearen Aufgaben mit der „General-Reduced-Gradient-Methode (GRG2)“ gelöst werden [4]. Zur ganzzahligen Optimierung wird die „Branch and Bound“-Methode herangezogen. Dieser Begriff kann am besten mit Methode des „Verzweigens und Begrenzens“ übersetzt werden. Es handelt sich um ein Entscheidungsbaumverfahren, das in seinem Verlauf immer wieder auf den SIMPLEX- und GRG2-Algorithmus zurückgreift. Aufgrund des expotentiell steigenden Rechenaufwandes rät der Hersteller bei den ganzzahligen Einschränkungen von mehr als 20 Unbekannten ab.

4 Die 2D-Helmert-Transformation

Am Beispiel einer zweidimensionalen Helmert-Transformation wird die Eignung von Excel für geodätische Berechnungen untersucht. Dazu wird zunächst die explizite Lösung mit Hilfe der Matrizenfunktionen berechnet (Kap. 4.1). Anschließend wird dieselbe Aufgabe als allgemeines Optimierungsproblem interpretiert und mit SOLVER numerisch gelöst (Kap. 4.2).

4.1 Strenge Lösung der 2D-Helmert-Transformation

Die Darstellung der 2D-Helmert-Transformation erfolgt nach [5, S. 315 ff]. Der funktionale Zusammenhang zwischen zwei rechtwinkligen Koordinatensystemen lautet:

$$\begin{aligned} X &= a + cx - dy \\ Y &= b + dx + cy \end{aligned} \tag{2}$$

Mit *X, Y* werden die Koordinaten des Zielsystems bezeichnet, mit *x, y* die des Startsystems. Die Unbekannten sind *a, b, c*, und *d*, aus denen sich der Unbekanntenvektor $\mathbf{x}_t$ zusammensetzt. Er beinhaltet zwei Verschiebungen (X_0, Y_0), eine Drehung (α) und ein Maßstabsfaktor (m).

$$\mathbf{x}_t = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} X_o \\ Y_0 \\ m \cdot \cos\alpha \\ m \cdot \sin\alpha \end{pmatrix} \tag{3}$$

Die Koordinaten im Zielsystem werden klassisch als „Beobachtungen“ $\mathbf{l}$ aufgefasst, während die Koordinaten des Startsystems die Koeffizientenmatrix $\mathbf{A}$ bilden.

$$\mathbf{A}_{2p,4)} = \begin{bmatrix} 1 & 0 & x_1 & -y_1 \\ 0 & 1 & y_1 & x_1 \\ 1 & 0 & x_2 & -y_2 \\ 0 & 1 & y_2 & x_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_p & -y_p \\ 0 & 1 & y_p & x_p \end{bmatrix} \tag{4}$$

$$\mathbf{l}_{(2p,1)} = \begin{bmatrix} X_1 \\ y_1 \\ X_2 \\ Y_2 \\ \vdots \\ X_p \\ Y_p \end{bmatrix} \tag{5}$$

Die unbekannten Transformationsparameter $\mathbf{x}_t$ und die Verbesserungen $\mathbf{v}_t$, die Restklaffungen in den identischen Punkten, werden wie folgt geschätzt:

$$\hat{\mathbf{x}}_t = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{l} \tag{6}$$

$$\mathbf{v}_t = \mathbf{A}\hat{\mathbf{x}}_t - \mathbf{l} \tag{7}$$

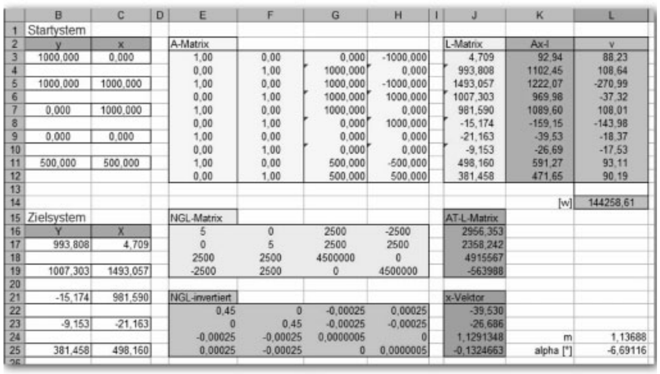
Dieses Modell der Helmert-Transformation ist linear, weshalb zur Berechnung keine Näherungswerte erforderlich sind. Werden die Koordinaten des Startsystems auf ihren Schwerpunkt bezogen, entartet die Normalgleichungsmatrix zu einer Diagonalmatrix: der Lösungsvektor kann für diesen Fall explizit und numerisch günstig angegeben werden. Darauf wird hier absichtlich verzichtet.

Das folgende Datenbeispiel mit fünf identischen Punkten ist synthetischer Natur und entstammt [2, Beispiel A]. Die Koordinaten wurden jedoch mit 1000 multipliziert. Ursache für die ungewöhnlich großen Verbesserungen sind grobe Fehler in den Koordinaten (vgl. Kap. 6).

Aus Gründen der Übersichtlichkeit wird die Berechnung in Excel mit „benannten Bereichen“ durchgeführt. Die A-Matrix und der Beobachtungsvektor $\mathbf{l}$ (*A, Lvek*) sind manuell zu erstellen. Mit den aufgelisteten Matrizenoperationen

{= MMULT(MTRANS(*A*; *A*)}
Berechnung der Normalgleichungsmatrix *NGL*
{= MINV(*NGL*)}
Inverse der Normalgleichungsmatrix *NGLINV*
{= MMULT(MTRANS(*A; Lvek*)}
Absolutgliedvektor *ATLvek*
{= MMULT(NGLINV; *ATLvek*)}
Unbekanntenvektor *xdach*

kann der Unbekanntenvektor bestimmt werden (Abb. 2, Zellen J22 bis J25).



	B	C	D	E	F	G	H	I	J	K	L
1	Startystem										
2	y	x		A-Matrix					L-Matrix	Ax-l	v
3	1000,000	0,000		1,00	0,00	0,000	-1000,000		4,709	92,94	88,23
4				0,00	1,00	1000,000	0,000		993,808	1102,45	108,64
5	1000,000	1000,000		1,00	0,00	1000,000	-1000,000		1493,057	1222,07	-270,99
6				0,00	1,00	1000,000	1000,000		1007,303	969,98	-37,32
7	0,000	1000,000		1,00	0,00	1000,000	0,000		981,590	1089,60	108,01
8				0,00	1,00	0,000	1000,000		-15,174	-159,15	-143,98
9	0,000	0,000		1,00	0,00	0,000	0,000		-21,163	-39,53	-18,37
10				0,00	1,00	0,000	0,000		-9,153	-26,69	-17,53
11	500,000	500,000		1,00	0,00	500,000	-500,000		498,160	591,27	93,11
12				0,00	1,00	500,000	500,000		381,458	471,65	90,19
13											
14										[w]	144258,61
15	Zielsystem			NGL-Matrix					AT-L-Matrix		
16	Y	X		5	0	2500	-2500		2956,353		
17	993,808	4,709		0	5	2500	2500		2368,242		
18				2500	2500	4500000	0		4915567		
19	1007,303	1493,057		-2500	2500	0	4500000		-563988		
20											
21	-15,174	981,590		NGL-invertiert					x-Vektor		
22				0,45	0	-0,00025	0,00025		-39,530		
23	-9,153	-21,163		0	0,45	-0,00025	-0,00025		-26,686		
24				-0,00025	-0,00025	0,0000005	0		1,1291348	m	1,13688
25	381,458	498,160		0,00025	-0,00025	0	0,0000005		-0,1324663	alpha [°]	-6,69116

Abb. 2: Berechnung der 2D-Helmert-Transformation

4.2 Numerische Lösung der 2D-Helmert-Transformation mit SOLVER

Am Beispiel der 2D-Helmert-Transformation wird der Lösungsansatz mit SOLVER detailliert erläutert. Im Prinzip sind die Verbesserungen als Funktion der Unbekannten und Beobachtungen in Form von Zellbezügen auszudrücken. Der Optimierungsalgorithmus modifiziert die variablen Zellen (Unbekannte) solange, bis das Optimierungsziel (hier das Minimum der Quadratsumme der Verbesserungen) erreicht ist. Am praktischen Beispiel sei dies verdeutlicht:

Schritt 0: Der SOLVER ist einmalig zu laden. In dem Menü *EXTRAS-Add-Ins* bei SOLVER ein Häkchen setzen und mit OK bestätigen.

Schritt 1: In den Spalten *B* und *C* sind die Koordinaten der 5 identischen Punkte im Start- und Zielsystem einzugeben (Abb. 3a). Die Zellen *I2* bis *I5* sind Platzhalter für die Unbekannten. In Spalte *E* werden die Transformationsgleichungen in Abhängigkeit der Unbekannten und der Koordinaten im Startsystem programmiert (vgl. Abb. 3b, c). Die Differenzen zu den Koordinaten des Zielsystems stehen in Spalte *F*. Schließlich wird daraus in Zelle *F16* die Quadratsumme der Koordinatenabweichungen formuliert.

Schritt 2: Mit markiertem Feld *F16* wird unter *EXTRAS-SOLVER* der SOLVER aufgerufen (Abb. 4a). Der Optimierungsroutine sind nun das Optimierungsziel (Zielzelle) sowie die Unbekannten (veränderbare Zellen) bekannt zu geben. Durch die bestehende Markierung von Zelle *F16* ist lediglich noch der Zielwert („minimal“) zu wählen.

Schritt 3: Die veränderbaren Zellen (Unbekannte) sind festzulegen. Alle anderen Zellen werden im Rahmen der Optimierung als unveränderlich behandelt.

Schritt 4: Zusätzlich können unter *Hinzufügen* (Abb. 4b) zusätzliche Nebenbedingungen (Einschränkungen) für

	A	B	C	D	E	F	G	H	I
1		Startsystem							
2		y	x		YT	dY		dx	0,00
3	1	1000,000	0,000		1000,000	6,192		dy	0,00
4	2	1000,000	1000,000		1000,000	-7,303		mcosa	1,0000000
5	3	0,000	1000,000		0,000	15,174		msina	0,0000000
6	4	0,000	0,000		0,000	9,153			
7	5	500,000	500,000		500,000	118,542			
8		Zielsystem						alpha	0,0000
9		Y	X		XT	dX		m	1,00000
10	1	993,808	4,709		0,000	-4,709			
11	2	1007,303	1493,057		1000,000	-493,057			
12	3	-15,174	981,590		1000,000	18,410			
13	4	-9,153	-21,163		0,000	21,163			
14	5	381,458	498,160		500,000	1,840			
15									
16					L2-Norm	258375,47			
17									

Abb. 3a: Tabellenblatt in Normaldarstellung mit groben Nährungswerten

	A	B	C	D	E	F	G	H	I
1		Startsystem							
2		y	x		YT	dY			dx
3	1	14029,64	12786,84		=$I$3+$I$5*C3+$I$4*B3	=E3-B9			dy
4	2	14914,63	12535,56		=$I$3+$I$5*C4+$I$4*B4	=E4-B10			mcosa
5	3	14771,83	11404,66		=$I$3+$I$5*C5+$I$4*B5	=E5-B11			msina
6	4	13221,62	11840,32		=$I$3+$I$5*C6+$I$4*B6	=E6-B12			
7		Zielsystem						alpha	=ARCTAN(I5/I4)
8		Y	X		XT	dX		m	=WURZEL(I4^2+I5^2)
9	1	19405,518	23159,823		=$I$2+$I$4*C3-$I$5*B3	=E9-C9			
10	2	20291,232	22909,817		=$I$2+$I$4*C4-$I$5*B4	=E10-C10			
11	3	20150,035	21778,202		=$I$2+$I$4*C5-$I$5*B5	=E11-C11			
12	4	18598,55	22211,755		=$I$2+$I$4*C6-$I$5*B6	=E12-C12			
13									
14					L2-Norm	=QUADRATESUMME(F3:F6;F9:F12)			

Abb. 3b: Tabellenblatt in Formeldarstellung

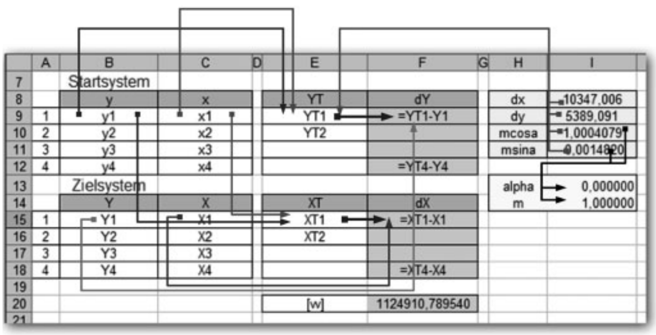


Abb. 3c: Abhängigkeiten der einzelnen Zellen

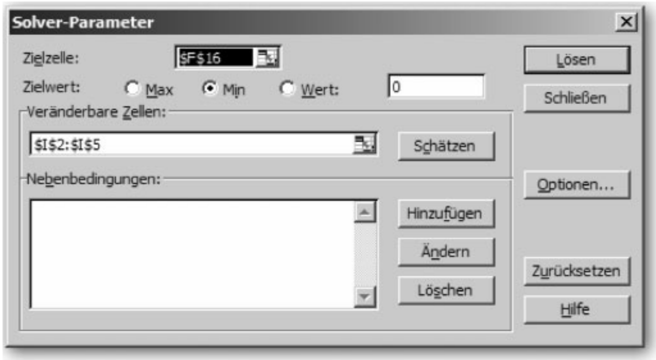


Abb. 4a: Hauptfenster des Optimierungswerkzeugs SOLVER

Abb. 4b: Zusätzliche Nebenbedingungen

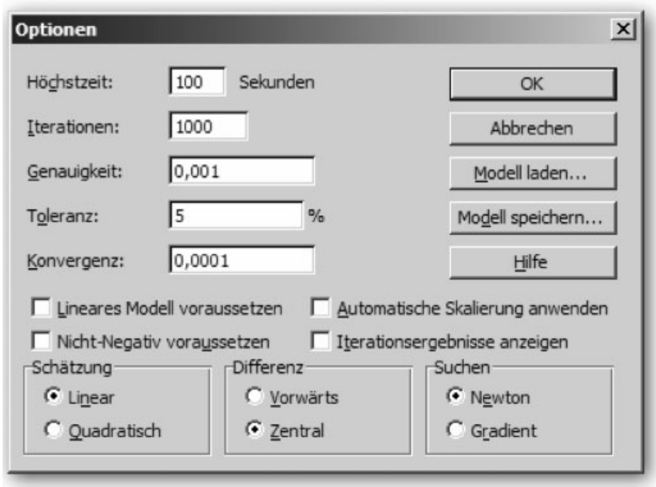


Abb. 4c: Berechnungsoptionen des SOLVER mit den Standardeinstellungen

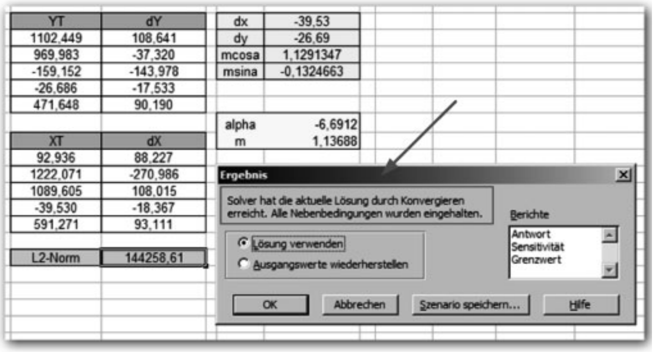


Abb. 4d: Lösungsvorschlag von SOLVER

bzw. zwischen den Unbekannten und Beobachtungen formuliert werden.
Wählbar sind die Relationen „Größer gleich“, „Kleiner gleich“, „Gleich“, sowie die Restriktionen „Ganzzahlig“ oder „binär“ für die Unbekannten. (Anm.: für betriebswirtschaftliche Optimierungsaufgaben ist die Lösungsmenge oft ganzzahlig, z.B. Anzahl Pakete, Maschinen usw.).
Schritt 5: In dem Menü *Optionen…* (Abb. 4c) können unterschiedliche Parameter wie Anzahl der Iterationen, Genauigkeit, usw. eingestellt werden. Nur wenn bei „Lineares Modell voraussetzen“ ein Häkchen gesetzt ist, startet der Algorithmus intern mit dem linearen Lösungsansatz. Ansonsten geht die Strategie zunächst von einer nicht-linearen Beziehung zwischen Beobachtungen und Unbekannten aus [3].
Schritt 6: Im Hauptfenster des SOLVER wird durch Drücken der Taste *Lösen* der Optimierungsprozess gestartet.
Als Antwort werden die gelösten Unbekannten und der erreichte Zielwert in die Zellen eingetragen. Der Nutzer kann die Lösung annehmen oder aber die Zellinhalte werden auf ihre Startwerte zurückgesetzt.
Schritt 7: Da es sich um ein numerisches „Black-Box-Verfahren“ handelt, ist es unbedingt erforderlich das Ergebnis auf Plausibilität und Gültigkeit zu prüfen (siehe Kap. 7).

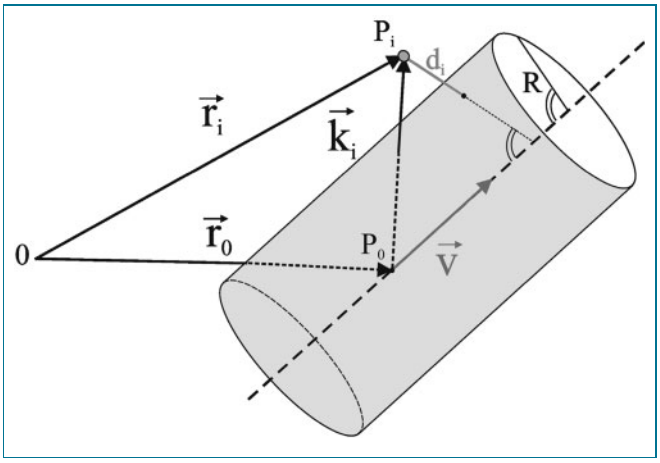


Abb. 5: Abstand eine Punktes P_i vom Zylindermantel mit Radius R

Auf beiden Lösungswegen entstehen identische Ergebnisse (siehe Abb. 2 und 4d), die auch mit den Ergebnissen aus [2] übereinstimmen.

5 Berechnung von Flächen und Körpern

Die Auswertung flächenhafter Messverfahren (z.B. Laserscanning und Lasertracking) oder taktiler Messsysteme (z. B. Messarme und Koordinatenmessmaschinen) basiert auf der Berechnung ausgleichender Flächen und Körper. Zur Verifizierung der Ergebnisse, z. B. einer Laserscanning-Auswertesoftware, ist eine eigene und transparente Berechnungsmöglichkeit sehr nützlich. Im Folgenden wird der Weg zur unabhängigen Berechnung der Basiselemente Ebene, Kugel und Zylinder aufgezeigt.
Der expliziten Berechnung dieser Regelflächen und -körper liegt der Allgemeinfall der Ausgleichungsrechnung, das Gauß-Helmert-Modell, zugrunde, denn in der Regel kann eine einzelne Beobachtung (hier Koordinate) nicht allein als Funktion der Unbekannten ausgedrückt werden. Für den numerischen Lösungsweg mit SOLVER (oder vergleichbaren Software-Lösungen) ist das nicht von Bedeutung.

5.1 Berechnung eines Zylinders

Ein Zylinder ist eindeutig festgelegt, durch seine räumliche Achse und den Zylinderradius R. Die Zylinderachse wird beschrieben mit einem willkürlichen Aufpunkt P_0 und dem Richtungsvektor $\vec{v}$ beliebiger Länge. Der räumliche Abstand d_i eines Punktes P_i von dem Zylindermantel beträgt:

$$d_i = \frac{|\vec{v} \times (\vec{r}_i - \vec{r}_0)|}{|\vec{v}|} - R \tag{8a}$$

Abb. 6a: Tabellenblatt zur Berechnung eines ausgleichenden Zylinders

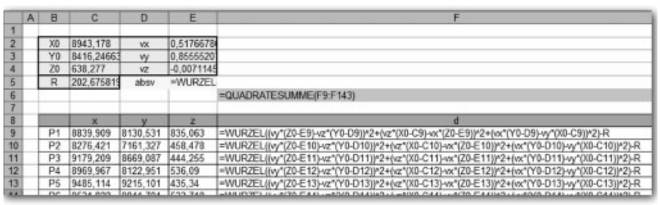


Abb. 6b: Formelbezüge zur Berechnung eines ausgleichenden Zylinders

oder in Komponentenschreibweise mit

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \text{ und } (\vec{r}_i - \vec{r}_0) = \vec{k}_i = \begin{pmatrix} k_x \\ k_y \\ k_z \end{pmatrix}$$

$$\text{gilt } d_i = \frac{\sqrt{(v_y \cdot k_z - v_z \cdot k_y)^2 + (v_x \cdot k_z - v_z \cdot k_x)^2 + (v_x \cdot k_y - v_y \cdot k_x)^2}}{\sqrt{v_x^2 + v_y^2 + v_z^2}} - R \qquad (8b)$$

Für jeden beobachteten Zylinderpunkt $P_i(x_i, y_i, z_i)$ wird eine Abstandsgleichung formuliert und die Quadratsumme aller Abstände mit SOLVER minimiert.

Die Anzahl der Freiheitsgrade beträgt in dieser Darstellung 7 (1 Aufpunkt, 1 Richtungsvektor, 1 Radius), jedoch müssen für eine eindeutige Lösung zusätzlich drei Bedingungsgleichungen eingeführt werden. Die Vieldeutigkeit des Aufpunktes wird aufgehoben, indem 2 von 3 Koordinaten vorgegeben werden. Analog ist es sinnvoll den Richtungsvektor der Zylinderachse zu normalisieren. Damit sind nur 4 der 7 gesuchten Parameter unabhängig. Pro Messpunkt liegen 3 Beobachtungen vor und zur geometrischen Festlegung des Zylinders sind mind. 4 Punkte zu beobachten, die nicht in einer Ebene liegen dürfen.

In Abb. 6a ist das Tabellenblatt zur Zylinderberechnung dargestellt. Insgesamt liegen 136 Beobachtungen vor. In Zelle *F9* bis *F145* ist der Abstand d in Abhängigkeit von X_0, Y_0, Z_0, v_x, v_y, v_z und R gemäß (Gl. 8b) verankert (Abb. 6b). Zelle *F6* addiert die Abstandsquadrate.

Die Bedingungen werden wie folgt realisiert: 2 von 3 Koordinaten (hier X_0 und Z_0) werden fest vorgegeben und nicht als variabel definiert. Die Länge des Achsvektors wird in Zelle *E5* explizit berechnet und unter den SOLVER-Optionen wird für diese Zelle die Bedingung = 1,00 eingeführt.

Die generelle Flexibilität der numerischen Lösung wird am Beispiel des Zylinders ersichtlich:

– Sind, bedingt durch die Aufgabenstellung, einer oder mehrere Parameter bekannt, z.B. der Radius R, so wird die bekannte Größe in die Zelle eingetragen und der Zelle ist – vor dem SOLVER-Aufruf – der Variablen-Status zu entziehen.
– Der Sonderfall des Zylinders ist die räumliche Gerade. Dazu muss nur der Radius R fest auf „0“ gesetzt werden und es kann eine ausgleichende Gerade berechnet werden, ohne die Beobachtungsgleichungen zu verändern.

Für die Kugel und die Ebene werden nur noch die Beobachtungsgleichungen diskutiert. Die Umsetzung kann gemäß den obigen Ausführungen erfolgen.

5.2 Berechnung einer ausgleichenden Kugel

Eine Kugel wird vollständig beschrieben durch ihre Mittelpunktkoordinaten $M\ (X_M,\ Y_M,\ Z_M)$ und den Radius R. Der Abstand d_i eines Punktes P_i zu der Kugeloberfläche beträgt:

$$d_i = \sqrt{(x_i - X_M)^2 + (y_i - Y_M)^2 + (z_i - Z_M)^2} - R \qquad (9)$$

Der erwähnenswerte Sonderfall ist hier der Kreis. Dazu wird lediglich auf die dritte Dimension verzichtet und die Abstandsgleichung vereinfacht sich entsprechend.

5.3 Berechnung einer ausgleichenden Ebene

Alle Punkte $P_i(x_i,\ y_i,\ z_i)$, welche die Gleichung

$$a \cdot x + b \cdot y + c \cdot z - 1 = 0 \qquad (10a)$$

erfüllen, liegen in einer Ebene E, welche durch die Parameter a, b und c festgelegt ist. Der Abstand d_i eines Punktes P_i zur Ebene lautet:

$$d_i = \frac{a \cdot x_i + b \cdot y_i + c \cdot z_i - 1}{\sqrt{a^2 + b^2 + c^2}} \qquad (10b)$$

6 Ausgleichung nach der L1-Norm

Zur Suche grober Fehler ist einer Ausgleichung nach der L1-Norm (Minimierung der Absolutsumme der Verbesserungen) ideal. Für eine numerische Lösung muss dazu lediglich die Zielfunktion geändert werden. An die Stelle der Quadratsumme der Verbesserungen tritt die Absolutsumme, die es wiederum zu minimieren gilt.

Das Beispiel zur 2D-Kordinatentransformation (Kap. 4) wird mit Excel nach der L1-Norm ausgeglichen. Die Ergebnisse stimmen mit denen aus [2] hinreichend überein und die Ausreißer werden eindeutig erkannt (Abb. 7, Spalte *G*).

Ob für die L1-Norm immer ein gültiges Ergebnis erzielt werden kann, sei hier ausdrücklich in Frage gestellt. Für die robuste Ausgleichung (L1-Norm) gibt es nicht immer Lösungen [2]; außerdem wird hier die Summe der Verbesserungen mit der Betragsfunktion beschrieben, die bekanntermaßen bei „0“ nicht differenzierbar ist. Stetigkeit und Differenzierbarkeit für die Zielfunktionen wurden bisher stillschweigend vorausgesetzt. Für jede kleinste-

	A	B	C	D	E	F	G	H	I	J	K
1		Startsystem									
2		y	x		YT	dY	Betrag dY		dx	-16,68	
3	1	1000,00	0,00		995,78	1,98	1,98		dy	-9,15	
4	2	1000,00	1000,00		974,39	-32,91	32,91		mcosa	1,0049372	
5	3	0,00	1000,00		-30,55	-15,37	15,37		msina	-0,0213928	
6	4	0,00	0,00		-9,15	0,00	0,00				
7	5	500,00	500,00		482,62	101,16	101,16				
8		Zielsystem							alpha	-1,2195	
9		Y	X		XT	dX	Betrag dX		m	1,00516	
10	1	993,81	4,71		4,71	0,00	0,00				
11	2	1007,30	1493,06		1009,65	-483,41	483,41				
12	3	-15,17	981,59		988,25	6,66	6,66				
13	4	-9,15	-21,16		-16,68	4,48	4,48				
14	5	381,46	498,16		496,48	-1,68	1,68				
15											
16						L1-Norm	647,65				
17											

Abb. 7: Ergebnisse der 2D-Helmertransformation nach der L1-Norm

Quadrate-Lösung ist das auch mit Sicherheit zumindest in „Lösungsnähe“ gegeben. Der Hersteller von SOLVER weist daraufhin, dass Stetigkeit und Differenzierbarkeit sehr nützlich und manchmal zwingend erforderlich sind. Leider wird dies nicht näher erläutert. So ist z.B. eine Zielfunktion mit der Einschränkung nur ganzzahlige Lösung streng genommen unstetig, aber trotzdem lösbar. Vermutlich wird dort der Weg der kombinatorischen Optimierung bestritten, indem alle in Frage kommenden Lösungen durchprobiert werden.

7 Erfahrungen und Empfehlungen

Die in SOLVER implementierten Optimierungsverfahren sind, wie alle vergleichbaren Produkte, Blackbox-Verfahren. Der Nutzer hat i.d.R. zuwenige Informationen zur Verfügung um zweifelsfrei zu beurteilen, ob das absolute Minimum nur annähernd oder hinreichend genau erreicht wurde, oder ob sogar nur ein lokales Nebenminimum oder ein Sattelpunkt gefunden wurde. Folgende Beispiele mögen dies verdeutlichen:
Gegeben sind zwei rationale Funktionen 4. Grades (Abb. 8). Funktion 1 weist einen Sattelpunkt bei $x = -1$ und ein absolutes Minimum für $x = 0$ auf. Funktion 2 besitzt ein lokales ($x \approx -2{,}2$) und ein absolutes Minimum ($x \approx -1{,}4$). Wird SOLVER mit einem Initialwert x_0 kleiner als das lokale Minimum bzw. der Sattelpunkt gestartet, stoppt der Algorithmus beim lokalen Maximum bzw. beim Sattelpunkt und gibt diese als Lösung an. Ist der Startwert größer, findet SOLVER jeweils das globale Minimum.
Findet SOLVER nicht das gesuchte globale Minimum kann dies auf unterschiedlichen Ursachen beruhen:
- Ein Nebenminimum/Sattelpunkt wurde gefunden.
- SOLVER bricht vorzeitig ab. Dafür gibt es im wesentlichen 4 Gründe:
  - Maximale Rechenzeit ist überschritten
  - Anzahl zulässiger Iterationen ist überschritten
  - Genauigkeitskriterium erfüllt. Wird das Kuhn-Tucker-Kriterium – eine mathematische Optimalitätsbedingung der nicht-linearen Optimierung - innerhalb der Genauigkeitsschranke eingehalten, gilt das Optimum als erreicht und SOLVER bricht mit der Erfolgsmeldung ab [3]. Für nicht-lineare Aufgaben (GRG2-Algorithmus) kann der Nutzer die zugehörige Vorgabe unter „Genauigkeit“ ändern (Abb. 4c).
  - Konvergenzkriterium erfüllt. Verbessert SOLVER die Zielfunktion innerhalb der letzten 5 Iterationen nicht um den unter „Konvergenz“ vorgegebenen Betrag erfolgt ein Abbruch. Das Konvergenzkriterium kann vorzeitig erfüllt sein, wenn die Zielfunktion in der Nähe des globalen Minimums sehr flach verläuft.

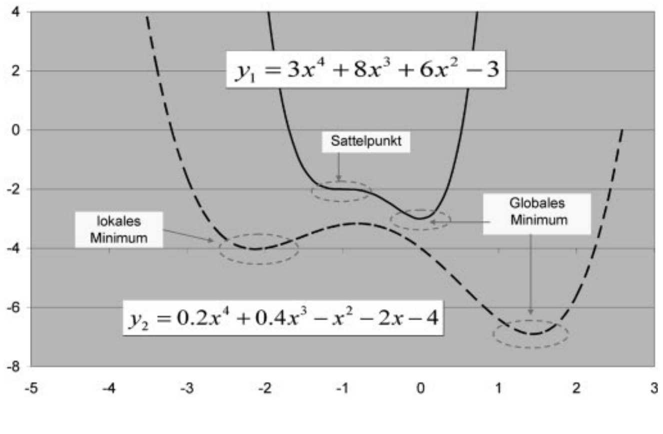


Abb. 8: Ganz-Rationale Funktion 4. Grades mit Nebenminimum

Findet SOLVER eine Lösung, ist entweder das Genauigkeits- oder das Konvergenzkriterium erfüllt. Die zugehörigen Meldungen lauten: „Solver hat eine Lösung gefunden“ oder „Solver hat eine Lösung durch Konvergieren gefunden“.

Empfehlungen

Die Standardeinstellungen für die Parameter „Genauigkeit“ und „Konvergenz“ sind für ingenieurwissenschaftliche Fragestellungen viel zu niedrig eingestellt. Nur deutlich engere Schranken (ca. 1E-8 und kleiner) führen zu befriedigenden Resultaten. Der Parameter „Toleranz“ ist nur für ganzzahlige Nebenbedingungen von Bedeutung. Damit wird die zulässige Abweichung von der Ganzzahligkeit beschrieben.
Es ist unbedingt erforderlich die Ergebnisse auf Plausibilität und Gültigkeit zu prüfen. Dazu ist ein theoretisches Verständnis für die Ausgleichungsrechnung und für die konkrete Aufgabenstellung erforderlich. Folgende Schritte werden empfohlen:
- Unbedingt die leicht zu übersehende Meldung im Ergebnisfenster (Abb. 4d rot umrahmt) beachten. SOLVER kennt an dieser Stelle verschiedene Antworten, 2 Erfolgsmeldungen und ca. 10 Fehlermeldungen.
- Lösung(en) mit unterschiedlichen Startwerten und geänderten Optionen herbeiführen und vergleichen. Die wichtigsten Parameter sind „Genauigkeit“ und „Konvergenz“ sowie die Option „mit automatischer Skalierung“. Diese ist immer sinnvoll, wenn sich Beobachtungen und / oder Unbekannte in mehrere Größenordnungen unterscheiden.
- Die Unbekannten sind im Tabellenblatt nach wie vor formelmäßig mit den Beobachtungen verknüpft. Durch eine manuelle Veränderung der Unbekannten ändert sich entsprechend die Zielfunktion (hier Quadratsumme der Verbesserungen). Somit kann stichprobenartig geprüft werden, ob es sich tatsächlich um die Lösung im Sinne der Zielfunktion handelt.
- Eine Minimallösung mit möglichst wenigen Beobachtungen gibt Aufschluss über genaue Startwerte. Diese Variante bietet sich auch an, bei Lösungen, die nicht vertrauenswürdig erscheinen bzw. generell bei einer großen Anzahl von Beobachtungen.
- Die Erfahrung zeigt, dass es zumindest numerische Nebenminima und Sattelpunkte gibt. Diese können nur mit geänderten Startwerten überwunden werden. Sinnvoll ist die Annäherung von verschiedenen Seiten.

8 Bewertung der numerischen Optimierung als Werkzeug zur Ausgleichungsrechnung

Die numerische Optimierungsstrategie hat, wie jedes andere Verfahren, Vor- und Nachteile, die es abzuwägen gilt.

Nachteile

– Ein großer, bisher noch nicht genannter Nachteil ist das Fehlen geodätischer Qualitätsmaße, wie Genauigkeit der Unbekannten und Funktionen davon oder Zuverlässigkeit. Eine numerische Sensitivitätsanalyse könnte hier Abhilfe schaffen.
– SOLVER steht für ein Black-Box-Verfahren.
– Es gibt keine Lösungssicherheit. Das Verfahren selbst liefert nur ungenügende Hinweise auf die Richtigkeit des Ergebnisses, da die meisten Aufgaben der Ausgleichungsrechnung zur nicht-linearen Optimierung zu zählen sind.

Vorteile

– Excel und SOLVER sind weltweit nahezu auf jedem PC installiert.
– Excel ist sehr verbreitet, auch als Ersatz für einen Taschenrechner. Viele Nutzer sind mit der Bedienoberfläche vertraut, wodurch eine zusätzliche Einarbeitungszeit entfällt.
– Keine Programmierung in einer Hochsprache erforderlich.
– Der Datenimport erfolgt einfach über die Zwischenablage oder den Importassistenten (ASCII-Datei)
– Die Suche nach groben Fehlern ist möglich (L1-Norm).

9 Zusammenfassung und Ausblick

In diesem Artikel wird aufgezeigt, wie mit dem Werkzeug der numerischen Optimierung auch Ausgleichungsaufgaben effektiv gelöst werden können. Für alle genannten Beispiele wurden Lösungen gefunden, die entweder mit den Ergebnissen aus der Literatur oder aus anderen Softwarelösungen hinreichend übereinstimmen. Dies gilt auch für weitere Beispiele, z.B. die 3D-Helmert-Transformation, die aus Platzgründen nicht erwähnt wurden.
Stillschweigend wurde immer von gleichgenauen Beobachtungen desselben Typs ausgegangen (Koordinaten). Solange die Beobachtungen unkorreliert sind, ist es ohne nennenswerten Mehraufwand möglich, Gewichte und unterschiedliche Beobachtungstypen einzuführen.
Zwei schwerwiegende Nachteile sind die fehlende Lösungssicherheit und der Mangel an geodätischen Qualitätsmaßen. Diese sind jedoch nicht Excel-spezifisch, sondern treten bei allen numerischen Softwarelösungen auf. Aufgrund dieser Nachteile kann eine numerische Optimierungssoftware nur mit mathematischem Sachverstand und fundierten Ausgleichungskenntnissen eigenständig genutzt werden. Sind diese Voraussetzungen jedoch gegeben, steht mit Excel ein preiswertes und leistungsfähiges Werkzeug praktisch auf jedem PC weltweit zur Verfügung, das bisher kaum gewürdigt wurde. Nach kurzer Einarbeitungszeit können einfache Ausgleichungsaufgaben mit den Matrizenfunktionen und mit SOLVER selbstständig gelöst werden.
Des Weiteren ist SOLVER eine effiziente Methode zur unabhängigen Überprüfung einer anderen Lösung, deren Ergebnisse nicht nachvollziehbar sind, oder für numerische Studien, z.B. um den Einfluss einzelner Beobachtungen oder Unbekannter zu quantifizieren.

The undoubted market leader for spreadsheet calculation is MS®-Excel. In addition to more then 300 functions Excel offers the possibility of matrix-functions and a powerful tool, called SOLVER, for mathematical programming. Unfortunately these functions are only rarely known among engineers.
In this article the capability of Excel as a tool for geodetic least-squares-adjustments is demonstrated and proofed. In a first step the calculation of 2D-coordinate transformations is explained and calculated in detail by two different and independent ways. Secondly different geometrical elements like cylinder, sphere, plane, circle and 3d-line are calculated by the efficient method of non-linear-programming with SOLVER.

10 Literatur

[1] [1] Benker, H.: Mathematische Optimierung mit Computeralgebrasystemen. Springer-Verlag, Heidelberg, 2002
[2] [2] Caspary, W.; Beineke, D.: Robuste Helmert-Transformation. Allgemeine Vermessungsnachrichten 7/2003, S. 242–247
[3] [3] Fylstra, D.; Lasdon, L.; Watson, J.; Waren, A.: Design and Use of the Microsoft Excel Solver“. INTERFACES, Vol. 28, No. 5, Sept–Oct 1998, pp. 29–55
[4] [4] Lasdon, L. S., Waren, A. D. (1978): Generalized reduced gradient software for linearly and nonlinearly constrained problems. in: Greenberg, H.J., (Ed.) Design and Implementation of Optimization Software. Sijthoff and Noordhoff, Holland, 1978, pp. 335–362
[5] [5] Niemeier, W.: Ausgleichungsrechnung. Walter de Gruyter-Verlag, Berlin, 2002
[6] [6] Späth, H.: Berechnung von Helmert-Transformationen mit dem Newton-Verfahren. Allgemeine Vermessungsnachrichten 5 /2007, S. 166–168
[7] [7] Troxell, D. S.: Optimization Software Pitfalls: Raising Awareness in the Classroom. INFORMS Transcations on Education Vol. 2 No 2, p 40–46, 2002

Anschrift des Verfassers:
Prof. Dr.-Ing. Rudolf Staiger,
Hochschule Bochum,
FB Vermessungswesen und Geoinformation,
Lennershofstr. 140, 44801 Bochum,
E-Mail: rudolf-staiger@fh-bochum.de