

Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 5560

KfK-Seminarreihe
"Aktuelle Forschungs-
gebiete in der Mathematik"
Seminarbeiträge 1994

C. P. Hugelmann, R. Seifert, T. Westermann
(Hrsg.)

Institut für Angewandte Informatik

April 1995

Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZK 5560

KfK-Seminarreihe
"Aktuelle Forschungsgebiete in der Mathematik"
Seminarbeiträge 1994

Herausgeber:
C. P. Hugelmann, R. Seifert, T. Westermann
Institut für Angewandte Informatik

Forschungszentrum Karlsruhe GmbH, Karlsruhe
1995

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe

ISSN 0947-8620

KfK-Seminarreihe: Aktuelle Forschungsgebiete in der Mathematik

Seminarbeiträge 1994

Zusammenfassung

1994 wurde im KfK die Seminarreihe über aktuelle Forschungsgebiete in der Mathematik fortgesetzt. Ziel dieser Seminarreihe ist, im KfK anwendungsorientierte, aktuelle Forschungsthemen aus der Mathematik zu präsentieren. Übersichtsvorträge sollen Einblicke in moderne Methoden und Verfahren der Mathematik ermöglichen. Die Vorträge standen daher wieder im engen Zusammenhang mit praxisbezogenen Anwendungen. Organisiert wurde die Seminarreihe von Claus-Peter Hugelmann (KfK, HDI), Rolf Seifert (KfK, IAI) und Thomas Westermann (FH Karlsruhe).

Im vorliegenden Bericht sind die Seminarbeiträge in schriftlicher Form zusammengefaßt.

KfK-Seminar Series on Selected Topics in Mathematics

Seminar reports 1994

Summary

In 1994 the KfK seminars on selected topics in applied mathematics were continued. The aim was to demonstrate the importance of applied mathematics and to present current research areas in mathematics. Survey lectures should give an insight in modern methods and methodologies. The seminars were organized by Claus-Peter Hugelmann (KfK, HDI), Rolf Seifert (KfK, IAI) and Thomas Westermann (FH Karlsruhe).

This report contains the collection of the seminar papers.

Inhaltsverzeichnis

Vorwort	5
Vorstellung	7
D. Ratz (Universität Karlsruhe)	9
Globale Optimierung mit Ergebnisverifikation	
Vorstellung	37
P. Mäder (Staatl. Seminar Freiburg)	39
Ein Überblick zur Geschichte der Zahl Null	
Vorstellung	57
H.R. Lerche, R. Sandvoß (Universität Freiburg)	59
Die Vorhersage und das Entdecken von Trendänderungen bei Finanzdaten	
Vorstellung	77
A. Tsodikov (St. Petersburg University, Rußland)	79
Nonparametric Estimation of a Survivor Function from incomplete Data	
Vorstellung	99
J. Weidner (IBM Heidelberg)	101
Rechnen in komplexen Geometrien	
Vorstellung	117
H. Bauer (FHT Reutlingen)	119
Computeralgebra und Ingenieurmathematik - Beispiele mit Maple	

Vorwort

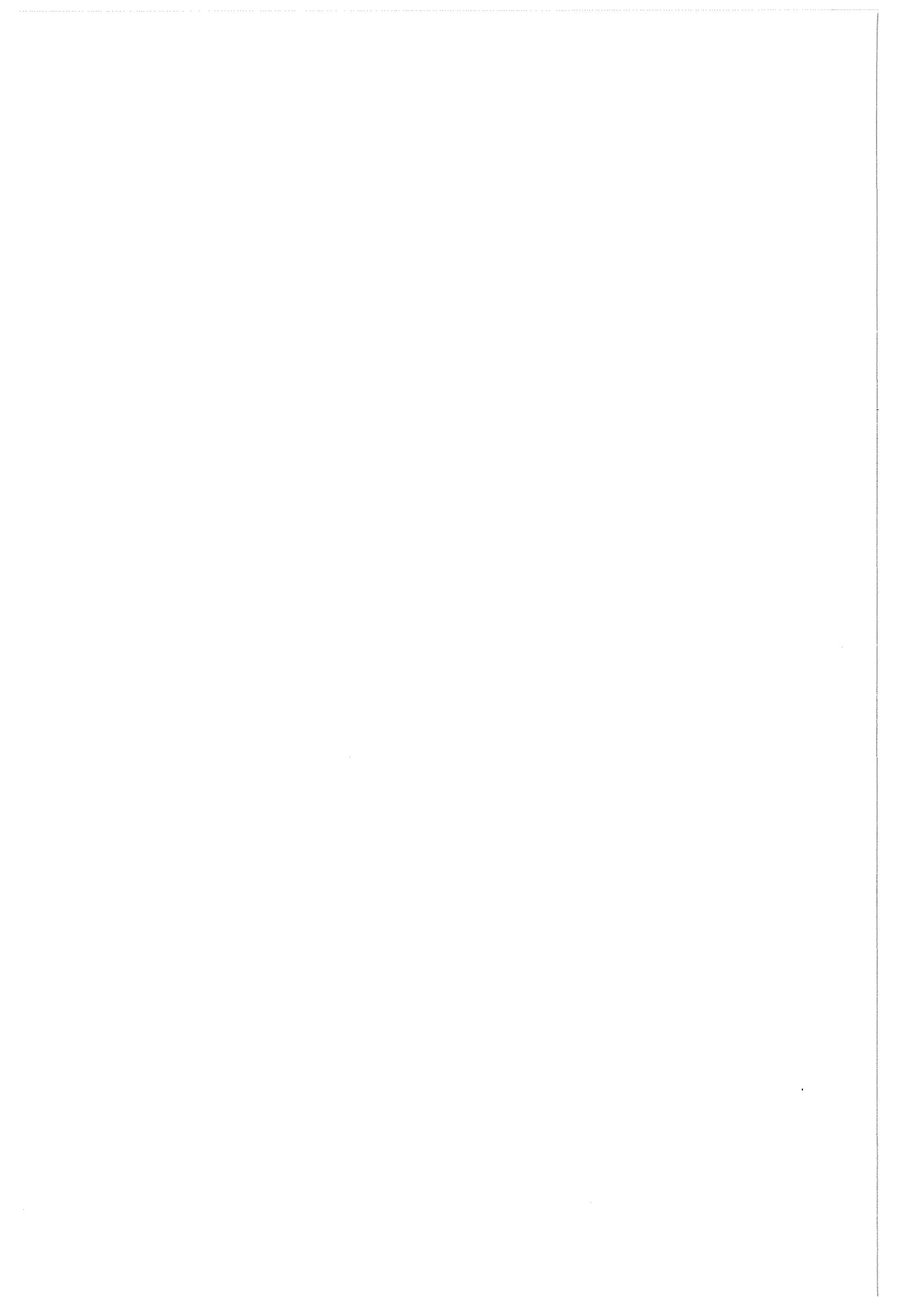
1994 wurde die Seminarreihe "Aktuelle Forschungsgebiete in der Mathematik - Praxisbezogene Anwendungen" fortgesetzt.

Ziel dieser Seminarreihe ist, im KfK anwendungsorientierte, aktuelle Forschungsthemen aus der Mathematik zu präsentieren. Gedacht ist an Übersichtsvorträge, die Einblicke in moderne Methoden und Verfahren der Mathematik ermöglichen. Mit dieser Seminarreihe soll auch der Kontakt zu externen Forschungseinrichtungen erweitert und vertieft werden, um weitere kompetente Wissenschaftler als Gesprächspartner für das Kernforschungszentrum zu gewinnen. Darüber hinaus soll aber auch ein breiter Gedankenaustausch innerhalb des KfK ermöglicht werden.

Die Zielgruppe für das Auditorium sind somit mathematisch-interessierte Mitarbeiter, die sich neuen mathematischen Verfahren und Methoden aufgeschlossen zeigen bzw. die selbst an mathematischen Fragestellungen arbeiten.

Im vorliegenden Bericht sind die Seminarbeiträge für 1994 in schriftlicher Form zusammengefaßt. Die Beiträge werden von den Herausgebern durch eine Vorstellung der Referenten eingeleitet.

Die Seminarreihe wird im Jahr 1995 fortgesetzt. Für Anregungen, Themen- und Vortragsvorschläge sind die Organisatoren Claus-Peter Hugelmann (HDI, Tel. 07247/824897), Rolf Seifert (IAI, Tel. 07247/824411) und Thomas Westermann (FH Karlsruhe, Tel. 0721/9251296) stets offen.



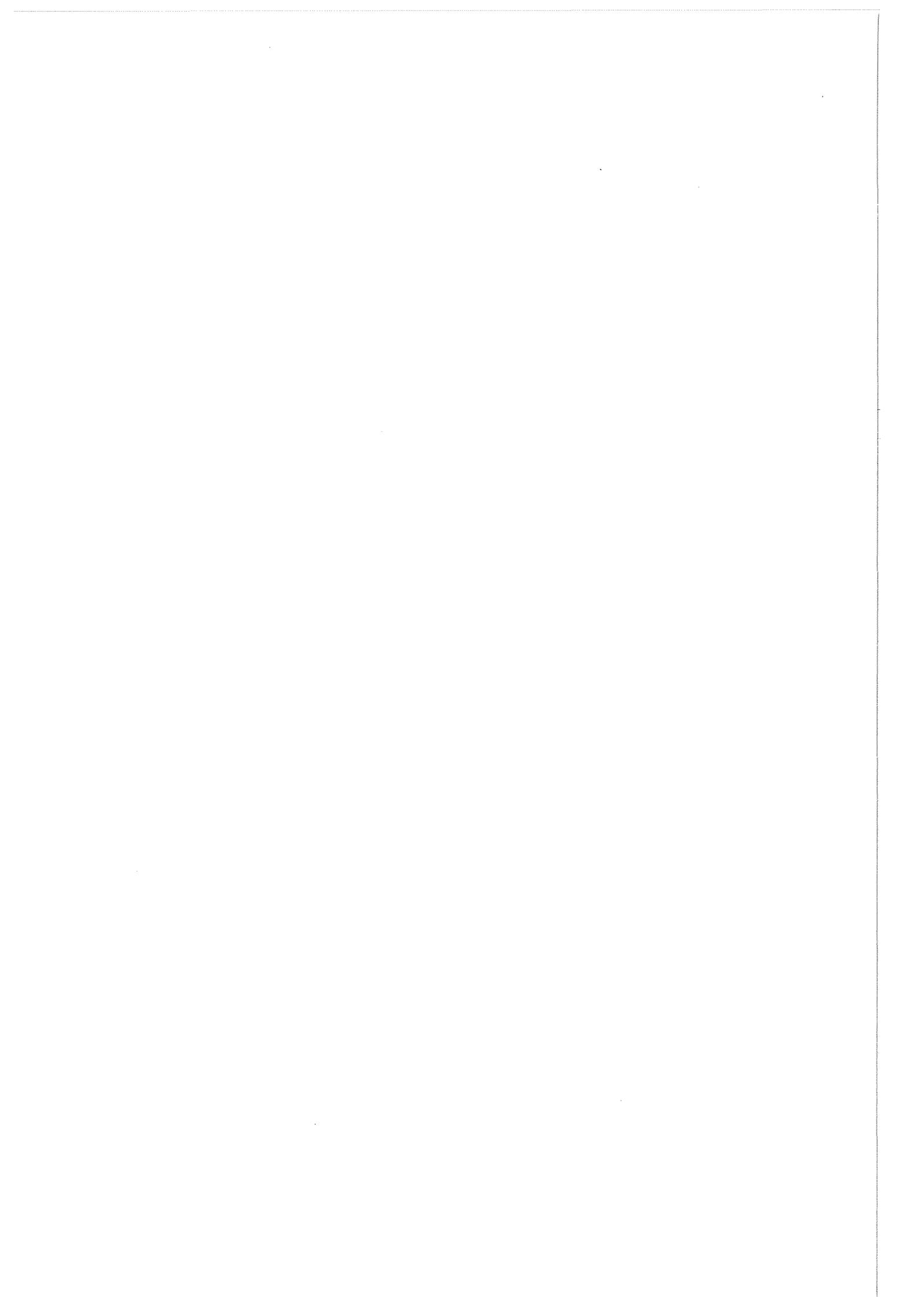


Dr. Dietmar Ratz

Zur Person und zum Vortrag

Dr. Dietmar Ratz studierte von 1982 bis 1988 Technomathematik an der Universität Karlsruhe mit den Nebenfächern Physik und angewandter Informatik. Darin eingeschlossen war ein Industriesemester bei Siemens. Seit 1988 ist er am Institut für Angewandte Mathematik in Karlsruhe als wissenschaftlicher Mitarbeiter tätig, wo er im Juli 1992 promovierte.

Viele Probleme der Kurvenanpassung aus dem technisch-wissenschaftlichen Bereich können als globale nichtlineare Optimierungsprobleme formuliert werden. Im Gegensatz zur lokalen Optimierung, bei der eine optimale Lösung in der Nähe einer vorgegebenen Stelle gesucht wird, verlangt die globale Optimierung das Auffinden des *besten* aus einer Vielzahl von Optima. Die Hauptschwierigkeit liegt nun darin, daß man es einem lokalen Extremum nicht ansieht, ob es auch global optimal ist, vielmehr muß der gesamte abzudeckende Bereich einschließlich des Randes betrachtet untersucht werden. Herr Dr. Ratz wird uns in seinem Vortrag eindrucksvoll darstellen, wie dies elegant mit den am Institut für Angewandte Mathematik von Prof. Kulisch entwickelten Einschließungsverfahren für die Intervallarithmetik gelöst werden kann.



Globale Optimierung mit Ergebnisverifikation

– Eine Einführung und ein Überblick über neuere Entwicklungen –

Dietmar Ratz

Institut für Angewandte Mathematik
Universität Karlsruhe

Zusammenfassung

Ziel der globalen Optimierung ist es, unter einer im allgemeinen großen Zahl *lokaler* Optima das *globale* Optimum zu lokalisieren. Globale Optimierungsverfahren können dazu zwar auf lokale Verfahren zurückgreifen, benötigen aber zusätzlich globale Informationen. Ein exzellentes Hilfsmittel zur Ermittlung globaler Informationen ist die Intervallrechnung, da es beispielsweise eine einzige Intervall-Funktionsauswertung ermöglicht, Aussagen über die Lage der Funktionswerte für alle Punkte (im allgemeinen unendliche viele) innerhalb des Intervalls zu machen.

Neuere Intervallverfahren sind in der Lage, garantierte Einschließungen für alle Lösungen eines globalen Optimierungsproblems zu berechnen. Selbst Eindeutigkeitsnachweise können im Rahmen der numerischen Durchführung erbracht werden. In den meisten Fällen ist die Effizienz dieser sogenannten Verifikationsverfahren vergleichbar mit der von klassischen Verfahren. Zahlreiche Standardtestaufgaben aus der Literatur können sogar schneller gelöst werden als von herkömmlichen Verfahren.

Es wird zunächst eine Einführung in die Grundlagen und Prinzipien solcher Verifikationsverfahren zur globalen Optimierung gegeben. Im Anschluß wird ein Überblick über neuere Methoden gegeben und mittels zahlreicher Beispiele ihre Effizienz demonstriert.

1 Einleitung, Problemstellung und Motivation

Viele Probleme aus dem Bereich technisch-wissenschaftlicher Anwendungen können als globale nichtlineare Optimierungsprobleme formuliert werden. Im Gegensatz zur lokalen Optimierung, bei der eine optimale Lösung in der Nähe eines vorgegebenen Punktes gesucht wird, verlangt die globale Optimierung das Auffinden des „besten“ lokalen Optimums. Während das Gebiet der lokalen nichtlinearen Optimierung bereits seit vielen Jahren erforscht wird und entsprechend zahlreiche Theorien, numerische Verfahren und Veröffentlichungen vorliegen, steckt das Gebiet der globalen Optimierung noch in den Kinderschuhen. Wie wichtig eine zukünftige intensive Forschung auf diesem Gebiet sein wird unterstreicht die Tatsache, daß die meisten sogenannten *real-world*-Probleme von globalem und nicht etwa lokalem Charakter sind (vgl. z. B. [18] und [23]).

Wir beschränken uns in diesem Artikel auf die Problemstellung der Minimierung, die aber natürlich leicht (durch Negation der Zielfunktion) in das entsprechende Maximierungsproblem überführt werden kann.

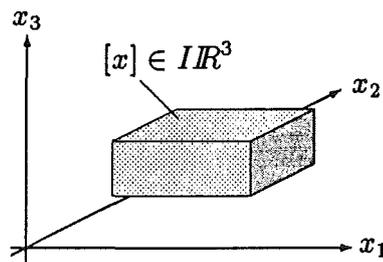


Abbildung 1: Ein dreidimensionaler Quader (Intervallvektor)

Gegeben seien also eine stetige Funktion $f : D \rightarrow \mathbb{R}$ und ein n -dimensionaler *Quader* $[x] \subseteq D \subseteq \mathbb{R}^n$ mit

$$[x] = ([x]_1, \dots, [x]_n) \quad \text{und} \quad [x]_i = [\underline{x}_i, \bar{x}_i]$$

den wir auch *Box* oder *Intervallvektor* nennen (vgl. Abbildung 1). Gesucht sind das *globale Minimum*

$$f^* = \min_{x \in [x]} f(x)$$

und *alle globale Minimalstellen* in $[x]$, d. h. die Menge

$$X^* = \{x \in [x] \mid f(x) = f^*\}.$$

Im allgemeinen existiert eine große Zahl lokaler Minima, und ein globales Optimierungsverfahren kann zwar auf bekannte lokale Verfahren zurückgreifen, benötigt aber zusätzlich globale Informationen um garantierte Aussagen über das tatsächliche Erreichen des globalen Minimums machen zu können. Setzt man nur lokale Methoden ein, so muß nämlich am Ende des Verfahrens eine der lokalen Lösungen zur globalen Lösung gemacht werden, ohne zu wissen, ob die „richtige“ lokale Lösung auch tatsächlich unter allen gefundenen ist.

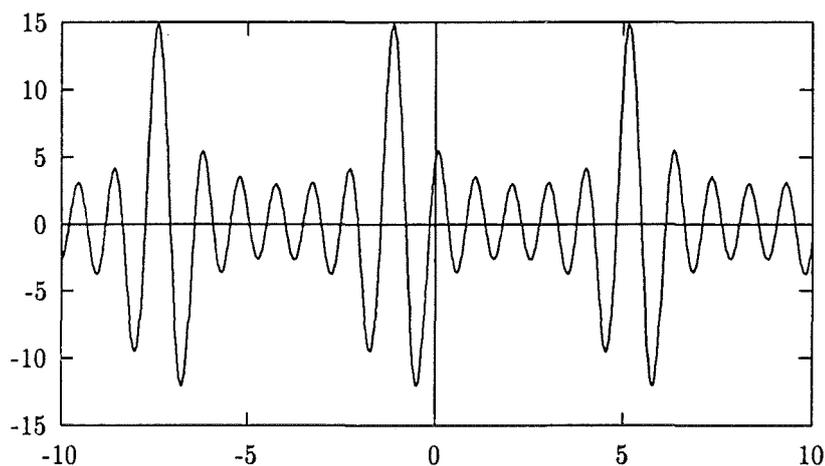


Abbildung 2: Funktion von Shubert

Klassische numerische Verfahren (approximative Verfahren) zur globalen Minimierung verwenden üblicherweise ein möglichst „dicht“ im Optimierungsgebiet verteiltes Raster

von Testpunkten, die als Startpunkte für lokale Iterationsverfahren (Abstiegsverfahren) dienen. Entsprechend problematisch gestaltet sich deshalb die Lösung von Problemen mit einer großen Zahl lokaler Minima. Wir wollen dies anhand der Funktion von Shubert (Abbildung 2) verdeutlichen, bei der viele Startpunkte und damit lokale Iterationen notwendig werden um die drei globalen Minimalstellen mittels approximativer Verfahren zu lokalisieren.

Abbildung 3 verdeutlicht eine weitere Problematik bei klassischen Optimierungsverfahren, wenn das globale Minimum in einem scharfen Peak liegt. Hier kann zum Beispiel der Fall eintreten, daß das eigentliche globale Minimum durch die (mit \times gekennzeichnete) Iterationsfolge eines Abstiegsverfahrens quasi „übersprungen“ und das in der Nähe liegende lokale Minimum 0 als vermeintliches globales Minimum ausgegeben wird.

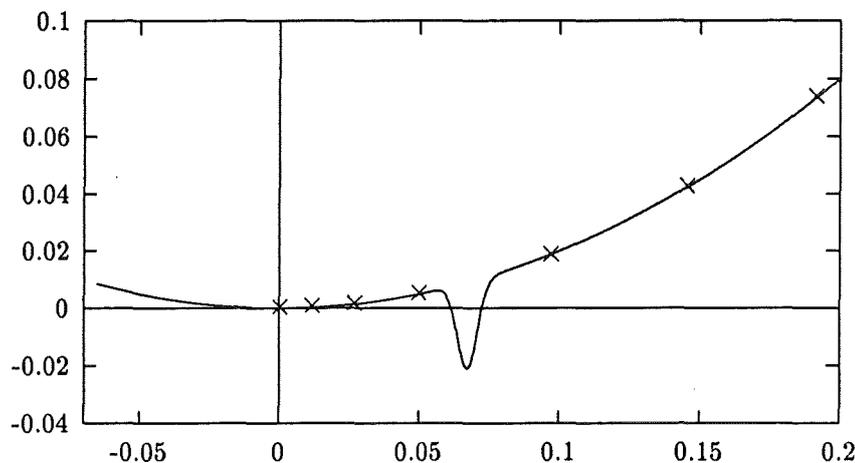


Abbildung 3: Funktion mit Peak

Die Notwendigkeit globaler Informationen, die sich somit offensichtlich stellt, wird durch den nachfolgenden Satz aus [23] nochmals unterstrichen.

Satz 1 *Ein globaler Optimierungsalgorithmus konvergiert genau dann gegen das globale Minimum einer stetigen Funktion f , wenn die Folge der durch den Algorithmus generierten Testpunkte überall dicht im kompakten Minimierungsbereich $[x]$ liegt.*

Beweis: Siehe [23].

Ein exzellentes Hilfsmittel zur Ermittlung globaler Informationen über Teilbereiche des Optimierungsgebietes ist die *Intervallrechnung*, denn Intervalle stellen ein ganzes Kontinuum dar, d. h. sie repräsentieren im allgemeinen unendlich viele Punkte, nämlich alle Punkte, die auf und zwischen den beiden Intervallgrenzen liegen. Unter Verwendung einer einzigen intervallarithmetischen Funktionsauswertung können beispielsweise Aussagen über die Lage der Funktionswerte für alle Punkte innerhalb eines Intervalls gemacht werden, denn die Intervallauswertung liefert eine Obermenge des Wertebereichs über dem Intervall. Somit ersetzt eine Intervallauswertung (unendlich) viele reelle Funktionsauswertungen. Bei der praktischen Durchführung auf dem Rechner werden auch alle auftretenden

Rundungsfehler mit erfaßt, so daß garantierte Fehlerschranken automatisch mitgeliefert werden.

Wenn wir mit diesem Hilfsmittel beispielsweise unsere bereits oben erwähnte Funktion mit dem Peak (Abbildung 3) behandeln wollen, so bringen bereits die drei Intervallauswertungen $[f_u]$, $[f_v]$ und $[f_w]$ von f über den Intervallen $[u]$, $[v]$ und $[w]$ wertvolle globale Informationen, die den approximativ berechneten Wert 0 in Frage stellen oder aber den zu untersuchenden Bereich verkleinern können. Schaut man sich die Funktionsintervalle am rechten Rand von Abbildung 4 an, so kann $[w]$ auf keinen Fall das globale Minimum enthalten, da die Funktionswerte in $[u]$ sämtlich kleiner sind. Außerdem deutet die Intervallauswertung über $[v]$ mit Funktionswerten kleiner als 0 darauf hin, daß dort das globale Minimum liegen könnte.

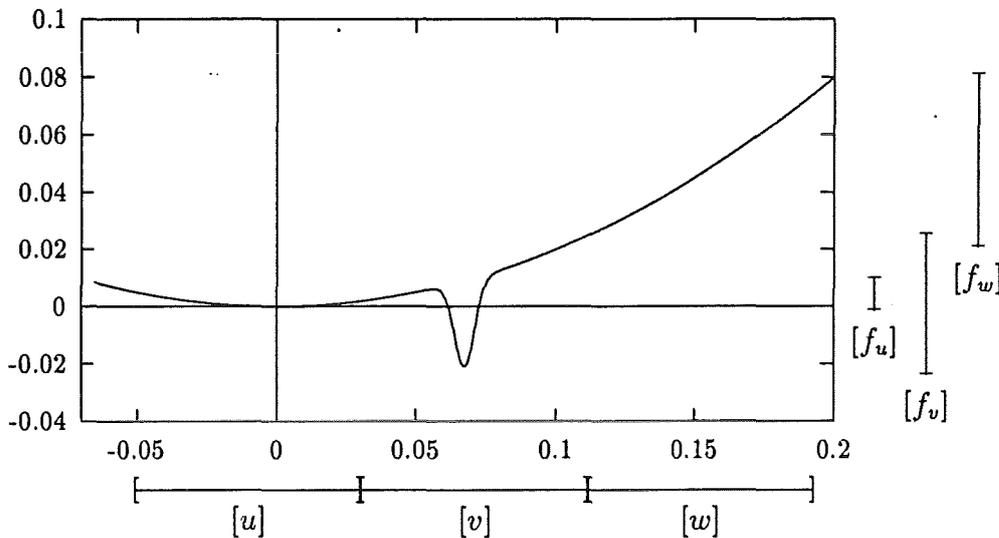


Abbildung 4: Intervallauswertungen für die Funktion mit Peak

2 Intervallarithmetik

Wir wollen nun einige Grundbegriffe der Intervallrechnung einführen und einige für das weitere Verständnis notwendige Eigenschaften erläutern. Eine ausführlichere Darstellung der Intervallrechnung findet sich zum Beispiel in [1] und in [6].

Die kompakte Menge

$$[a] := [\underline{a}, \bar{a}] := \{x \in \mathbb{R} \mid \underline{a} \leq x \leq \bar{a}\}$$

mit $\underline{a}, \bar{a} \in \mathbb{R}$ heißt reelles *Intervall*. Man nennt $\underline{a} = \inf[a]$ das *Infimum* oder die *Unterschranke* von $[a]$ und $\bar{a} = \sup[a]$ das *Supremum* oder die *Oberschranke* von $[a]$. Ein Intervall $[a]$ heißt *Punktintervall*, wenn gilt $\underline{a} = \bar{a}$. Die Menge aller reellen Intervalle bezeichnet man mit $I\mathbb{R} = \{[a] \mid \underline{a} \leq \bar{a}\}$. Die reellen Zahlen $m([a]) = \frac{1}{2}(\underline{a} + \bar{a})$ und $d([a]) = \bar{a} - \underline{a}$ heißen *Mittelpunkt* und *Durchmesser* von $[a]$.

Für zwei Intervalle $[a]$ und $[b]$ gilt

$$[a] \subseteq [b] \iff \underline{b} \leq \underline{a} \wedge \bar{a} \leq \bar{b},$$

$$[a] \overset{\circ}{\subset} [b] \iff \underline{b} < \underline{a} \wedge \bar{a} < \bar{b}.$$

Die *Verbandsoperationen* \cap (Schnitt) und \cup (Intervallhülle) sind für zwei Intervalle $[a]$ und $[b]$ definiert durch

$$[a] \cap [b] := [\max\{\underline{a}, \underline{b}\}, \min\{\bar{a}, \bar{b}\}],$$

$$[a] \cup [b] := [\min\{\underline{a}, \underline{b}\}, \max\{\bar{a}, \bar{b}\}].$$

Dabei ist der Schnitt nur definiert, falls $\max\{\underline{a}, \underline{b}\} \leq \min\{\bar{a}, \bar{b}\}$.

Die intervallarithmetischen Operationen $\circ \in \{+, -, \cdot, /\}$ werden definiert durch

$$[a] \circ [b] := \{a \circ b \mid a \in [a], b \in [b]\}.$$

Sie können auf Operationen mit den Intervallgrenzen zurückgeführt werden:

$$[a] + [b] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}],$$

$$[a] - [b] = [\underline{a} - \bar{b}, \bar{a} - \underline{b}],$$

$$[a] \cdot [b] = [\min\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}],$$

$$[a] / [b] = [a] \cdot [1/\bar{b}, 1/\underline{b}] \text{ für } 0 \notin [b].$$

Beispiele:

$$\begin{aligned} [1, 2] + [3, 4] &= [4, 6], \\ [1, 2] - [1, 2] &= [-1, 1], \\ [-1, 2] \cdot [4, 6] &= [-6, 12]. \end{aligned}$$

Das zweite Beispiel zeigt, daß für die Subtraktion im allgemeinen $[a] - [a] \neq [0, 0]$ gilt. Addition und Multiplikation sind kommutativ und assoziativ, für Intervalle gilt jedoch nur die sogenannte *Subdistributivität*

$$[a] \cdot ([b] + [c]) \subseteq [a] \cdot [b] + [a] \cdot [c],$$

bei der für $[a], [b], [c] \in IR$ nur in Ausnahmefällen Gleichheit gilt (z. B. wenn $\underline{b} \geq 0$ und $\underline{c} \geq 0$).

Beispiel:

$$\begin{aligned} [-1, 1] \cdot ([1, 2] + [-2, -1]) &= [-1, 1] \cdot [-1, 1] = [-1, 1] \\ [-1, 1] \cdot [1, 2] + [-1, 1] \cdot [-2, -1] &= [-2, 2] + [-2, 2] = [-4, 4] \end{aligned}$$

Die zentrale Eigenschaft der Intervalloperationen ist die *Inklusionsisotonie* oder *Einschließungseigenschaft*

$$a \in [a] \wedge b \in [b] \implies a \circ b \in [a] \circ [b]$$

$$[a] \subseteq [c] \wedge [b] \subseteq [d] \implies [a] \circ [b] \subseteq [c] \circ [d]$$

für alle $\circ \in \{+, -, \cdot, /\}$ mit $a, b \in \mathbb{R}$ und $[a], [b], [c], [d] \in IR$.

In ähnlicher Form ist es möglich die Elementarfunktionen $\varphi : D \subset \mathbb{R} \rightarrow \mathbb{R}$ (wie z. B. sin, cos, exp etc.) für Intervalle zu definieren durch

$$\varphi([x]) := \{\varphi(x) \mid x \in [x]\}.$$

Auch diese können teilweise wiederum auf Operationen für die Intervallgrenzen zurückgeführt werden.

Beispiele: $\varphi([x]) = [\varphi(\underline{x}), \varphi(\bar{x})], \quad \varphi \in \{\arctan, \operatorname{arsinh}, \ln, \sinh\},$
 $\varphi([x]) = [\varphi(\bar{x}), \varphi(\underline{x})], \quad \varphi \in \{\operatorname{arccot}, \operatorname{arcoth}\},$
 $\sqrt{[x]} = [\sqrt{\underline{x}}, \sqrt{\bar{x}}],$
 $e^{[x]} = [e^{\underline{x}}, e^{\bar{x}}].$

Die zentrale Eigenschaft der Inklusionsisotonie lautet für die Elementarfunktionen

$$[a] \subseteq [b] \implies \varphi([a]) \subseteq \varphi([b]).$$

Wir wollen drei wichtige Sachverhalte im Zusammenhang mit der Intervallarithmetik unterstreichen:

- Der Aufwand für eine Intervalloperation ist etwa doppelt so groß wie für die entsprechende reelle Operationen.
- Beim Rechnen mit Intervallen auf einer Rechenanlage ist zu beachten, daß alle Operationen mit *Außenrundung* ausgeführt werden müssen, um alle Rundungsfehler mit zu erfassen.
- Die reellen Intervalle müssen auf *Maschinenintervalle* abgebildet werden. Dazu werden die Intervallgrenzen durch gerichtete Rundungen (Außenrundung) auf Maschinenzahlen abgebildet. Man beachte jedoch, daß ein Intervall auf dem Rechner zwar Maschinenzahlen als Unter- und Obergrenzen besitzt, daß es aber trotzdem auch *alle reellen* Werte zwischen den Grenzen umfaßt! Das Maschinenintervall stellt somit auf dem Rechner ein Kontinuum dar.

Unter Einsatz der Intervallarithmetik ist es nun auf einfache Art und Weise möglich, den Wertebereich $W_f([x]) = \{f(x) \mid x \in [x]\}$ einer Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ über einem Intervall $[x]$ einzuschließen. Ersetzt man im entsprechenden Funktionsausdruck alle reellen Größen durch entsprechende Intervallgrößen und alle auftretenden Operationen durch die entsprechenden Intervalloperationen, so erhält man die *intervallmäßige Auswertung* $f([x])$ von f über $[x]$. Man nennt dies auch *natürliche Intervallerweiterung*.

Es gilt stets $W_f([x]) \subseteq f([x])$, d. h. die intervallmäßige Auswertung (auf dem Rechner einschließlich Außenrundung) liefert eine Obermenge (Einschließung) des Wertebereichs. Damit ist es prinzipiell möglich mit einer einzigen Intervallauswertung die Garantie dafür zu liefern, daß eine Funktion f keine Nullstelle bzw. keine negative Werte hat. Gilt nämlich $0 \notin f([x])$, so gilt auch $0 \notin W_f([x])$. Abbildung 5 demonstriert, daß auch eine Vielzahl von Gleitkommaauswertungen keine solche Garantie liefern kann, denn zwischen den Auswertestellen könnte immer noch ein „Ausreißer“ ins Negative vorliegen. Erst die Intervallauswertung, als Obermenge des Wertebereichs, kann dies ausschließen.

Ganz analog zum reellen können die notwendigen Operationen natürlich auch auf Intervallvektoren und -matrizen ausgedehnt werden, wobei Begriffe wie Mittelpunkt, Durchmesser, Vereinigung oder Schnitt jeweils komponentenweise interpretiert werden. Damit ist es möglich Einschließungsalgorithmen zur Lösung von mehrdimensionalen Problemen zu entwickeln wie z. B. für lineare und nichtlineare Gleichungssysteme oder globale Optimierungsprobleme (vgl. auch [6]).

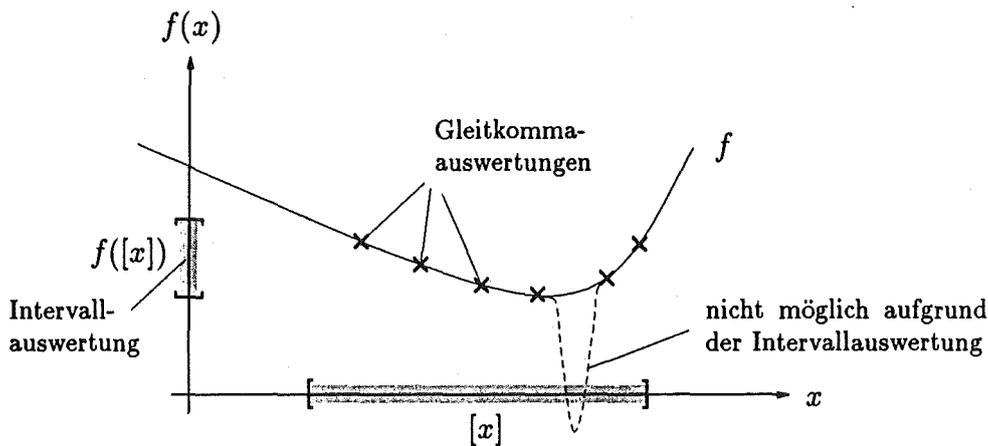


Abbildung 5: Intervallauswertung garantiert positive Werte von f in $[x]$

3 Globale Optimierung mit Ergebnisverifikation

Die Zielsetzung von Intervallverfahren zur Lösung des globalen Optimierungsproblems ist die Berechnung von *verifizierten Einschließungen* für das globale Minimum f^* und für die globalen Minimalstellen $x^* \in X^*$ innerhalb des Optimierungsbereichs $[x] \in \mathbb{IR}^n$.

Der zugrundeliegende Algorithmus basiert auf dem sogenannten *Intervall-Branch-and-Bound-Prinzip*:

Zerlege den Startbereich $[x]$ in Teilbereiche $[y] \subset [x]$ (*branches*), bestimme garantierte Schranken (*bounds*) für f auf den Teilbereichen $[y]$ und eliminiere diejenigen Teilbereiche, die aufgrund dieser Schranken kein globales Minimum enthalten können.

Neben der natürlicherweise eingesetzten Intervallarithmetik muß das Verfahren mit einer möglichst optimalen Aufteilungsstrategie und Listenverwaltung für die Teilbereiche arbeiten. Weitere Hilfsmittel zur Beschleunigung des Algorithmus sind

- Lokale (approximative) Optimierungsmethoden,
- Cut-Off-Tests,
- Monotonietest (falls f eine C^1 -Funktion),
- Konkavitätstest (falls f eine C^2 -Funktion) und
- Intervall-Newton-Schritt (falls f eine C^2 -Funktion).

Dabei ist es möglich, die benötigten Ableitungswerte mittels der sogenannten *automatischen Differentiation* zu berechnen. Diese ermöglicht es, bei der Berechnung eines Funktionswertes automatisch die Ableitungswerte mitzuberechnen. Dazu muß lediglich die Funktionsvorschrift bekannt sein, während die Ableitungsformeln nicht explizit angegeben werden müssen. Wir wollen auf diese Technik jedoch nicht weiter im Detail eingehen und verweisen für eine ausführlichere Darstellung auf [6].

Nach einer Beschreibung des Grund-Algorithmus, werden wir im folgenden auf die einzelnen Hilfsmittel eingehen, die in einem effizienten Verfahren zum Einsatz kommen. Auch dabei werden wir nicht bis ins kleinste Detail vorstoßen, um den Rahmen dieses Beitrages nicht zu sprengen.

3.1 Der Grund-Algorithmus

Wir wollen zunächst einmal schematisch die Arbeitsweise unseres Algorithmus beschreiben, der mit einer Liste L arbeitet, in der jeweils die noch zur Bearbeitung anstehenden Teilbereiche $[y]$ des Optimierungsbereichs $[x]$ abgespeichert werden. Außer dem jeweiligen Teilbereich $[y]$ ist zusätzlich die auf diesem Teilbereich berechnete garantierte Unterschranke

$$\underline{f}_y = \inf f([y])$$

für die Funktionswerte in $[y]$ mit abgespeichert. Zusätzlich verwendet der Algorithmus einen Wert \tilde{f} , der eine *garantierte Oberschranke für den Wert des globalen Minimums* darstellt, d. h. $\tilde{f} \geq f^*$. Der Algorithmus arbeitet nun mit L und \tilde{f} folgendermaßen:

Startphase:

- Die Arbeitsliste L wird mit dem Startbereich $[x]$ initialisiert:

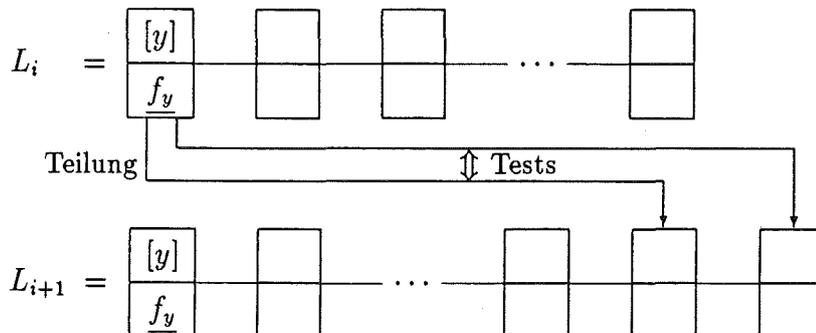
$$L = L_0 := \begin{array}{|c|} \hline [x] \\ \hline \underline{f}_x \\ \hline \end{array}, \text{ wobei } [\underline{f}_x, \overline{f}_x] = [f_x] = f([x])$$

- Die garantierte Oberschranke $\tilde{f} \geq f^*$ wird durch eine Intervallauswertung mit Punktargument initialisiert:

Wähle $c \in [x]$ (z. B. den Mittelpunkt), berechne $[f_c] := f_{\square}(c)$ intervallmäßig und setze $\tilde{f} := \overline{f}_c$.

Iteration:

- In der $(i + 1)$ -ten Iteration wird der erste Bereich $[y]$ in der Liste L_i (aus dem i -ten Iterationsschritt) aus der Liste entfernt und in zwei Teile aufgeteilt. Falls durch entsprechende Tests die Existenz einer globalen Minimalstelle in diesen Teilbereichen nicht ausgeschlossen werden kann, so werden die Teilbereiche an die Liste angehängt.



- Im nun neuen ersten Element der Liste L_{i+1} wird ein neues $c := m([y])$ gewählt und $[f_c] := f_0(c)$ intervallmäßig berechnet. Dann wird ein Update des Wertes \tilde{f} durch die Bestimmung von $\tilde{f} := \min\{\tilde{f}, \overline{f}_c\}$ durchgeführt.
- Mit dem neuen Wert \tilde{f} kann nun möglicherweise ein sogenannter *Cut-Off-Test* durchgeführt werden (s. u.).

Terminierung:

- Durch eine Genauigkeitsforderung (z. B. in Form einer Bedingung für die Durchmesser der Intervalle $[y]$ oder $f([y])$) kann der Algorithmus abgebrochen werden.

Ergebnis: Für das globale Minimum und die globalen Minimalstellen ergibt sich nach dem Ende des Algorithmus, daß

$$f^* \in [\min\{\underline{f}_y \in L\}, \tilde{f}], \quad \text{und} \quad X^* \subseteq \bigcup_{[y] \in L} [y].$$

Zu jedem Zeitpunkt während der Durchführung des Algorithmus stellt der kleinste \underline{f}_y -Wert in der Liste stets die beste bekannte *garantierte Unterschranke* und \tilde{f} die beste bekannte *garantierte Oberschranke* für das globale Minimum f^* dar. Somit kann auch die Distanz beider Werte als mögliches Abbruchkriterium für den Algorithmus verwendet werden.

3.2 Der Cut-Off-Test

Mit der garantierten Oberschranke \tilde{f} für das Minimum f^* ist es möglich, alle Teilboxen $[y]$ aus der Liste zu entfernen, die die Beziehung

$$f^* \leq \tilde{f} < \underline{f}_y$$

erfüllen, da \underline{f}_y ja jeweils eine Unterschranke für die tatsächlichen Funktionswerte von f auf der Teilbox $[y]$ darstellt. Abbildung 6 verdeutlicht dies an einem Beispiel. Die schraffierten

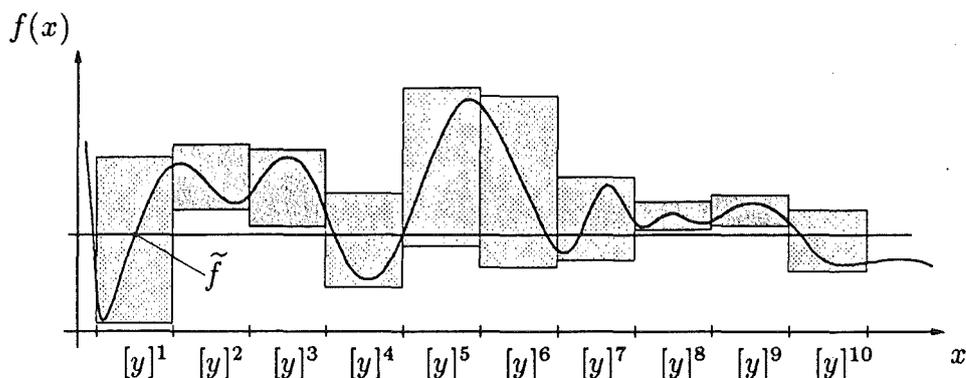


Abbildung 6: Der Cut-Off-Test

Rechtecke stellen dabei jeweils die Intervallauswertungen von f über dem entsprechenden Teil der x -Achse dar. Mit der Höhenlinie auf dem Niveau von \tilde{f} sieht man sofort, daß die dunkel schraffierten Gebiete kein globales Minimum enthalten können.

An diesem Beispiel wird auch deutlich, daß dieser Cut-Off-Test um so besser (schneller) funktioniert, je besser der Wert \tilde{f} ist. Somit können gerade hier lokale (approximative) Verfahren (Abstiegsverfahren) zur Verbesserung des Punktes c und damit der garantierten Obergrenze \tilde{f} zum Einsatz kommen. Würde nämlich im Beispiel (Abbildung 6) der Wert von \tilde{f} in der Nähe des linken Randes von $[y]^1$ bestimmt werden, so könnte der Cut-Off-Test sogar alle anderen Teilintervalle $[y]^k$ mit $k = 2, \dots, 10$ aus der Liste entfernen.

Diese Tatsache verdeutlicht die Vorgehensweise der Verifikationsverfahren zur globalen Optimierung, nämlich die *Ausnutzung der jeweiligen Vorteile beim wechselseitigen Einsatz von Gleitkomma- und Intervallrechnung*. Der Wert c kann approximativ „verbessert“ werden, während der garantierte Wert f durch eine Intervallauswertung bestimmt werden muß.

3.3 Algorithmische Darstellung

Wir wollen nun noch eine vereinfachte algorithmische Beschreibung unseres grundlegenden Verfahrens geben, um die prinzipielle Vorgehensweise zu verdeutlichen. Bei gegebener Arbeitsliste L , Listenelementen $([y], \underline{f}_y)$, und zu optimierender Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$, verwenden wir dabei die folgenden Notationen:

Notation	Bedeutung
$L := \{ \}$	Initialisierung von L als leere Liste
$L := L + ([y], \underline{f}_y)$	Einhängen von $([y], \underline{f}_y)$ in L
$([y], \underline{f}_y) := \text{PopHead}(L)$	Aushängen von $([y], \underline{f}_y)$ mit kleinstem \underline{f}_y aus L
\underline{f}_y	Unterschranke der Intervallauswertung $[f_y] := f([y])$
$f_0(c)$	Intervallauswertung von f im Punkt c

Algorithmus GlobalOptimize ($f, [x], \varepsilon, L_{\text{res}}, [f^*]$)

1. $[f_c] := f_0(m([x])); \tilde{f} := \overline{f}_c;$
2. $[y] := [x]; L := \{ \}; L_{\text{res}} := \{ \};$
3. **repeat**
 - (a) $k := \text{OptimalComponent}([y]); \text{Bisect}([y], k, [u]^1, [u]^2);$
 - (b) **for** $i := 1$ **to** 2 **do**
 - i. $[f_u] := f([u]^i);$
 - ii. **if** $\tilde{f} < \underline{f}_u$ **then next**;
 - iii. Apply additional tests and methods;
 - iv. $L := L + ([u]^i, \underline{f}_u); \{ \text{Store } [u]^i \}$
 - (c) $\text{Bisect} := \text{false};$
 - (d) **while** $(L \neq \{ \})$ **and** **(not Bisect)** **do**

- i. $([y], \underline{f}_y) := \text{PopHead}(L); \quad c := m([y]);$
 - ii. Apply approximate method to improve c ;
 - iii. $[f_c] := f_0(c); \quad \tilde{f} := \min\{\tilde{f}, \tilde{f}_c\}; \quad \text{CutOffTest}(L, \tilde{f});$
 - iv. **if** $(d(f([y])) < \varepsilon)$ **or** $(d([y]) < \varepsilon)$ **then**
 $L_{\text{res}} := L_{\text{res}} + ([y], \underline{f}_y);$
else $\text{Bisect} := \text{true};$
- until** $(\text{not } \text{Bisect});$
4. $[f^*] := [\min\{f_y \in L\}, \tilde{f}];$
 5. **return** $L_{\text{res}}, [f^*];$

Als Eingabedaten erhält GlobalOptimize die Funktion f , die Start-Box $[x]$ und einen (oder auch mehrere) Genauigkeitsparameter ε . In Schritt 1 wird zunächst durch eine intervallmäßige Mittelpunktsauswertung eine erste garantierte Oberschranke \tilde{f} für f^* bestimmt. Danach wird die aktuelle Box $[y]$ mit der Startbox $[x]$ und die Arbeitsliste L sowie die Ergebnisliste L_{res} jeweils mit einer leeren Liste initialisiert.

Schritt 3 stellt die eigentliche Iteration dar, in der in Schritt (a) jeweils eine Bisektion der aktuellen Box $[y]$ bezüglich einer möglichst optimal gewählten Koordinate k durchgeführt wird (Details zu dieser Richtungswahl finden sich z. B. in [20]). Danach werden die beiden Hälften $[u]^1$ und $[u]^2$ daraufhin untersucht, ob sie eine globale Minimalstelle enthalten können oder nicht. In Schritt (b)iii können dabei die in den nachfolgenden Abschnitten beschriebenen Methoden zum Einsatz kommen. Kann die Teilbox nicht eliminiert werden, so wird sie in Schritt (b)iv an die Liste L angehängt.

In Schritt 3(d) wird jeweils eine neue aktuelle Box $[y]$ aus der Liste ausgehängt und ausgehend von deren Mittelpunkt eine lokale approximative Methode zur Verbesserung des Wertes c und damit der Oberschranke \tilde{f} angewendet. Mit diesem Wert wird dann der Cut-Off-Test durchgeführt. Wenn die aktuelle Box bereits einer bestimmten Abbruchbedingung bezüglich des Durchmessers genügt, dann wird diese in die Ergebnisliste L_{res} eingehängt, andernfalls wird sie weiter halbiert.

In Schritt 4 wird noch die bestmögliche Einschließung $[f^*]$ für den Wert des globalen Minimums bestimmt, und zuletzt werden L_{res} und $[f^*]$ als Ergebnisse des Algorithmus zurückgegeben.

In den folgenden Abschnitten wollen wir nun noch auf die weiteren Hilfsmittel eingehen, die in Schritt 3(b)iii zum Einsatz kommen können, sofern die zu behandelnde Funktion f die jeweils erforderlichen Differenzierbarkeitsvoraussetzungen erfüllt.

3.4 Der Monotonietest

Ist f streng monoton in einer Box $[y] \subset [x]$, so kann diese aus der Liste gelöscht werden, denn $[y]$ kann in diesem Fall keinen stationären Punkt und somit auch kein lokales oder gar globales Minimum enthalten. Die Ausnahme bilden natürlich Randminima auf dem Rand des ursprünglichen Optimierungsgebietes $[x]$, die keine stationäre Punkte sind.

Erfüllt also die intervallmäßige Auswertung des Gradienten $[g] = \nabla f([y])$ die Bedingung

$$0 \notin [g]; \quad \text{für ein } i = 1, \dots, n,$$

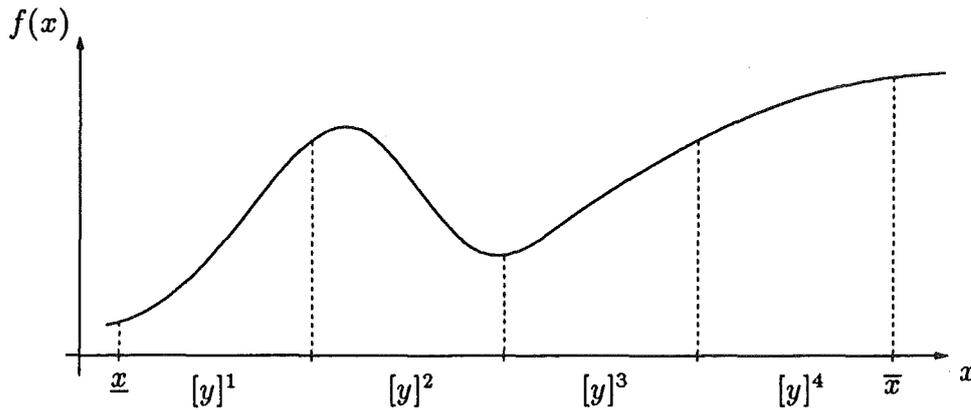


Abbildung 7: Zum Monotonietest

dann besitzt der Gradient keine Nullstelle in $[y]$, und f ist streng monoton in $[y]$ bezüglich der i -ten Koordinate. Die Box $[y]$ kann also gelöscht werden. Enthält $[y]$ allerdings Teile des Randes von $[x]$, so dürfen diese (zumindest vom Monotonietest) je nach Monotonie-eigenschaft *nicht* gelöscht werden.

In dem in Abbildung 7 skizzierten Fall mit $n = 1$ und für vier Teilintervalle kann der Monotonietest $[y]^1$ auf den Randpunkt \underline{x} reduzieren, die Box $[y]^2$ bleibt unverändert, und $[y]^3$ kann gelöscht werden. Da f in $[y]^4$ monoton *steigend* ist, kann das ganze Intervall gelöscht werden, da ja ein Minimum gesucht wird.

In diesem Beispiel reduziert der Monotonietest die ursprüngliche Liste von vier Intervallen $[y]^1$, $[y]^2$, $[y]^3$ und $[y]^4$ auf zwei Intervalle $[\underline{x}]$ und $[y]^2$. Hier wäre sogar der Fall gegeben, daß ein anschließender Cut-Off-Test mit einer Funktionsauswertung in $[\underline{x}]$ außerdem noch $[y]^2$ eliminieren würde und damit die eindeutige Lösung $x^* = \underline{x}$ zum Preis von nur vier Intervallauswertungen des Gradienten und zwei Intervallauswertungen von f bestimmt wäre.

4 Der Konkavitätstest

Dieser Test prüft eigentlich die „Nicht-Konvexität“, und er erhielt seinen Namen in der Literatur wohl zur Vereinfachung der Sprechweise. Er wird verwendet, um festzustellen ob die zu minimierende Funktion f auf einer Teilbox $[y] \subset [x]$ *nicht konvex* ist. Ist dies der Fall, so braucht $[y]$ nicht mehr weiter auf eine Minimalstelle untersucht zu werden, außer wenn $[y]$ Teile des Randes von $[x]$ enthält. In diesem Fall kann $[y]$, wie auch im Monotonietest, zumindest auf den Rand, der möglicherweise eine globale Minimalstelle enthält, reduziert werden.

Eine Funktion f ist konvex in $[y]$, wenn ihre Hessematrix überall in $[y]$ positiv semidefinit ist (vgl. Satz 1.6.3 in [24]). Eine notwendige Bedingung für die positive Semidefinitheit ist, daß alle Diagonalelemente der Hessematrix nichtnegativ sind (vgl. Satz 1.1.2 in [20]). Gilt nun für alle Hessematrizen in $[y]$, daß *ein* Diagonalelement kleiner als 0 ist, so kann die Hessematrix *nicht* positiv semidefinit sein.

Erfüllt also die intervallmäßige Auswertung der Hessematrix $[H] = \nabla^2 f([y])$ die Be-

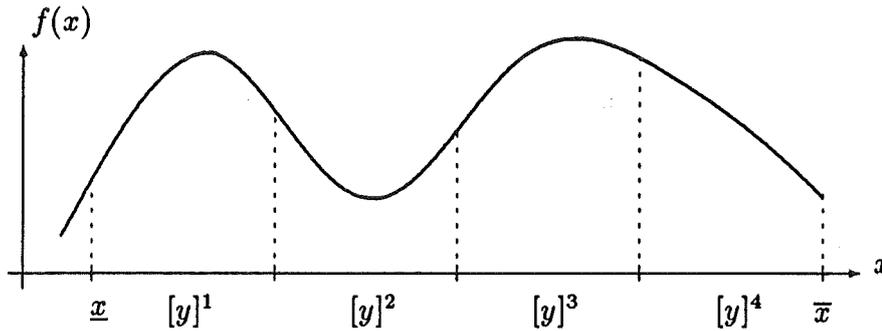


Abbildung 8: Zum Konkavitätstest

dingung

$$\overline{H}_{ii} < 0 \quad \text{für ein } i = 1, \dots, n,$$

dann ist $H_{ii} < 0$ für $H = \nabla^2 f(y)$ und für alle $y \in [y]$. Somit kann $[y]$ gelöscht werden.

Abbildung 8 skizziert den eindimensionalen Fall für vier Teilintervalle, in dem der Konkavitätstest $[y]^1$ auf den Randpunkt \underline{x} reduzieren, $[y]^3$ löschen und $[y]^4$ auf den Randpunkt \overline{x} reduzieren kann. Die Box $[y]^2$ bleibt erhalten.

4.1 Intervall-Newton-(Gauß-Seidel)-Schritt

Ein weiteres Hilfsmittel zur Elimination bzw. zur Verkleinerung einer Teil-Box $[y]$ ist die Anwendung eines einzelnen Schrittes aus der Intervall-Newton- bzw. Gauß-Seidel-Iteration zur Lösung des nichtlinearen Gleichungssystems

$$\nabla f(y) = 0, \quad y \in [y].$$

Dabei wird eine a-priori Einschließung $[y]$ der Nullstelle des Gradienten durch Auflösen des Intervallgleichungssystems

$$[A](c - x) = b$$

mit $[A] \in I\mathbb{R}^{n \times n}$ und $b \in \mathbb{R}^n$ verbessert. $[A]$ und b sind dabei gegeben durch

$$[A] := R \cdot \nabla^2 f([y]) \quad \text{und} \quad b := R \cdot \nabla f(c),$$

wobei $R \approx (m(\nabla^2 f([y])))^{-1}$ und $c := m([y])$.

Die neue (verbesserte) Einschließung $N'_{GS}([y])$ berechnet sich gemäß

$$N'_{GS}([y]) := [z] \left. \begin{array}{l} [z] := [y] \\ [z]_i := \left(c_i - \left(b_i + \sum_{\substack{j=1 \\ j \neq i}}^n [A]_{ij} \cdot ([z]_j - c_j) \right) / [A]_{ii} \right) \cap [z]_i \\ i = 1, \dots, n \end{array} \right\}$$

Bei der Division durch $[A]_{ii}$ tritt hier das zunächst formale Problem auf, daß der Nenner die Null enthalten könnte. Aus diesem Grund bedient man sich im Intervall-Gauß-Seidel-Schritt der sogenannten *erweiterten Intervallarithmetik* (vgl. [6], [9]), die eine Division

durch Intervalle, die die Null enthalten, erlaubt. Wir wollen dies hier nicht weiter vertiefen und verweisen für eine detaillierte Beschreibung des Intervall-Gauß-Seidel-Schrittes auf [1], [6, Kapitel 13,14] und [20, Abschnitt 2.5].

Im Rahmen unseres globalen Optimierungsverfahrens wäre eine vollständige Intervall-Newton-Iteration zwar möglich, aus „Kostengründen“ wird dies aber nicht praktiziert. Der einzelne Newton-Schritt ist aufgrund der notwendigen Hessematrix-Auswertung relativ teuer, während die anderen Tests, die möglicherweise eine Box löschen bevor ein weiterer Newton-Schritt notwendig wird, wesentlich billiger sind. Außerdem behandelt der Newton-Schritt das Nullstellenproblem für den Gradienten und zielt damit auf die Berechnung eines stationären Punktes ab, der nicht notwendigerweise ein Minimalpunkt sein muß.

Die Anwendung des Intervall-Gauß-Seidel-Schrittes innerhalb unseres Verfahrens kann nun folgende Ergebnisse liefern:

- Die Ergebnisbox $N'_{GS}([y])$ ist leer, d. h. $[y]$ enthält keinen stationären Punkt.
- Die Ergebnisbox $N'_{GS}([y])$ ist signifikant verkleinert worden.
- Ergebnisbox $N'_{GS}([y])$ ist insgesamt verkleinert und in Teilboxen aufgespalten worden.

Wir wollen dies im eindimensionalen Fall kurz graphisch erläutern.

Ähnlich wie das klassische Newton-Verfahren kann auch das Intervall-Newton-Verfahren geometrisch dadurch interpretiert werden, daß in jedem Iterationsschritt an der Stelle $c = m([y])$ zwei Geraden an den Graphen der Funktion f angelegt werden. Es handelt sich dabei um die Geraden mit der Steigung $f'([y])$ bzw. $\overline{f'([y])}$, also mit der kleinsten bzw. größten Steigung von f im Intervall $[y]$. Ihre Schnittpunkte mit der x -Achse entsprechen den Intervallgrenzen der neuen Iterierten, die nachfolgend noch mit der alten Iterierten $[y]$ geschnitten wird. Anhand von zwei Skizzen wollen wir diese geometrische Interpretation für den Fall $0 \notin f'([y])$ und für den Fall $0 \in f'([y])$ veranschaulichen.

In Abbildung 9 ist für $[y] = [\underline{y}, \bar{y}]$ der Fall $0 \notin f'([y])$ skizziert. Die Gerade mit der kleinsten Steigung schneidet die x -Achse in λ , die Gerade mit der größten Steigung schneidet sie in ρ . In einem Newton-Schritt wird also zunächst das Intervall

$$[w] := c - \frac{f(c)}{f'([y])} = [\lambda, \rho]$$

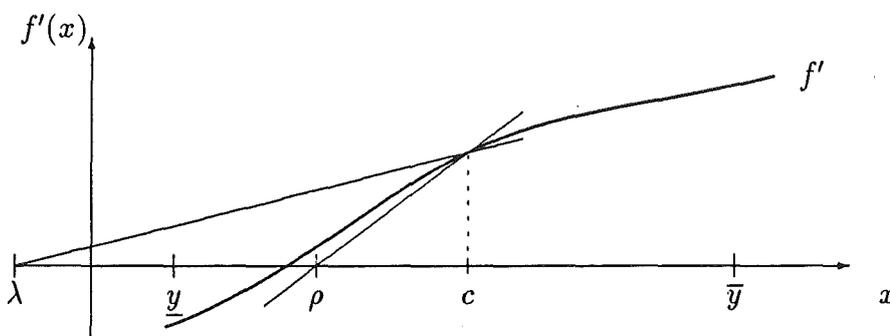


Abbildung 9: Intervall-Newton-Schritt mit $0 \notin f'([y])$

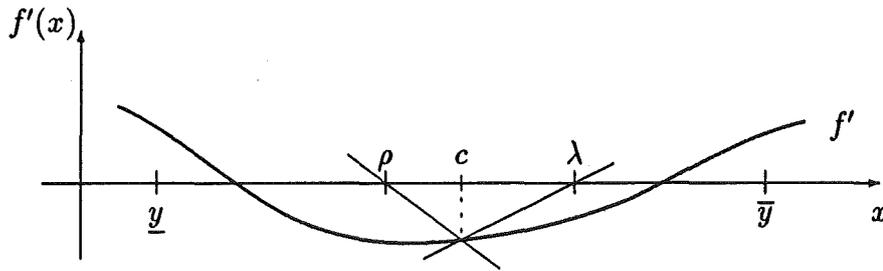


Abbildung 10: Intervall-Newton-Schritt mit $0 \in f'([y])$

berechnet, das in der Skizze links vom Mittelpunkt liegt. Die im Anschluß daran erfolgende Schnittbildung liefert somit

$$N'([y]) := [w] \cap [y] = [\lambda, \rho] \cap [y, \bar{y}] = [y, \rho],$$

als neue Iterierte und somit verbesserte Einschließung der Nullstelle von f' .

In Abbildung 10 ist für $[y] = [y, \bar{y}]$ der Fall $0 \in f'([y])$ skizziert. Die Gerade mit der kleinsten Steigung schneidet die x -Achse in ρ , die Gerade mit der größten Steigung schneidet sie in λ . In einem (erweiterten) Newton-Schritt wird also zunächst das erweiterte Intervall

$$[w] := c - \frac{f(c)}{f'([y])} = (-\infty, \rho] \cup [\lambda, \infty)$$

berechnet und die anschließende Schnittbildung liefert

$$N'([y]) := [w] \cap [y] = ((-\infty, \rho] \cup [\lambda, \infty)) \cap [y, \bar{y}] = [y, \rho] \cup [\lambda, \bar{y}],$$

also die Vereinigung zweier Intervalle, die nach wie vor die Nullstellen von f' in $[y]$ enthält.

4.2 Eindeutigkeitsaussagen

Unter Anwendung von speziellen Fixpunktsätzen, deren Voraussetzungen mittels Intervallrechnung *auf dem Rechner überprüft werden können*, ist es möglich, daß das Verfahren selbst nachweist, daß eine Teilbox $[y]$ in der Ergebnisliste L_{res} eine (lokal in $[y]$) eindeutige Minimalstelle enthält. Dies ist möglich durch den Nachweis eines eindeutigen stationären Punktes in $[y]$, d. h. den Nachweis von Existenz und Eindeutigkeit einer Nullstelle von ∇f in $[y]$, zusammen mit dem Nachweis, daß f in $[y]$ streng konvex ist, d. h. durch den Nachweis der positiven Definitheit aller $\nabla^2 f$ in $[y]$.

Aufgrund der Eigenschaften des Intervall-Gauß-Seidel-Schrittes läßt sich der erste Punkt recht leicht überprüfen. Wir fassen diese im nachfolgenden Satz zusammen.

Satz 2 Sei $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ eine zweimal stetig differenzierbare Funktion, und sei $[y] \in I\mathbb{R}^n$ mit $[y] \subseteq D$. Dann besitzt $N'_{\text{GS}}([y])$ die folgenden Eigenschaften:

1. Jede Nullstelle x^* in $[y]$ von ∇f liegt auch in $N'_{\text{GS}}([y])$.
2. Ist $N'_{\text{GS}}([y]) = \emptyset$, dann existiert keine Nullstelle von ∇f in $[y]$.
3. Gilt $N'_{\text{GS}}([y]) \overset{\circ}{\subset} [y]$, dann existiert eine eindeutige Nullstelle von ∇f in $[y]$ und damit auch in $N'_{\text{GS}}([y])$.

Beweis: Siehe z. B. [9].

Somit muß das Verfahren lediglich überprüfen, ob $N'_{GS}([y])$ echt im Inneren von $[y]$ liegt. Mit einer ähnlichen Inklusionsbedingung läßt sich auch die positive Definitheit aller Hessematrizen in $[y]$ nachweisen. Wir wollen jedoch auch dies, um den Rahmen dieses Beitrags nicht zu sprengen, hier nicht weiter vertiefen und verweisen auf [6, Kapitel 14] und [20, Abschnitt 2.7].

5 Historie der Methoden und Varianten

Um dem interessierten Leser bzw. der interessierten Leserin die Möglichkeit zu geben, sich tiefer in die vorgestellte Methodik einzuarbeiten, geben wir nun einen kleinen historischen Überblick. Darin führen wir die wichtigsten Stationen der Entstehung verschiedener Varianten von Intervallverfahren zur globalen Optimierung einschließlich entsprechender Literaturreferenzen auf. Wir erheben jedoch keinen Anspruch auf Vollständigkeit.

1966: Durch das Buch von Moore [17] wird der Grundstock zur Verwendung der Intervallarithmetik für die globale Optimierung gelegt.

1974: Skelboe kombiniert in [22] eine Branch-and-Bound-Strategie mit Moores Intervallverfahren und entwickelt ein erstes ableitungsfreies Verfahren, das darauf abzielt, Einschließungen für f^* zu berechnen.

1979: Ichida und Fujii [10] entwickeln eine Modifikation des Skelboe-Verfahrens unter Einbeziehung des Mittelpunktstests und des Intervall-Newton-Verfahrens. Auch diese zielt darauf ab, nur Einschließungen für f^* zu berechnen.

1979/80: Eines der wichtigsten Intervall-Verfahren, veröffentlicht Hansen 1979 für den eindimensionalen Fall [7] und 1980 für den mehrdimensionalen Fall [8]. Es integriert die Mittelpunkts-, Monotonie- und Konkavitätstests sowie ein spezialisiertes Intervall-Newton-Verfahren mit dem Ziel, Einschließungen für f^* und für X^* zu berechnen.

1988: Ratschek und Rokne fassen in [18] die obigen Verfahren zusammen und führen umfangreiche Konvergenzuntersuchungen und Vergleiche durch.

1988/89/90: In [2], [3] und [4] beschreibt Csendes ein Verfahren mit Cut-Off-Test und Monotonietest speziell für Parameterschätzungsaufgaben im Bereich von Fertigungstoleranzen. Auch hier gilt das Interesse vornehmlich f^* .

1991/92: Jansson und Knüppel stellen in [11] und [12] ein Intervall-Branch-and-Bound-Verfahren vor, das ohne Ableitungen auskommt und unter intensiver Nutzung lokaler approximativer Verfahren sehr schnell gute Näherungen und garantierte Schranken für f^* liefert.

1991/92/93/94: In [19], [20], [6] und [21] beschreiben wir effiziente Modifikationen des Verfahrens von Hansen unter Einsatz von optimierter Bisektion, Cut-Off-, Monotonie- und Konkavitätstests, Intervall-Gauß-Seidel-Schritt mit modifiziertem

Box-Splitting, spezieller Randbehandlung, Eindeutigkeitsnachweisen und automatischer Differentiation. Zielsetzung ist dabei, Einschließungen von hoher Genauigkeit für f^* und X^* zu berechnen.

1992: In seinem Buch [9] faßt Hansen Modifikationen früherer Varianten seines Verfahrens, algorithmische Beschreibungen, theoretische Konvergenzuntersuchungen und zahlreiche Testbeispiele zusammen.

1993: In der „Numerical Toolbox for Verified Computing I“ [6] wird (unter anderem) erstmals Public-Domain-Software für die globale Optimierung mit automatischer Ergebnisverifikation zur Verfügung gestellt.

1994: Jansson beschreibt in [13] ein sehr effizientes Verfahren unter Verwendung von Ableitungen, mit Eindeigkeitstests für stationäre Punkte und mit einem sogenannten Expansionsprinzip. Dieses Verfahren ermöglicht es, hochgenaue Einschließungen sowohl für f^* als auch für X^* zu berechnen.

6 Numerische Resultate und Laufzeitvergleiche

Mit den nachfolgend aufgeführten Ergebnissen für Testbeispiele aus der Literatur wollen wir die Effizienz der Intervallverfahren für die globale Optimierung unterstreichen. Wir beschränken uns dabei auf die Resultate in [9], [14], [20] und [21].

6.1 Standard-Testbeispiele

Wir wollen uns zunächst einmal den Funktionen widmen, die in [5] zu einem Standardsatz zum Test von globalen Optimierungsverfahren erklärt worden sind. Neben den Funktionsvorschriften geben wir auch die mit einer geforderten relativen Genauigkeit 10^{-12} berechneten Einschließungen für die Minimalwerte und die globalen Minimalstellen an.

S5, S7, S10: Die Shekel-Funktionen ($x \in \mathbb{R}^4$):

$$f_{Sm}(x) = - \sum_{i=1}^m \frac{1}{(x - A_i)(x - A_i)^T + c_i},$$

für $m = 5, m = 7$ und $m = 10$. Dabei ist

$$A = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix} \quad \text{und} \quad c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5 \end{pmatrix}.$$

Startbereich: $0 \leq x_i \leq 10, i = 1, \dots, 4$.

Ergebnisse für S5:

Minimalstelle: [4.000037152818980E+000, 4.000037152821031E+000]
[4.000133276591354E+000, 4.000133276591888E+000]
[4.000037152819639E+000, 4.000037152819693E+000]
[4.000133276591559E+000, 4.000133276591561E+000]

Minimum: [-1.015319967905829E+001, -1.015319967905816E+001]

Ergebnisse für S7:

Minimalstelle: [4.000572916185218E+000, 4.000572916186515E+000]
[4.000689366185151E+000, 4.000689366185395E+000]
[3.999489708859148E+000, 3.999489708859153E+000]
[3.999606158858631E+000, 3.999606158858633E+000]

Minimum: [-1.040294056681887E+001, -1.040294056681845E+001]

Ergebnisse für S10:

Minimalstelle: [4.000746531591439E+000, 4.000746531592502E+000]
[4.000592934138421E+000, 4.000592934138670E+000]
[3.999663398040321E+000, 3.999663398040324E+000]
[3.999509800586807E+000, 3.999509800586809E+000]

Minimum: [-1.053640981669226E+001, -1.053640981669182E+001]

H3: Die Hartman-Funktion der Dimension 3 ($x \in \mathbb{R}^3$):

$$f_{H3}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 A_{ij} (x_j - P_{ij})^2 \right).$$

$$A = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix} \quad \text{und} \quad P = \begin{pmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4378 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$$

Startbereich: $0 \leq x_i \leq 1, i = 1, \dots, 3.$

Ergebnisse:

Minimalstelle: [1.145248868047914E-001, 1.145248868047942E-001]
[5.555230190395223E-001, 5.555230190395230E-001]
[8.525997844999939E-001, 8.525997844999945E-001]

Minimum: [-3.861305797100195E+000, -3.861305797100181E+000]

H6: Die Hartman-Funktion der Dimension 6 ($x \in \mathbb{R}^6$):

$$f_{H6}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^6 A_{ij} (x_j - P_{ij})^2 \right).$$

$$A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix} \quad \text{und}$$

$$P = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$$

Startbereich: $0 \leq x_i \leq 1, i = 1, \dots, 6$.

Ergebnisse:

Minimalstelle: [2.016895110066914E-001, 2.016895110067208E-001]
 [1.500106918234515E-001, 1.500106918234638E-001]
 [4.768739742218904E-001, 4.768739742219030E-001]
 [2.753324304940550E-001, 2.753324304940572E-001]
 [3.116516166001127E-001, 3.116516166001138E-001]
 [6.573005340656198E-001, 6.573005340656208E-001]

Minimum: [-3.322368011415553E+000, -3.322368011415477E+000]

BR: Die Branin-Funktion ($x \in \mathbb{R}^2$):

$$f_{BR}(x) = \left(\frac{5}{\pi} x_1 - \frac{5.1}{4\pi^2} x_1^2 + x_2 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10.$$

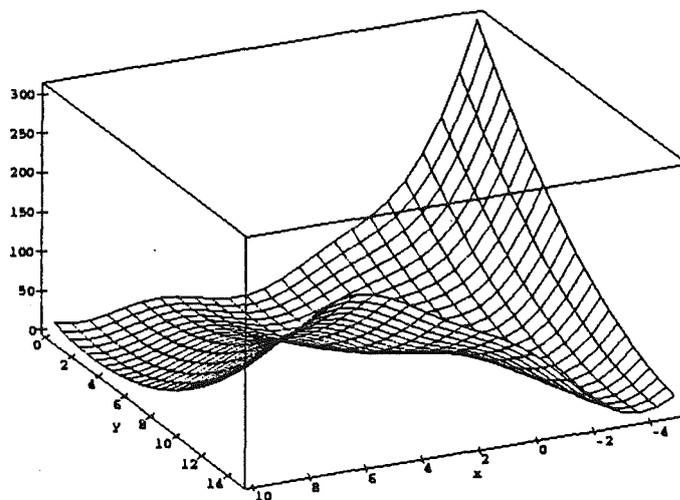


Abbildung 11: Die Branin-Funktion

Startbereich: $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$.

Ergebnisse:

- Minimalstelle(n):
1. [3.141592653589792E+000, 3.141592653589795E+000]
[2.274999999999996E+000, 2.275000000000005E+000]
 2. [9.424777960769374E+000, 9.424777960769387E+000]
[2.474999999999978E+000, 2.475000000000021E+000]
 3. [-3.141592653589798E+000, -3.141592653589789E+000]
[1.227499999999998E+001, 1.227500000000002E+001]
- Minimum: [3.978873577297381E-001, 3.978873577297435E-001]

SHCB: Die *Six-Hump-Camel-Back*-Funktion ($x \in \mathbb{R}^2$):

$$f_{\text{SHCB}}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4.$$

Startbereich: $-2.5 \leq x_i \leq 2.5$.

Ergebnisse:

- Minimalstelle(n):
1. [-8.984201310032836E-002, -8.984201310031070E-002]
[7.126564030207390E-001, 7.126564030207405E-001]
 2. [8.984201310031189E-002, 8.984201310032657E-002]
[-7.126564030207403E-001, -7.126564030207390E-001]
- Minimum: [-1.031628453489896E+000, -1.031628453489858E+000]

Wir wollen nun die Laufzeiten verschiedener Verfahren vergleichen. Wir verwenden dazu sogenannte Standardzeiteinheiten (auch STUs genannt). Bei der STU handelt es

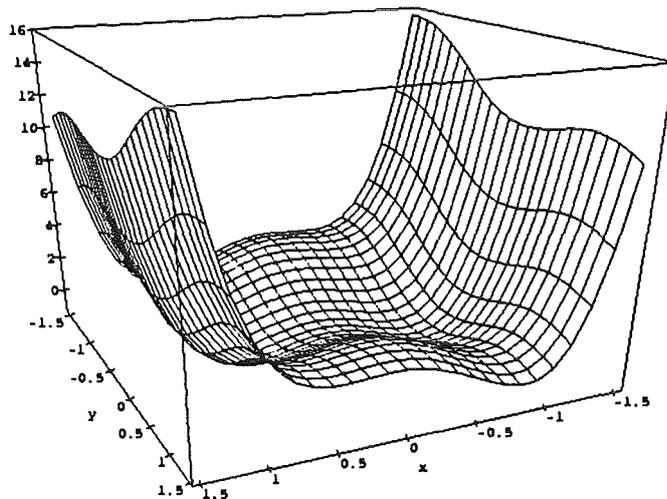


Abbildung 12: Six-Hump-Camel-Back-Funktion

sich um die sogenannte *Standard Time Unit*, die der Berechnungszeit für 1000 reelle Auswertungen der Shekel-5-Funktion (S5) entspricht. Sie wird als nahezu von Compiler und Rechner unabhängige Zeitmeßkonstante üblicherweise in der Literatur verwendet.

In der nachfolgenden Tabelle vergleichen wir die Verifikationsverfahren von Hansen [9], Jansson [13] und Ratz [20] mit klassischen Näherungsverfahren [23]. Bei letzteren ist der angegebene STU-Wert ein Mittel aus den Zeiten, die verschiedene dieser approximativen Verfahren benötigen (vgl. [23]).

Vergleich der Standardzeiteinheiten						
	H3	H6	S5	S7	S10	BR
Näherungsverfahren (im Schnitt)	11.5	29.1	15.4	19.4	22.2	6.2
Verifikationsverfahren (Hansen)	4.4	3641	2.0	6.4	10.2	0.7
Verifikationsverfahren (Ratz)	2.3	57.3	1.2	1.5	2.3	0.7
Verifikationsverfahren (Jansson)	5.6	40.1	1.5	1.8	2.3	2.2

Die Zahlen machen deutlich, daß die neuen Verfahren in ihrer Effizienz durchaus mit den Näherungsverfahren vergleichbar und diesen in den meisten Fällen sogar deutlich überlegen sind. Das ableitungsfreie Verfahren von Jansson [12], das Näherungen für den Minimalwert und die Minimalstellen sowie garantierte Schranken für den Minimalwert liefert, kann sogar mit noch kürzeren Laufzeiten aufwarten.

6.2 Weitere Testbeispiele

Wir wollen nun noch Resultate für einige weitere Testfunktionen aus [9] und [23] angeben. Darunter befinden sich auch extreme Beispiele mit mehreren 1000 oder sogar 10^{10} lokalen Minimalstellen im Optimierungsbereich.

In den jeweils nach den numerischen Ergebnissen folgenden Tabellen zu den Testfunktionen vergleichen wir für die Verfahren von Hansen [9], Ratz [20] und Jansson [14] die Anzahl der notwendigen Funktions-, Gradienten- und Hessematrixauswertungen sowie die Laufzeiten in Standardzeiteinheiten. Der STU-Wert für das Hansen-Verfahren ist ein aus den in [9] angegebenen Laufzeiten grob ermittelter Wert, da keine STU-Angaben vorliegen. Beim Verfahren von Jansson ist zu bemerken, daß aufgrund des Einsatzes approximativer Verfahren neben den Intervallauswertungen noch reelle Auswertungen notwendig sind, die jedoch in den Tabellen nicht aufgeführt sind.

L5: Die Levy-Funktion Nr. 5 ($x \in \mathbb{R}^2$):

$$f_{L5}(x) \sum_{i=1}^5 i \cos((i-1)x_1 + i) \sum_{j=1}^5 j \cos((j+1)x_2 + j) + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2.$$

Startbereich: $-10 \leq x_i \leq 10$, $i = 1, 2$ (darin: 760 lokale Minima!).

Ergebnisse:

Minimalstelle: $[-1.306853009753580E+000, -1.306853009753564E+000]$
 $[-1.424845041560682E+000, -1.424845041560681E+000]$

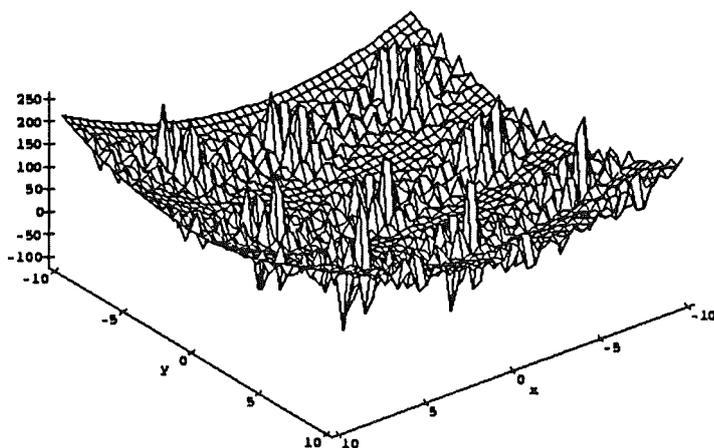


Abbildung 13: Die Levy-Funktion Nr. 5

Minimum: $[-1.761375780016305E+002, -1.761375780016283E+002]$

Die Abbildungen 13 und 14 zeigen die Levy-Funktion Nr. 5 im originalen Optimierungsgebiet und in einem etwas kleineren Ausschnitt davon.

Levy-Funktion Nr. 5			
$\epsilon = 10^{-12}$	Hansen	Ratz	Jansson
f -Auswertungen	2166	59	732
∇f -Auswertungen	2021	319	2
$\nabla^2 f$ -Auswertungen	725	69	11
Laufzeit in STUs	> 30	13.4	7.9

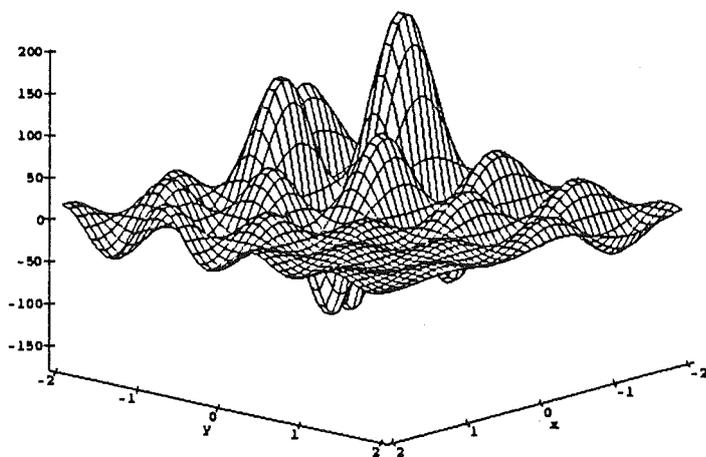


Abbildung 14: Die Levy-Funktion Nr. 5 (Detail)

L12: Die Levy-Funktion Nr. 12 ($x \in \mathbb{R}^{10}$):

$$f_{L12}(x) = \sum_{i=1}^9 y_i^2(1 + 10 \sin^2(\pi(1 + y_{i+1}))) + \sin^2(\pi(1 + y_1)) + y_{10}^2,$$

mit $y_i = (x_i - 1)/4, i = 1, \dots, 10$

Startbereich: $-10 \leq x_i \leq 10, i = 1, \dots, 10$ (darin 10^{10} lokale Minima!).

Ergebnisse:

Minimalstelle: [9.99999999999998E-001, 1.000000000000001E+000]
 [9.99999999999998E-001, 1.000000000000001E+000]
 ...
 [9.99999999999998E-001, 1.000000000000001E+000]
 [9.99999999999998E-001, 1.000000000000001E+000]

Minimum: [0.000000000000000E+000, 6.522368011415477E-030]

Abbildung 15 zeigt die Levy-Funktion Nr. 12 in nur zwei Dimensionen, so daß sich der Leser einen Eindruck über die Gestalt der Funktion machen kann.

Levy-Funktion Nr. 12			
$\varepsilon = 10^{-12}$	Hansen	Ratz	Jansson
f -Auswertungen	559	89	141
∇f -Auswertungen	497	177	1
$\nabla^2 f$ -Auswertungen	184	41	5
Laufzeit in STUs	> 34	20.1	4.7

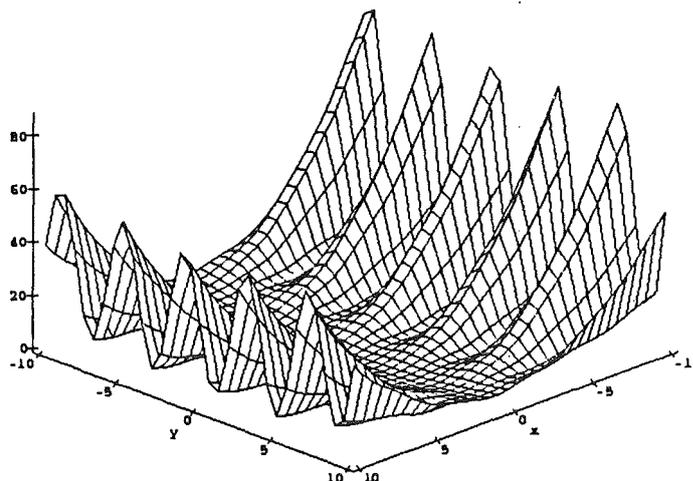


Abbildung 15: Die Levy-Funktion Nr. 12 mit $x \in \mathbb{R}^2$

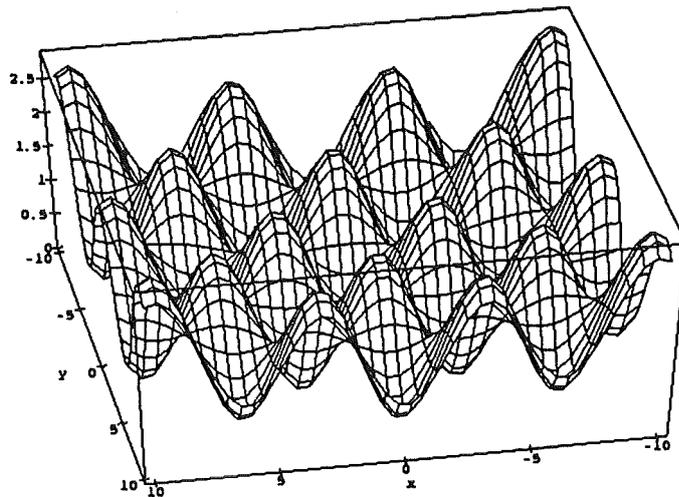


Abbildung 16: Die Griewank-Funktion für $n = 2$

Gn: Die Griewank-Funktionen ($x \in \mathbb{R}^n$):

$$f_{Gn}(x) = \sum_{i=1}^n \frac{x_i^2}{a_n} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

Startbereich: $-500 \leq x_i \leq 600$, $i = 1, \dots, n$ (darin für $n = 2$ etwa 500 und für $n = 10$ mehrere 1000 lokale Minima!).

Ergebnisse:

Minimalstelle: [0.0000000000000000E+000, 0.0000000000000000E+000]
 [0.0000000000000000E+000, 0.0000000000000000E+000]
 ...
 [0.0000000000000000E+000, 0.0000000000000000E+000]
 Minimum: [0.0000000000000000E+000, 0.0000000000000000E+000]

Auch die Griewank-Funktion läßt sich von Intervallverfahren effizienter bearbeiten als von klassischen Verfahren. Vergleicht man die in [23] aufgeführten Ergebnisse für die Methoden von Griewank und Snyman mit den Resultaten für die Verfahren von Jansson [14] und Ratz [20], so ist zunächst festzustellen, daß die verallgemeinerte Abstiegmethode von Griewank nur ein suboptimales lokales Minimum findet. Für $n = 2$ benötigt Snymans Multi-Start-Algorithmus etwa 1.4 STUs und für $n = 10$ bereits 90 STUs. Das Verfahren von Jansson benötigt für $n = 2$ etwa eine STU und für $n = 10$ rund neun STUs. Unser Verfahren aus [20] benötigt für $n = 2$ etwa zwei STUs und für $n = 10$ rund 16 STUs.

RB: Die Rosenbrock-Funktion ($x \in \mathbb{R}^2$):

$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$$

Startbereiche: $-1.2 \leq x_i \leq 1.2$, $i = 1, 2$ bzw. $-10^6 \leq x_i \leq 10^6$, $i = 1, 2$.

Ergebnisse:

Minimalstelle: [9.99999999999603E-001, 1.00000000000133E+000]
 [9.9999999999705E-001, 1.00000000000266E+000]

Minimum: [0.000000000000000E+000, 2.969873293021112E-023]

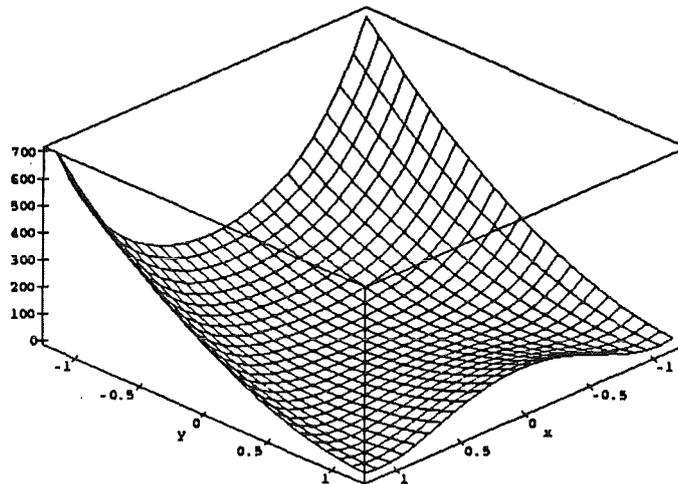


Abbildung 17: Die Rosenbrock-Funktion

Die Rosenbrock-Funktion wurde in der Literatur als eine *schwere* Testfunktion für neue Optimierungsverfahren akzeptiert. Eine ihrer Besonderheiten ist die Tatsache, daß ihre Hessematrix auf allen Punkten der Parabel $800x^2 - 800y + 4 = 0$ singularär ist.

Die nachfolgende Tabelle, in der wiederum die Verfahren von Hansen, Ratz und Jansson verglichen werden, demonstriert die (im Vergleich zur enormen Vergrößerung des Optimierungsbereiches) nur geringe Erhöhung der Auswertungszahlen und Laufzeiten, wenn der Startbereich um den Faktor 10^6 vergrößert wird.

Rosenbrock-Funktion						
$\varepsilon = 10^{-12}$	Hansen	Ratz	Jansson	Hansen	Ratz	Jansson
Startbereich X_i	[-1.2, 1.2]			[-10 ⁶ , 10 ⁶]		
f -Auswertungen	640	111	150	12321	1213	12172
∇f -Auswertungen	583	187	1	12827	2399	1
$\nabla^2 f$ -Auswertungen	238	50	13	4949	593	13
Laufzeit in STUs	> 2.5	0.2	1.5	> 50	4.1	35

7 Schlußbemerkung

Viele der Anwender und Entwickler numerischer Verfahren und entsprechender Software werden heutzutage immer noch vom Begriff Intervallrechnung geradezu abgeschreckt. Dies liegt wohl vor allem an der Tatsache, daß dieses relativ junge Gebiet der Mathematik,

aufgrund zahlreicher Fehlinterpretationen von einigen naiven Anwendern der Anfangszeit, mit einem schlechten Ruf zu kämpfen hat. So wird von vielen nach wie vor die Intervallrechnung als diejenige Sparte der Numerik abgetan, die „eigentlich den Wert 5 berechnen will, als Ergebnis aber eine Einschließung $[-1000, 1000]$ liefert“. Der Grund liegt darin, daß es bei naiver Anwendung der Intervallarithmetik zu drastischen Überschätzungen der Wertebereiche kommen kann.

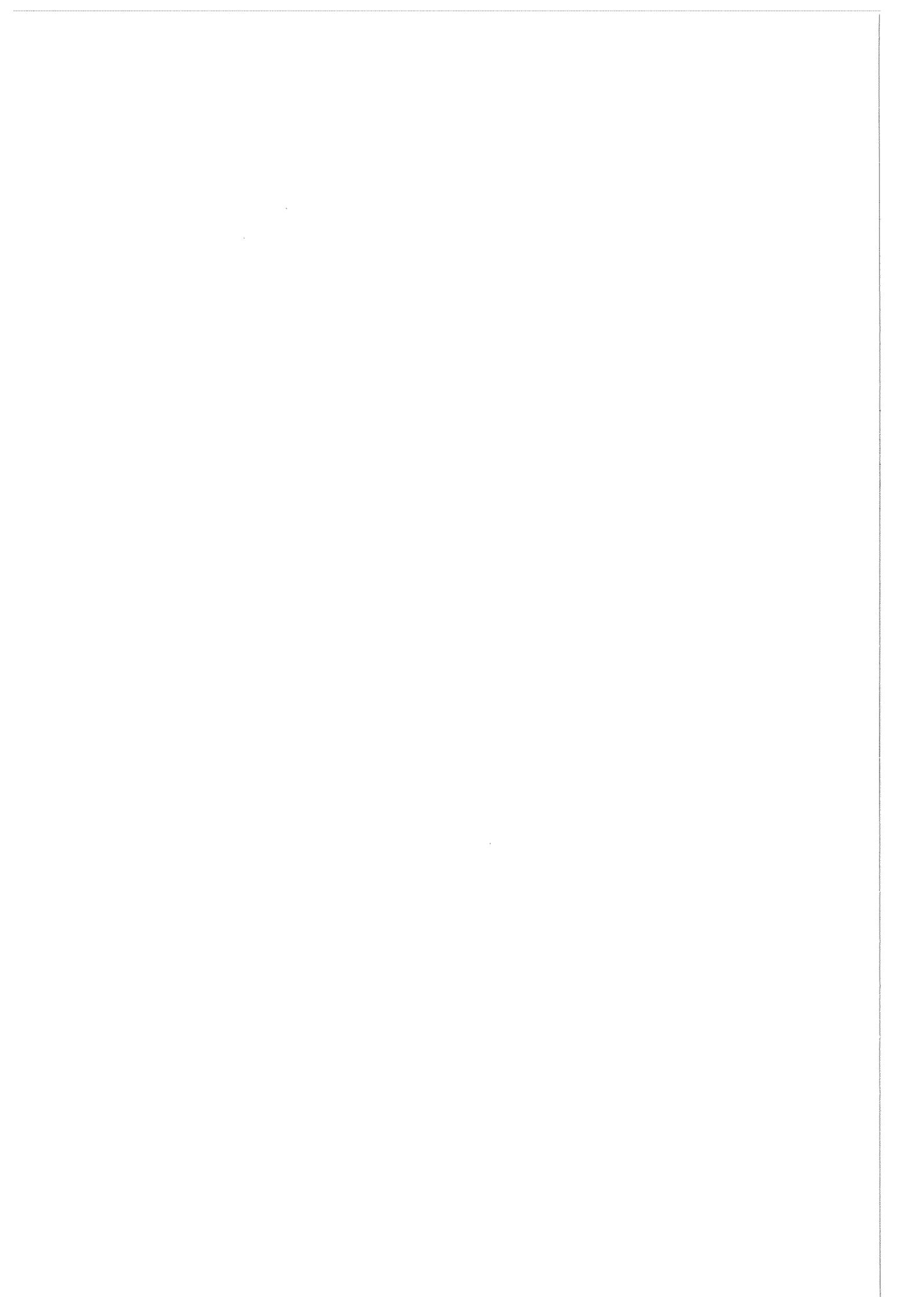
Wir hoffen, dieser Artikel weckt bei einigen Lesern, die sich vielleicht bisher eher zu den abgeschreckten zählten, zumindest ein gewisses Interesse an Intervallverfahren. Die vorangegangenen Abschnitte demonstrierten schließlich, daß es speziell auf dem Gebiet der globalen Optimierung hochgenaue und schnelle Intervallverfahren gibt, die sogar effizienter als klassische approximative Methoden arbeiten können. Darüber hinaus sind diese Verfahren in der Lage, mathematische Aussagen über den garantierten Einschluß der gesuchten Lösung(en) zu machen.

Die Grundlage für die Durchführung solcher Verfahren auf einem Rechner bildet eine mathematisch formulierte Rechnerarithmetik [16] und eine entsprechende Implementierung dieser Arithmetik, wie sie z. B. in PASCAL-XSC [15] realisiert wurde. Weitere Voraussetzung für eine erfolgreiche Entwicklung solcher Intervallverfahren ist natürlich stets, daß die eingangs angesprochenen Überschätzungs- und Aufblähungseffekte durch ein problemspezifisches und sorgfältiges Design der Algorithmen weitestgehend vermieden werden.

Literatur

- [1] Alefeld, G., Herzberger, J.: *Introduction to Interval Computations*. Academic Press, New York, 1983.
- [2] Csendes, T.: *Nonlinear Parameter Estimation by Global Optimization*. Acta Cybernetica, Tom 8 (Fasc. 4), 361–370, 1988.
- [3] Csendes, T.: *An Interval Method for Bounding Level Sets of Parameter Estimation Problems*. Computing 41, 75–86, 1989.
- [4] Csendes, T.: *Interval Methods for Bounding Level Sets: Revisited and Tested with Global Optimization Problems*. BIT 30, 650–657, 1990.
- [5] Dixon, L. C., Szegö, G. (Hrsg.): *Towards Global Optimization 2*. North-Holland, Amsterdam, 1978.
- [6] Hammer, R., Hocks, M., Kulisch, U., Ratz, D.: *Numerical Toolbox for Verified Computing I – Basic Numerical Problems*. Springer-Verlag, Heidelberg, New York, 1993.
- [7] Hansen, E.: *Global Optimization Using Interval Analysis – The One-Dimensional Case*. Journal of Optimization Theory and Applications 29, 331–344, 1979.
- [8] Hansen, E.: *Global Optimization Using Interval Analysis – The Multi-Dimensional Case*. Numerische Mathematik 34, 247–270, 1980.
- [9] Hansen, E.: *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 1992.

- [10] Ichida, K., Fujii, Y.: *An Interval Arithmetic Method for Global Optimization*. Computing **23**, 85–97, 1979.
- [11] Jansson, C.: *A Global Minimization Method: The One-Dimensional Case*. Technical Report 91.2, Forschungsschwerpunkt Informations- und Kommunikationstechnik, TU Hamburg-Harburg, 1991.
- [12] Jansson, C., Knüppel, O.: *A Global Minimization Method: The Multi-Dimensional Case*. Technical Report 92.1, Forschungsschwerpunkt Informations- und Kommunikationstechnik, TU Hamburg-Harburg, 1992.
- [13] Jansson, C.: *On Self-Validating Methods for Optimization Problems*. In: Herzberger, J.: *Studies in Computational Mathematics*, North-Holland, Amsterdam, erscheint 1994.
- [14] Jansson, C., Knüppel, O.: *Numerical Results for a Self-Validating Global Optimization Method*. Technical Report 94.1, Forschungsschwerpunkt Informations- und Kommunikationstechnik, TU Hamburg-Harburg, 1994.
- [15] Klatte, R., Kulisch, U., Neaga, M., Ratz, D., Ullrich, Ch.: *PASCAL-XSC – Sprachbeschreibung mit Beispielen*. Springer-Verlag, Heidelberg, 1991.
- [16] Kulisch, U., Miranker, W. L.: *Computer Arithmetic in Theory and Practice*. Academic Press, New York, 1981.
- [17] Moore, R. E.: *Interval Analysis*. Prentice Hall, Engelwood Cliffs, New Jersey, 1966.
- [18] Ratschek, H., Rokne, J.: *New Computer Methods for Global Optimization*. Ellis Horwood Limited, Chichester, 1988.
- [19] Ratz, D.: *An Inclusion Algorithm for Global Optimization in a Portable PASCAL-XSC Implementation*. In: Atanassova, L. und Herzberger, J. (Hrsg.): *Computer Arithmetic and Enclosure Methods*, 329–338. North-Holland, Elsevier, Amsterdam, 1992.
- [20] Ratz, D.: *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen*. Dissertation, Universität Karlsruhe, 1992.
- [21] Ratz, D.: *Box-Splitting Strategies for the Interval Gauss-Seidel Step in a Global Optimization Method*. Computing, Springer-Verlag, Wien, erscheint 1994.
- [22] Skelboe, S.: *Computation of Rational Interval Functions*. BIT **4**, 87–95, 1974.
- [23] Törn, A., Žilinskas, A.: *Global Optimization*. Lecture Notes in Computer Science, No. 350, Springer-Verlag, Berlin, 1989.
- [24] Wolfe, M. A.: *Numerical Methods for Unconstrained Optimization – An Introduction*. Van Nostrand Reinhold, New York, 1978.





Prof. Dr. Peter Mäder

Zur Person und zum Vortrag

Prof. Mäder studierte in Freiburg gleichzeitig Mathematik und kath. Theologie. Seit nunmehr 20 Jahren ist er am staatlichen Seminar für Schulpädagogik (Gymnasien) mit der Ausbildung von Referendaren betraut und, soweit es sein Deputat zuläßt, erteilt daneben selbst auch noch Unterricht an Gymnasien in Mathematik und Religion.

So wie sich diese beiden Tätigkeiten gegenseitig befruchten und ergänzen, so wirken wohl beide unterschiedliche Fachbereiche zusammen hin auf eine mehr erkenntnistheoretisch-philosophische Betrachtungsweise der Mathematik. Herr Mäder beschäftigt sich nämlich zusätzlich mit Aspekten der Entwicklung der Mathematik und hat dies auch bereits in Buchform dargelegt. Sein Buch ist 1992 im Metzler Schulbuchverlag erschienen und heißt "Mathematik hat Geschichte". Hier schlägt er nun ein neues Kapitel auf und zeigt am Beispiel der Ziffer 0 einmal mehr, daß sich die Selbstverständlichkeiten unserer Zeit auch erst mühsam und zögerlich entwickeln mußten.

Peter MÄDER
Brahmsstr. 6
D 79104 FREIBURG
Tel. 0761 / 57855

"Wie die Puppe ein Adler sein wollte, der Esel ein Löwe, die Äffin eine Königin - so wollte die Null eine Ziffer sein !"

EIN ÜBERBLICK ZUR GESCHICHTE DER ZAHL NULL

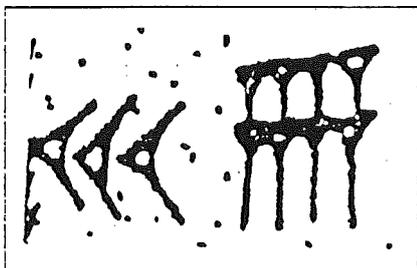
Das oben genannte Zitat (nach Menninger, II. Teil, S. 240; siehe das Literaturverzeichnis am Ende dieses Beitrags) stammt aus dem Frankreich des 15. Jahrhunderts und verdeutlicht es : tatsächlich war lange unverständlich, warum für etwas, das doch gar nicht vorhanden ist ("das Leere") auch noch ein eigenes Zeichen stehen sollte.

Auf der Suche nach der Zahl Null begegnen wir entscheidenden Epochen der Mathematikgeschichte.

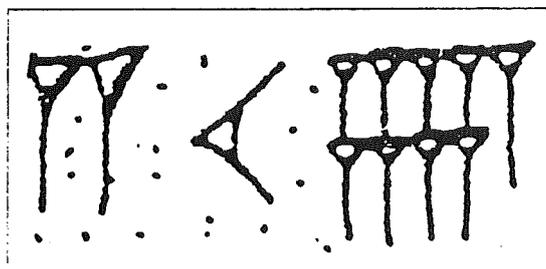
Das Lückenzeichen im Sechziger-System der Babylonier

Die ältesten uns bekannten mathematischen Dokumente sind bis zu 4000 Jahre alt; sie stammen aus dem Zweistromland zwischen den Flüssen Euphrat und Tigris. Archäologen entdeckten ganze Bibliotheken von Tontäfelchen (Keilschrift), und darunter waren auch Texte mit Zahlzeichen und mathematischen Abhandlungen.

So nennt etwa ein Tontäfelchen in einer Liste von Kleinvieh u. a. 38 kleine Lämmer und 139 Jungziegen :



3 Zehner + 8 Einer



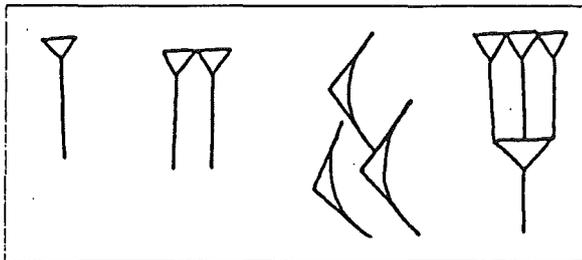
$$2 \cdot 60^1 + (1 \cdot 10 + 9 \cdot 1) \cdot 60^0$$

Die rechts stehende Darstellung wird nur verständlich, wenn man weiß, daß die damalige Zahlschreibweise im 60er-System (in einer Kombination mit dem Zehnersystem) erfolgte. Die senkrecht stehenden Keile stehen für die Einer, die schrägen Keile für die Zehner.

So sieht dann z.B. die Zahl 3754 so aus :

$$1 \cdot 60^2 + 2 \cdot 60^1 + (3 \cdot 10 + 4) \cdot 60^0$$

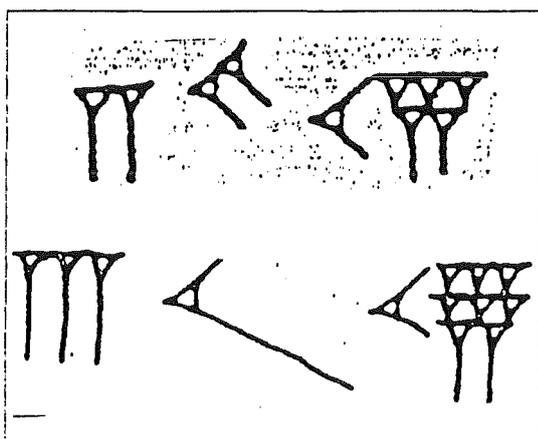
$$= 3600 + 120 + 34 = 3754$$



Die "babylonische Mathematik" im weitesten Sinne umfaßt die Epochen der Sumerer, der Akkader (ab 2400 v. Chr.), die babylonischen Dynastien (ab dem 18. Jhdt. v.) und die nachfolgenden Zeiten bis hin zu den Seleukiden (bis zum 1. Jhdt. v. Chr.).

Wie aber ist erkennbar, ob bei dieser Zahldarstellung gewisse Potenzen von 60 ausgelassen wurden, weil sie in der Darstellung der betreffenden Zahl nicht vorkommen ? Nur aus dem Kontext und aus der Art und Weise, wie die Zeichen gruppiert werden, ist hier eine Antwort möglich. Denn ein Lückenzeichen, eine Kennzeichnung der Null, fehlt in diesem System !

Belege für ein Lückenzeichen gibt es dann aber tatsächlich aus den letzten vorchristlichen Jahrhunderten (und das war nicht mehr die große Zeit der babylonischen Kultur). Zwei kleine schräge Haken oder ein langgezogener schräger Keil konnten die Lücke kennzeichnen :



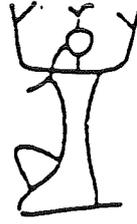
$$2 \cdot 60^2 + 0 \cdot 60^1 + 15 \cdot 60^0 = 7215$$

$$3 \cdot 60^2 + 0 \cdot 60^1 + 18 \cdot 60^0 = 10818$$

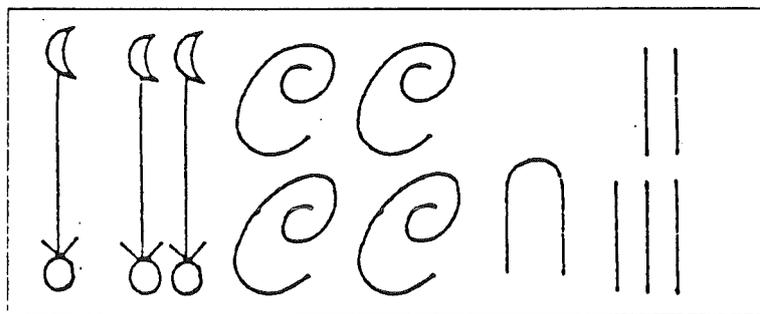
Daß das Lückenzeichen auch ganz rechts stehen kann (als Koeffizient von 60^0), ist nur in astronomischen Tafeln aus jener Zeit bezeugt.

Die Hieroglyphen der ägyptischen Mathematik

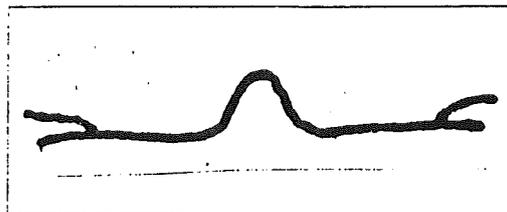
Die ägyptische Mathematik hatte ihre je eigenen Bilder (deren Bedeutung allerdings nicht immer ganz geklärt ist) für die Zehnerpotenzen. Ein Strich stand für 1, eine Fessel für 10, ein Strick für 100, eine Lotuspflanze für 1000, ein gestreckter Finger für 10000, eine Kaulquappe für 100000 und eine Gottheit für eine Million. Eine Kennzeichnung der Null war somit nicht erforderlich.

1	10	100	1.000	10.000	100.000	1.000.000
						

Die Zahl 3415 :



Vom Horus-Tempel in Edfu (in Oberägypten) gibt es einen interessanten Beleg aus dem 2. Jhdt. v. Chr. : da wird eine (allerdings nicht ganz eindeutig geklärt) Formel zur Berechnung von Viereckflächen, die mit den vier Seitenlängen arbeitet, angegeben. Bei der Anwendung dieser Formel auf Dreiecke wird einfach eine Seitenlänge des Vierecks auf Null gesetzt. Und dafür gibt es ein eigenes Zeichen : die ausgestreckten leeren Hände.



Dem griechischen Alphabet entlang

Die Griechen benutzten in der Antike (und dann auch noch weit über sie hinaus) einfach die Buchstaben ihres Alphabets zur Zahldarstellung. Drei aus dem Semitischen stammende Buchstaben kamen als Zeichen für 6, für 90 und für 900 noch hinzu.

1	α	10	ι	100	ρ
2	β	20	κ	200	σ
3	γ	30	λ	300	τ
4	δ	40	μ	400	υ
5	ε	50	ν	500	φ
6	ς	60	ξ	600	χ
7	ζ	70	ο	700	ψ
8	η	80	π	800	ω
9	θ	90	ς	900	Ϟ

Weil ein Zeichen für die Null fehlte, mußte immer weiter dem Alphabet entlanggegangen werden (vgl. die Kennzeichnung von 10, 20, 30, ... , 100, 200, ...). Die Zahl 12 etwa konnte so auch nicht in der Form $\alpha\beta$ ($1 \cdot 10 + 2$) geschrieben werden; sie wurde als $\iota\beta$ ($10 + 2$) dargestellt.

Tausender konnten durch einen vorgesetzten Haken gekennzeichnet werden :
' α = 1000, ' β = 2000 usw.

Ptolemäus (100 - 160 n. Chr.) setzte dann allerdings in seinem astronomischen Werk "Almagest" (= die große Zusammenfassung) ein Lückenzeichen ein :

$$\overline{\mu\alpha} \circ \overline{\iota\eta} = 41^{\circ} 00' 18'' .$$

Ob dieses Lückenzeichen (der kleine Kreis) für das griechische Wort οὐδέν (= nichts) steht, ist allerdings umstritten. Für die byzantinische Zeit aber ist eine Verwendung des omikron (ο) in diesem Sinne gesichert.

Hebräische Buchstaben als Zahlzeichen

Im Grunde gingen die Hebräer wie die Griechen vor. Auch sie nahmen die Buchstaben ihrer Schrift, um Zahlen zu kennzeichnen :

ט	ח	ז	ו	ה	ד	ג	ב	א
9	8	7	6	5	4	3	2	1
צ	פ	ע	ס	נ	מ	ל	כ	י
90	80	70	60	50	40	30	20	10
				ת	ש	ר	ק	
				400	300	200	100	

Hebräisch wird (wie auch Arabisch) von rechts nach links geschrieben und gelesen. Bei einer dreistelligen Zahl des Dezimalsystems stehen also die Hunderter ganz rechts, die Einer ganz links. Die Araber schreiben allerdings ihre Zahlzeichen - anders als ihre Texte - in der uns vertrauten Anordnung (die Einer stehen rechts).

Entscheidend ist, daß auch bei den Hebräern ein Zeichen für die Null fehlt und ebenso wie bei den Griechen kein Positionssystem vorliegt. Deshalb reichten eben zehn Zeichen (als Ziffern) nicht aus, sondern es mußte im Alphabet ständig weiter fortgeschritten werden.

Eine theologische Bedeutung führte zu einer besonderen Darstellung der Zahlen 15 und 16. Die Zahlzeichen für 10, 5 und 6 (vgl. dazu die Übersicht) kommen als Buchstaben im Gottesnamen JAHWE (2 Mose 3,14 : "Ich werde mit und für euch da sein") vor. Um diese Buchstaben des Gottesnamens nicht in einen profanen mathematischen Zusammenhang zu bringen, wurden die Zahlen 15 und 16 anders kombiniert : $6 + 9$ und $7 + 9$:

Die nicht besetzte Spalte auf dem Rechenbrett

Die Idee, einfach Buchstaben des Alphabets als Zahlzeichen zu verwenden, findet sich auch bei anderen Völkern. Auch in Rußland war - sicher bedingt durch den byzantinischen Einfluß - sehr lange eine alphabetische Zahlenschreibweise im Gebrauch.

Entscheidend sind dabei nicht die Zeichen (Ziffern könnten durchaus wie Buchstaben aussehen), der gemeinsame mathematische Hintergrund liegt vielmehr in der fehlenden Null. So braucht man immer neue Ziffern, das vorgegebenen Alphabet reicht noch nicht einmal dafür aus.

Solche alphabetischen Buchstaben-ziffern waren für das schriftliche Rechnen genauso ungeeignet wie die römischen Ziffern. Allerdings wurden diese Schreibweisen auch gar nicht für Rechnungen verwendet. Denn dazu gab es das Rechenbrett, den Abakus.

Es gibt viele Varianten dieses Rechenbretts bis in die Gegenwart hinein (vor allem in Rußland und Ostasiens). Auch die Form des Rechentisches war möglich.

Das Rechenbrett und der Rechentisch lösen das Problem der Null auf eine sehr direkte Weise : die zugehörige Spalte (oder Zeile) bleibt einfach unbesetzt.

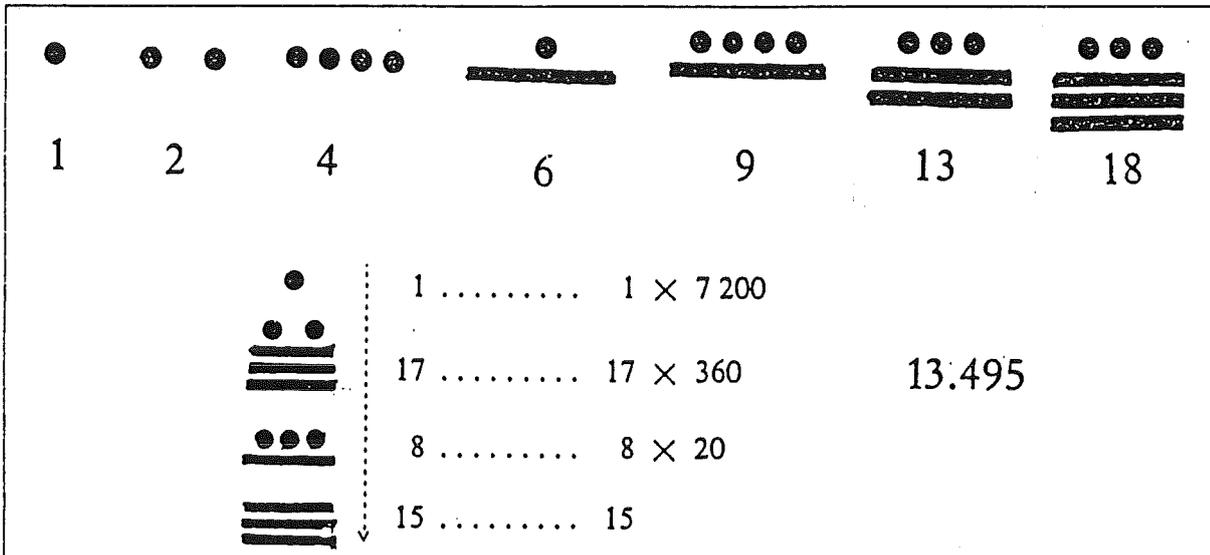


Die leere Muschel der Mayas

Die Mayas bewohnten ein Gebiet, das heute zum Süden Mexikos, zu Guatemala, Honduras und El Salvador gehört. Die Blütezeit ihrer Zivilisation ist um das 10. Jhdt. n. Chr. anzusetzen. Man weiß wenig über die Geschichte dieser Völkerfamilie, zumal fast alle der im 16. Jhdt. noch erhaltenen Schriften von den spanischen Eroberern brutal vernichtet wurden.

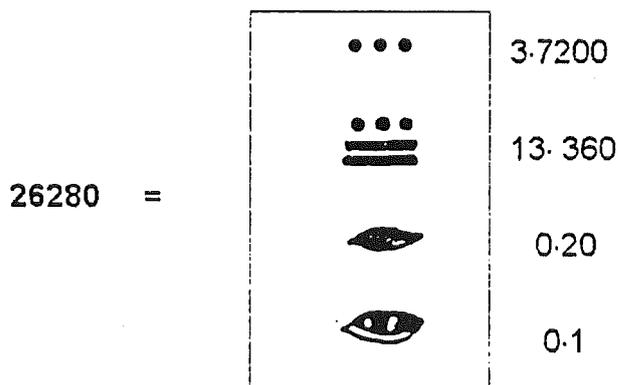
Ganz herausragend waren die mathematischen und astronomischen Kenntnisse der Mayas. Zur Darstellung der Zahlen kannten sie verschiedene Zeichen : Götter-, Menschen- und Tierköpfe oder auch nur ein einfaches graphisches System, das allein

mit Punkten und Strichen auskam. Stufenzahlen konnten die Zahlen 1, 20, 18·20 (= 360), 18·20² (= 7200) sein.



Auch die 20er-Potenzen 1, 20, 400, 8000 waren als Stufenzahlen möglich. Bemerkenswert für uns ist aber vor allem, daß bei den Mayas ein Zeichen für die Null existierte : eine leere Muschel oder ein leeres Schneckenhaus. Die Mayas hatten dafür den Namen "xok' ol" .

Ein Text aus dem Codex Dresdensis (in Dresden aufbewahrt) zeigt die Verwendung dieses Nullzeichens :



Das Lückenzeichen bei den Chinesen

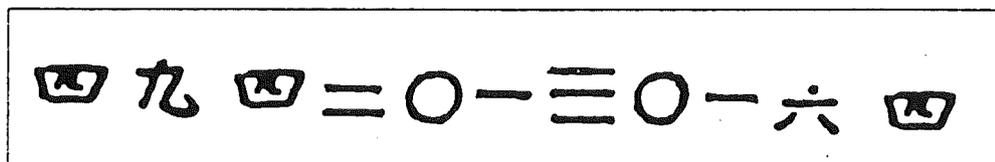
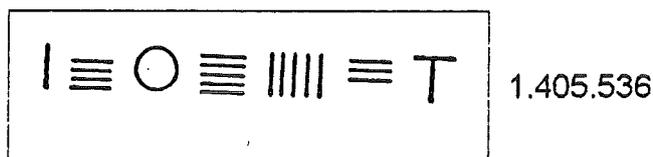
Sehr alte, in Europa lange Zeit kaum bekannte mathematische Traditionen auf hohem Niveau haben ihre Wurzeln in den Kulturen des Fernen Ostens und Südostasiens. Das älteste zusammenfassende Werk der chinesischen Mathematik, das in seinen ältesten Teilen bis ins 3. Jahrhundert v. Chr. zurückreichen könnte, ist die "Mathematik in neun Büchern".

Unterschiedliche chinesische Zahlschriften haben sich in der geschichtlichen Entwicklung herausgebildet :

一	壹	丨	一	丨	1
二	貳	丨丨	二	丨丨	2
三	參	丨丨丨	三	丨丨丨	3
四	肆	丨丨丨丨	四	丨丨丨丨	4
五	伍	丨丨丨丨丨	五	丨丨丨丨丨 〇	5
六	陸	丨丨丨丨丨丨	六	丨	6
七	柒	丨丨丨丨丨丨丨	七	丨丨	7
八	捌	丨丨丨丨丨丨丨丨	八	丨丨丨	8
九	玖	丨丨丨丨丨丨丨丨丨	九	丨丨丨丨	9
十	拾	丨丨丨丨丨丨丨丨丨丨	十	丨〇	10
百	百	丨丨丨丨丨丨丨丨丨丨丨	百	丨〇〇	100
千	仟	丨丨丨丨丨丨丨丨丨丨丨丨	千	〇	1000
萬	萬	丨丨丨丨丨丨丨丨丨丨丨丨丨	萬	〇	10000
		丨丨丨丨丨丨丨丨丨丨丨丨丨丨		〇	0

Die Strichziffern hängen offensichtlich mit einer ursprünglichen Verwendung von Rechenstäbchen auf dem Rechenbrett zusammen. Kam eine bestimmte Stelle bei einer Zahl gar nicht vor, so hat man wohl zunächst an dieser Stelle einfach eine Lücke (ohne eigenes Zeichen) gelassen. Ein in China arbeitender indischer Astronom erwähnte im 8. Jhdt. n. Chr., daß für die Leerstelle auf dem Rechenbrett ein Punkt gesetzt wurde. Aus dem Jahr 1247 ist uns das erste gedruckte Nullzeichen in Form eines Kreises überliefert.

Zahldarstellungen in einem dezimalen Positionssystem waren nun ohne weiteres möglich.



Die Zahl 4,9420130164 (= log 87501) entstammt einer chinesischen Logarithmentafel von 1713.

Es ist historisch ungeklärt, inwieweit die chinesische Mathematik bei der Entwicklung des dezimalen Positionssystem mit dem Zeichen für die Null selbständig gearbeitet hat oder ob indische Einflüsse mitwirkten.

Der entscheidende Durchbruch in der indische Mathematik

Es war der nicht hoch genug einzuschätzende Beitrag der indischen Mathematik, ein heute weithin in der ganzen Welt übernommenes Stellenwertsystem geschaffen zu haben, das mit zehn Ziffern einschließlich der Null auskommt (und sicher gab es ähnliche Entwicklungen auch in anderen ostasiatischen Kulturen). Daß die äußere Form der Ziffern, ihre Schreibweise, sich dann auf dem Weg über die arabisch-islamischen Länder bis hin nach Europa vielfach verändert hat, spielt dabei keine Rolle.

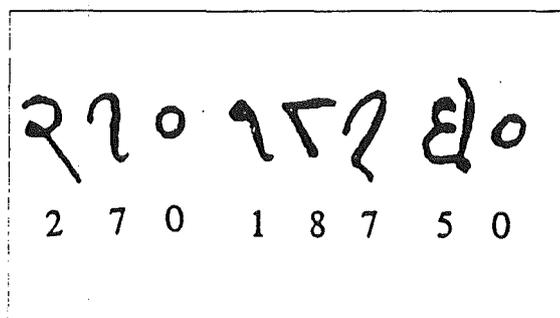
Die Mathematik war in Indien eine seit altersher hoch angesehene Wissenschaft. Eine der ersten Quellen indischer Mathematik sind die "Schnurregeln", die bis ins 2. Jahrtausend v. Chr. zurückgehen : Anweisungen zum Spannen von Schnüren beim Ausmessen von Altären und Tempeln.

Ab dem 3. Jhdt. v. Chr. setzten sich in der Zahlschreibweise immer mehr die Brahmi-Ziffern durch. Zunächst kam dieses System keineswegs nur mit zehn Ziffern aus; tatsächlich mußte es - damit durchaus mit den Buchstaben-Ziffern der Griechen, Hebräer u.a. vergleichbar - zusätzliche Zeichen für 10, 20, 30, ..., 100 usw. einführen.

—	=	≡	𑀓	𑀔	𑀕	𑀖	𑀗	𑀘
1	2	3	4	5	6	7	8	9
𑀠	𑀡	𑀢	𑀣	𑀤	𑀥	𑀦	𑀧	𑀨
10	20	30	40	50	60	70	80	90
𑀩	𑀪	𑀫	𑀬	𑀭	𑀮	𑀯	𑀰	
100	200	500	1000	4000	70000			

Doch die indische Mathematik unternahm dann schon in den ersten nachchristlichen Jahrhunderten erfolgreiche Versuche, allein mit den ersten neun Ziffern und der Null alle anderen Zahlen zu schreiben. Damit war der entscheidende Schritte gelungen ! Indische Mathematiker des 5. und 6. Jahrhunderts waren bereits mit dem dezimalen Positionssystem vertraut und rechneten mit der Null. Die Kunde von dieser neuen Art, Zahlen zu schreiben, drang auch nach Westen vor (als ältester Beleg dafür gilt ein Hinweis von 662 aus Syrien).

Das bis heute älteste Zeugnis einer geschriebenen Null (in der Form eines kleinen Kreises) findet sich auf der Wand eines kleinen Tempels in Gwalior (bei Lashkar in Mittelindien). Es handelt sich um eine Schenkungsurkunde von 870 n. Chr..



Es existieren aber Zeugnisse außerhalb Indiens, die älter als die frühesten uns bekannten schriftlichen indischen Dokumente sind.

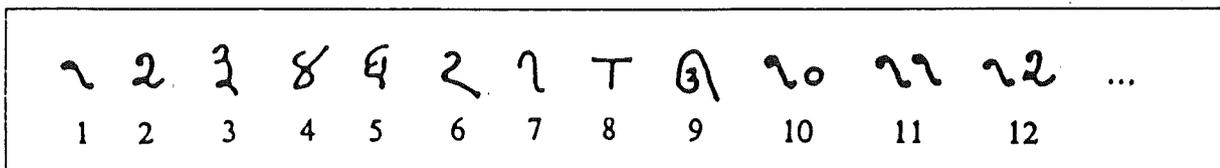
Das sind Inschriften aus dem 7. Jhdt. n. Chr. :



605 608 606
Kambodscha Sundainsel Banka Sumatra

Und so gibt es die bis heutige nicht eindeutig beantwortete Frage, wie eigenständig nun wirklich der Beitrag der Inder war und in welcher Weise Entlehnungen aus anderen asiatischen Kulturen eine Rolle bei der Entwicklung dieses dezimalen Positionssystems mit der Null gespielt haben.

In manchen Formen hatten bereits die indischen Ziffern des 9. Jahrhunderts eine gewisse Ähnlichkeit mit den uns vertrauten und geläufigen Ziffern.



(Numerierung eines Gedichts mit 26 Strophen, 875 n. Chr.)

Schwierigkeiten bereiteten den indischen Mathematikern zunächst gewisse Rechnungen mit der Null. So ging Brahmagupta (geboren 598) noch von $a : 0 = 0$ aus; später verband sich aber damit doch die Vorstellung von einer gleichsam unendlich großen Zahl. Für Bhaskara II. (geboren 1115) blieb $a : 0$ (mit von Null verschiedenem a) unverändert, "was man hier auch hinzufügen oder abziehen möge". Unklar blieb allerdings die Frage von "Null durch Null".

Die Übernahme der neuen Zahlzeichen durch die Araber

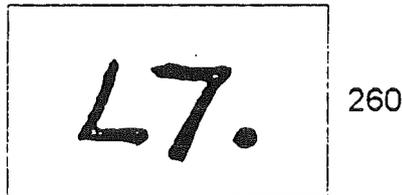
Die sehr rasche Ausbreitung des Islams ab dem 7. Jahrhundert hatte auch ein starkes Erblühen der Wissenschaften und Künste zur Folge. Zwar kannte zuweilen auch der Islam wissenschaftsfeindliche Tendenzen, im allgemeinen aber förderte er sehr die kulturellen Entwicklungen.

So waren die Araber über Jahrhunderte hinweg das führende Kulturvolk in weiten Teilen Asiens und des Mittelmeerraums. Sie haben das Wissen der griechischen Antike, das sonst vielleicht in Vergessenheit geraten wäre, nach Europa tradiert (so

etwa die Philosophie des Aristoteles); islamische Mathematiker beschäftigten sich in ihren Kommentaren mit dem Werk Euklids u.a. Über Handelsbeziehungen - bis hin nach Indien und China - kamen die Araber mit den Wissenschaften Ostasiens in Berührung.

Einer der einflußreichsten dieser Mathematiker war Al-Chwarizmi (um 850). Über den Titel seines berühmten Werks zur Gleichungslehre hat er das Wort "al-gabr" (das "Rückversetzen"; gemeint sind Äquivalenzumformungen) als das Fachwort "Algebra" in die Mathematik eingeführt. Bedeutsam für die Übernahme der in Indien kennengelernten Ziffern war dann vor allem sein "Buch über die Addition und Subtraktion nach der Rechenweise der Inder". Erstmals hat hier ein arabisches Werk das dezimale Positionssystem aufgegriffen und weitergegeben : "Wir haben uns entschlossen, die Rechenweise der Inder mit Hilfe von neun Zeichen zu zeigen". Al-Chwarizmi spricht zunächst nur von neun Ziffern, doch zuweilen findet sich in seinem Buch ein kleiner Kreis für die Null, deren Bedeutung er ausdrücklich betont. Die Ziffern selbst hat Al-Chwarizmi wohl nicht in seinen Text (der uns nur in einer Übersetzung aus dem 13. Jhdt. vorliegt) eingetragen. Ohnehin war es in der damaligen arabischen Mathematik weithin üblich, Zahlen in ihrer Wortform anzugeben.

Als das älteste uns überlieferte arabische Dokument für indische Ziffern (mit der Null) gibt ein Papyrus mit der Jahreszahl 260 nach der Hedschra, der Flucht (Übersiedlung) Mohammeds von Mekka nach Medina, die 622 stattfand :



Die neue Schreibweise mit den Ziffern der Inder setzte sich erst allmählich durch und konnte andere Zahlensysteme nur langsam verdrängen. Ab dem 12. Jhdt. wurden die neuer Ziffern innerhalb der arabischen Kultur weithin allgemein anerkannt.

Im westlichen Teil des arabischen Kulturgebiets, vor allem in Marokko, erschienen diese neuen Ziffern in anderer Gestalt. So gibt es die ostarabischen und die westarabischen Traditionen. In der westarabischen Form sind diese Ziffern dann auch nach Spanien und ins übrige Europa gelangt.

Die heutigen ostarabischen Ziffern zeigt ein Kalenderblatt aus dem Jahr 1974 :

كانون الاول ١٩٧٤

الاحد	الاثنين	الثلاثاء	الاربعاء	الخميس	الجمعة	السبت
١	٢	٣	٤ قدسيا ياربا قدس وسنا صحتنا	٥ القدس سنا شيع بيانا وامي الا ريتيا	٦ اول جمادى من القدر قدس القدر من القدر	٧
٨	٩ سبل سنا وقره الا	١٠ بد. قناتا بيلا	١١	١٢	١٣	١٤
١٥ اص الا يمد	١٦	١٧	١٨	١٩	٢٠ قنا حه اليلا	٢١
٢٢ اص قنا	٢٣	٢٤ اركون القدر سور وقناتا	٢٥ حبه اليلا اليلا	٢٦ بنت صغره اليلا	٢٧	٢٨
٢٩ قدس وسنا	٣٠	٣١				

Die Zeilen dieses Kalenderblatts sind dabei von recht nach links zu lesen; die Zahlen sind so aufgebaut, wie wir das gewohnt sind (die Einer stehen ganz rechts). Der Unterschied liegt allein in der äußeren Gestalt der Ziffern.

Diese nachfolgende Abbildung dokumentiert Belege aus Indien (die ersten vier Zeilen), aus der ostarabischen (Zeilen 5 und 6) und der westarabischen Tradition (7, 8, 9). Das Beispiel 9 stammt aus Granada.

	1	2	3	4	5	6	7	8	9	0
1.	—	=	≡	∴	⋈	⊖	∩	∪	∞	
2.	∞	3	3	⊖	∩	∪	∞	∞	∞	•
3.	∩	∪	∞	∞	∞	∞	∞	∞	∞	∞
4.	∞	∩	∪	∞	∞	∞	∞	∞	∞	∞
5.	1	2	3	4	5	6	7	8	9	0
6.	1	2	3	4	5	6	7	8	9	0
7.	1	2	3	4	5	6	7	8	9	0
8.	1	2	3	4	5	6	7	8	9	0
9.	1	2	3	4	5	6	7	8	9	0

Wenn wir heute unsere Zahlformen als "arabische Ziffern" bezeichnen, so meinen wir damit die Übernahme der indisch-arabischen Ziffern aus der westarabischen Tradition. Dort haben sie die uns vertraute Gestalt.

Der Weg der indisch-arabischen Ziffern (mit der Null !) nach Europa

Die älteste uns bekannte Verwendung der indisch-arabischen Ziffern in Europa (noch ohne die Null) findet sich in einer Handschrift des nordspanischen Klosters Albeida aus dem Jahr 976. Zu den frühesten Zeugnissen in Deutschland zählt ein Codex des Klosters Salem (um 1200 geschrieben); er enthält Auszüge aus der "Algebra" Al-Chwarizmis. Aber die in Europa üblichen Mittel zur Zahldarstellung waren damals noch die römischen Ziffern und der Abakus (abgesehen von den byzantinischen Einflüssen in Osteuropa).

Durch Handel und wissenschaftliche Kontakte war man aber im Abendland inzwischen auch außerhalb der vom Islam besetzten Länder mit der arabischen Kultur in Berührung gekommen. Über Spanien (in Andalusien lagen bis Ende des 15. Jahrhunderts wichtige Zentren der maurisch-arabischen Welt) und Sizilien (878 - 1091 unter arabischer Herrschaft) waren die Null und die anderen indisch-arabischen Ziffern nach Europa gelangt. Einen ganz entscheidenden und sehr weitreichenden Beitrag dazu leistete Leonardo von Pisa (Fibonacci), der auf Reisen bis hin Syrien und Ägypten die arabische Mathematik kennengelernt hatte : "Unter wunderbarer Anleitung wurde ich in die Kunst der neuen indischen Ziffern eingeführt". 1202 schrieb Leonardo von Pisa sein "Buch des Abakus" (*liber abaci*). Dieses Werk ist trotz des mißverständlichen Titels gerade kein Buch über den Abakus ("ars abaci" war in Italien mittlerweile zur Bezeichnung von "Rechenkunst" überhaupt geworden), sondern eine klare Absage an das Rechenbrett und ein Bekenntnis zur Arithmetik mit den indisch-arabischen Ziffern. Nicht nur den Kreis der Gelehrten, sondern etwa auch für Kaufleute hatte Leonardo geschrieben.

Und so wurden die römischen Ziffern und das Rechnen auf dem Abakus immer mehr zurückgedrängt. 1338 bestanden in Florenz bereits sechs Schulen zur Ausbildung der Kaufleute im Sinne der modernen Arithmetik (obwohl 40 Jahre zuvor den Geldwechslern noch die Verwendung der neuen Ziffern verboten worden war).

Der englische Mathematiker und Astronom Johannes de Sacrobosco (um 1200 - 1256) schrieb eine vielbeachtete und bis ins 17. Jhdt. immer wieder neu aufgelegte Einführung in die indisch-arabische Zahlschrift : "Man muß nun wissen, daß es gemäß der 9 Einheiten 9 geltende Zahlzeichen (9 bedeutende Figuren) gibt, die die Einer darstellen. Eine zehnte heißt theca oder circulus (Kreis) oder cifra oder Figur des Nichts (*figura nihili*), weil sie nichts bedeutet. Doch sie gibt an der (richtigen) Stelle den anderen Figuren (höheren) Wert." (zitiert nach Menninger, II, Teil, S. 217).

Der um 1240 in Paris lebende Mönch Alexandre de Villedieu verfaßte sogar ein Gedicht (*carmen*), um den Gebrauch der neuen Zahlzeichen zu fördern. Diese Verse

wurden weit verbreitet und sogar ins Isländische übersetzt : "Diese neue Kunst heißt Algorismus, in der wir aus diesen zweimal fünf Ziffern 0 9 8 7 6 5 4 3 2 1 der Inder Nutzen zu ziehen." (zitiert nach Menninger , II. Teil, S. 227)

Leonardo von Pisa hatte von "cephirum" gesprochen und dabei in lateinischer Form das arabische Wort "sifr", das für "Null; Nichts" steht, verwendet. Die arabischen Mathematiker hatten dieses Wort für die Bezeichnung des neuen Lückenzeichens gewählt und dabei die Bedeutung des indische "sunya" (=leer) übernommen (vgl. damit das xok 'ol für die leere Muschel der Mayas).

Bereits im 12. Jhdt. waren in Europa als Übernahme von "sifr" die Bezeichnungen "cifra" und "cifra" aufgetaucht. Daraus wurde dann das "zero" im Italienischen und in anderen romanischen Sprachen. Carl Friedrich Gauß nannte in einer Abhandlung von 1799 die Null noch "cifra".

Das deutsche Wort "Ziffer" kommt aus dieser arabischen Wurzel. Ursprünglich nur für die Null vorgesehen, erfuhren "Ziffer" und das französische Wort "chiffre" dann immer mehr eine Bedeutungserweiterung zur Benennung sämtlicher Zahlzeichen von 0 bis 9. Lateinische Handschriften des 12. und 13. Jahrhunderts sprechen auch von "circulus" ("kleiner Kreis"), "nihil" oder "figura nihili" ("Darstellung des Nichts"). Das deutsche Wort "Null" hat sich ab dem 18. Jhdt. durchgesetzt, nachdem zuvor noch "nulla" ("nichts") mit der lateinischen Endung üblich war.

Es hatte bis über das Ende des Mittelalters hinaus gedauert, bis diese indisch-arabischen Ziffern mit der Null andere Darstellungs- und Rechenformen endgültig verdrängen konnten. In der Übergangsphase kamen zuweilen in ein- und derselben Zahl indisch-arabische, römische und sogar griechische Ziffern vor, z.B. $1 \cdot 5 \cdot \text{IIII} = 1504$ oder $\text{C}\delta = 104$ (δ ist der vierte Buchstabe des griechischen Alphabets) oder $\text{IV}02 = 1502$, $\text{M} \cdot \text{DC} \cdot \text{Z4} = 1624$ u.a.

Aber nicht nur bei den Gelehrten, in Klöstern und Universitäten, und nicht nur bei Kaufleuten war der Durchbruch der neuen Ziffern mit der Null nicht mehr aufzuhalten. In Deutschland hatten einen wesentlichen Anteil daran die Rechenbücher des 16. Jahrhunderts, vor allen die von Adam Ries verfaßten Schriften.

Ein Merkgedicht aus dem 15. Jhdt. (nach Menninger, II. Teil, S, 257) weist ganz deutlich auf die Verwendung der neuen Ziffern mit der Null hin :

"... mit den kanstu recht numeriren
All zuahl aussprechen und volführen."

Grundlegende Literatur :

GERICKE, Helmuth : Mathematik in Antike und Orient. Mathematik im Abendland
(Zusammenfassung zweier Bände in einem Buch).
Fourier Verlag, Wiesbaden 1992

IFRAH, Georges : Universalgeschichte der Zahlen.
Campus Verlag, Frankfurt / New York 1986

JUSCHKEWITSCH. A. P. : Geschichte der Mathematik im Mittelalter.
Teubner, Leipzig 1964

MENNINGER, Karl : Zahlwort und Ziffer - eine Kulturgeschichte der Zahl
(Zusammenfassung zweier Bände in einem Buch).
Vandenhoeck & Ruprecht, Göttingen, 3. Auflage 1979

TROPFKE, Johannes : Geschichte der Elementarmathematik
(vollständig neu bearbeitet von K. Vogel, K. Reich, H. Gericke),
Band 1 : Arithmetik und Algebra.
De Gruyter, Berlin und New York, 4. Auflage 1950

Weitere, systematisch angeordnete Literaturangaben zur Geschichte der Mathematik
bei :

MÄDER, Peter : Mathematik hat Geschichte, Metzler Schulbuch (Schroedel),
Hannover 1992

Abbildungsnachweis :

Seite 1 : Ifrah, S. 216

Seite 2 : Mäder, S. 3 ; Ifrah, S. 423, S. 421

Seite 3 : Hans Wußing, Vorlesungen zur Geschichte der Mathematik.
Deutscher Verlag der Wissenschaften, Berlin 1979, S. 35;
Gericke, 1. Band, S. 59

Seite 6 : Menninger, II. Teil, S. 153

Seite 7 : Ifrah, S. 472, S. 473

Seite 8 : Menninger, II. Teil, S. 274

Seite 9 : Tropfke, S. 37

Seite 10 : Juschkewitsch, S. 104; Ifrah, S. 487

Seite 11 : Tropfke, S. 45; Ifrah, S. 491; Ifrah, S. 486

Seite 25 : Tropfke, S. 52

Seite 13 (unten) : Tropfke, S. 66



Prof. Hans Rudolf Lerche

Zur Person und zum Vortrag

Prof. Lerche studierte bis zu seinem Diplom in Frankfurt, promovierte und habilitierte sich in Heidelberg. Dazwischen (1982) lag ein einjähriger Aufenthalt in Berkeley, der ihm bis heute noch gute Kontakte zu Wissenschaftlern in den USA beschert. Seit 1986 ist er Professor an der Universität Freiburg am Institut für Stochastik. Neben vielen Lehrverpflichtungen zur Statistikausbildung für verschiedene Fachrichtungen arbeitet er auf dem Gebiet der Sequentialstatistik mit Anwendungen in der Finanzmathematik und der medizinischen Statistik.

Aus diesem Spezialgebiet, das ursprünglich der Problematik der Qualitätskontrolle entstammt, stellt Prof. Dr. Hans Rudolf Lerche in seinem Vortrag eine Verallgemeinerung des Blackwellverfahrens vor. Durch die Reduktion des ursprünglichen Datensatzes auf eine Folge von 0/1-Sequenzen ist damit prinzipiell eine Zeitreihe aus jedem beliebigen Gebiet untersuchbar. Dies wird uns am Beispiel von Börsenindex und Währungskursen demonstriert werden.



Die Vorhersage und das Entdecken von Trendänderungen bei Finanzdaten

H. R. Lerche

R. Sandvoß

Institut für Mathematische Stochastik

Albert-Ludwigs-Universität Freiburg i. Br.

e-mail: lerche@galton.mathematik.uni-freiburg.de

sandvoss@galton.mathematik.uni-freiburg.de

1 Einleitung

Die Prognose von Finanzdaten ist eine wichtige Aufgabe der Ökonometrie und Finanzmathematik. Eine Vielzahl von Ansätzen und Vorschlägen gibt es für die verschiedenen Märkte. Neuronale Netze zum Beispiel erfreuen sich einer großen Popularität, beeindrucken sie doch durch Aufwand und Komplexität der Rechnung einerseits und schwierige Interpretierbarkeit andererseits.

In der Mathematik ist es üblich, bei der Durchführung einer Rechnung auch stets den Rechengang verständlich zu machen und aus der Rechnung auf die Struktur der untersuchten Objekte zurückzuschließen. Diesen Grundsätzen soll auch diese empirische Studie folgen.

Basierend auf dem Blackwell-Verfahren, einem stochastischen Algorithmus zur sukzessiven Vorhersage von unendlichen 0-1-Folgen, studieren wir die Vorhersage von Steigen und Fallen von Börsenkursen. Dabei gewinnen wir neben den Vorhersageresultaten auch Aussagen über das Gedächtnis der Zeitreihen und über den Informationsgehalt ihrer Trendänderungspunkte. Es sei betont, daß unsere Vorgehensweise rein explorativ ist und wir keinerlei Modellannahmen über die zugrundeliegenden Zeitreihen machen.

Untersucht werden sowohl Wechselkurse von US-Dollar/DM in stündlicher Notierung für den Zeitraum vom 02.05.1993 bis 29.08.1993 als auch die Schlußkurse des deutschen Aktienindex DAX vom 14.08.1992 bis 23.09.1993.

Die beiden Datensätze gehören zu ganz verschiedenen Märkten, die sich auch in ihrer Größe beträchtlich unterscheiden. Die Umsätze im US-Dollar/DM-Markt sind etwa 100 bis 200 mal so groß wie im deutschen Aktienmarkt. Um eine gewisse Vergleichbarkeit zu ermöglichen, haben wir deshalb beim ersten Datensatz stündliche Daten gewählt.

Als Arbeitshypothese schwebte uns während der Untersuchung die folgende Aussage vor: je größer der Markt, desto fairer die Preise. Mathematisch ausgedrückt: je größer der Markt, desto mehr tendiert der Preisprozeß gegen ein Martingal. Diese These sollte der Leser im Gedächtnis behalten und für sich entscheiden, inwieweit unsere Ergebnisse sie stützen. Eine Zusammenfassung der Resultate findet sich am Ende der Arbeit.

Wir danken Herrn Ulrich Müller, Olsen & Associates, Zürich, für die Bereitstellung des US-Dollar/DM Datensatzes.

2 Das Blackwellsche Vorhersageverfahren

Es sei x_1, x_2, \dots eine unendliche 0-1-Folge. Ein Vorhersageverfahren p_1, p_2, \dots ist eine zufällige unendliche 0-1-Folge, wobei p_{n+1} die Vorhersage für x_{n+1} ist. Der Wert von p_{n+1} kann von x_1, \dots, x_n und von weiteren von den Beobachtungen unabhängigen Zufallsgrößen (sog. Randomisierungen) abhängen.

Es sei $e_i = \mathbf{1}_{\{p_i=x_i\}}$ die Indikatorfunktion des Ereignisses, daß die i -te Beobachtung x_i richtig vorhergesagt wird. Desweiteren seien $\bar{x}_n = \frac{1}{n} \sum_{i=1}^n x_i$ und $\bar{e}_n = \frac{1}{n} \sum_{i=1}^n e_i$ die relativen Häufigkeiten der „1“ in der Folge x_1, x_2, \dots bzw. der richtigen Vorhersagen bis zum Zeitpunkt n .

Wir betrachten zunächst ein plausibles deterministisches Vorhersageverfahren. Sei

$$p_{n+1}^0 = \begin{cases} 1 & \text{für } \bar{x}_n > \frac{1}{2}, \\ 0 & \text{für } \bar{x}_n \leq \frac{1}{2}, \end{cases} \quad (1)$$

$n \geq 1$, und $p_1^0 \equiv 0$. Dieses Verfahren hat Stärken und Schwächen.

Ist x_1, x_2, \dots eine Folge von unabhängigen, identisch verteilten (u. i. v.) Bernoulli-Variablen, d. h. Zufallsgrößen mit $P(x_i = 1) = p = 1 - P(x_i = 0)$, so gilt wegen des Gesetzes der großen Zahlen für jedes p , $0 \leq p \leq 1$,

$$\bar{e}_n \longrightarrow \max(p, 1 - p) \quad \text{für } n \rightarrow \infty \quad \text{fast sicher.} \quad (2)$$

Für Bernoulli-Variablen leistet $(p_n^0, n \geq 1)$ das Bestmögliche. Denn wäre die Wahrscheinlichkeit p bekannt und beispielsweise $p > \frac{1}{2}$, dann sagt man am besten stets „1“ vorher und erreicht $\bar{e}_n \rightarrow p$. Ist $p \leq \frac{1}{2}$ und bekannt, so gilt entsprechend $\bar{e}_n \rightarrow 1 - p$, falls man stets „0“ vorhersagt.

Liegt jedoch die zyklische Folge $1, 0, 1, 0, 1, 0, \dots$ vor, so versagt das deterministische Verfahren $(p_n^0, n \geq 1)$ ganz, denn es gilt $\bar{e}_n = 0$ für $n \geq 1$.

Das Blackwellsche Verfahren besitzt keine dieser Schwächen. Es ist folgendermaßen definiert: Es sei $\mu_n = (\bar{x}_n, \bar{e}_n) \in [0, 1]^2$ und $S = \{(x, y) \in [0, 1]^2 \mid y \geq \max(x, 1 - x)\}$.

Außerdem seien D_1 , D_2 und D_3 das linke, das rechte und das untere Dreieck im Einheitsquadrat, d. h. $D_1 = \{(x, y) \in [0, 1]^2 \mid x \leq y \leq 1 - x\}$, usw. (siehe Bild 1).

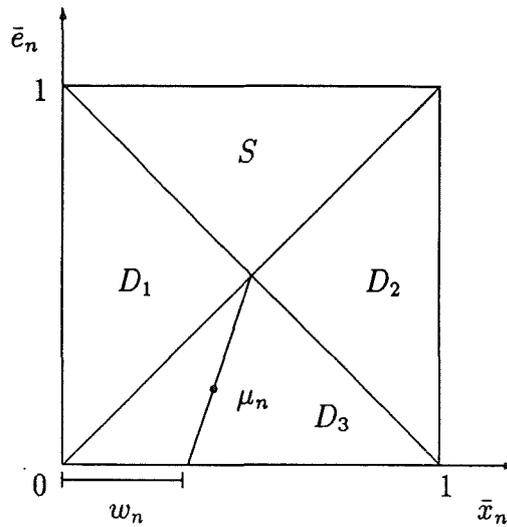


Bild 1

Liegt $\mu_n \in D_3$, so bezeichne w_n den Punkt, in dem die Verbindungsgerade der Punkte $(\frac{1}{2}, \frac{1}{2})$ und μ_n die x-Achse schneidet; $w_n = (\bar{x}_n - \bar{e}_n)/(1 - 2\bar{e}_n)$. Das Blackwell-Verfahren lautet dann für $n \geq 1$

$$\bar{p}_{n+1} = \begin{cases} 0 & \text{für } \mu_n \in D_1, \\ 1 & \text{für } \mu_n \in D_2, \\ Z_{n+1} & \text{für } \mu_n \in D_3 \end{cases} \quad (3)$$

und $\bar{p}_1 \equiv 0$, wobei Z_1, Z_2, \dots unabhängige Zufallsgrößen mit $P(Z_{n+1} = 1) = w_n = 1 - P(Z_{n+1} = 0)$ sind. Im Innern von S kann man \bar{p}_{n+1} beliebig wählen.

Man beachte, der Zufall wird nur dann in die Vorhersage miteinbezogen, wenn $\mu_n \in D_3$ gilt, d. h. wenn die bisherige Erfolgsquote niedrig ist.

Im folgenden bezeichne d den euklidischen Abstand in \mathbb{R}^2 und $d(x, A)$ den Abstand des Punktes x von der Menge A , $A \subset \mathbb{R}^2$.

Satz 1 Beim Blackwell-Verfahren $(\bar{p}_n, n \geq 1)$ konvergiert für jede unendliche 0-1-Folge x_1, x_2, \dots die Folge $(\mu_n, n \geq 1)$ gegen S , das heißt

$$d(\mu_n, S) \longrightarrow 0 \quad \text{für } n \rightarrow \infty \quad \text{fast sicher.} \quad (4)$$

Die Aussage des Satzes hat Minimax-Charakter. Die Bemerkungen nach (2) zeigen, daß für u. i. v. Bernoulli-Variablen Aussage (4) bestmöglich ist. Für jede andere 0-1-Folge ist das Blackwell-Verfahren ($\bar{p}_n, n \geq 1$) (asymptotisch) mindestens ebenso gut. Folglich sind u. i. v. Bernoulli-Variablen am schwersten vorherzusagen. Für eine ausführliche Diskussion des Verfahrens siehe [1] und [3].

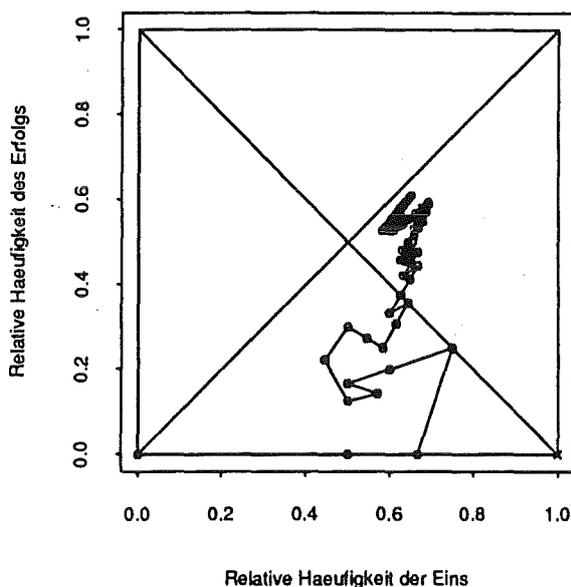


Bild 2: Vorhersage von $P(x_i = 1) = 0,6 = 1 - P(x_i = 0), i = 1, \dots, 100$

3 Modifikationen des Blackwell-Algorithmus

Für die Anwendung auf reale Daten erweist es sich als nützlich, diverse Varianten des Blackwell-Verfahrens zu betrachten. Für sie können dann zwar keine Konvergenzsätze mehr bewiesen werden, doch sind diese Modifikationen bei konkreten Anwendungen bisweilen effizienter.

Da häufig reale Daten ein gewisses endliches Gedächtnis haben, erscheint es sinnvoll, den Einfluß „früher“ Datenpunkte zu dämpfen oder ganz zu unterdrücken. Wir betrachten drei Modifikationen.

3.1 Das gleitende Mittel

Sei m eine natürliche Zahl mit $m \geq 1$. Wir definieren

$$\bar{x}_{n,m} = \begin{cases} \frac{1}{m} \sum_{i=0}^{m-1} x_{n-i} & \text{für } n \leq m, \\ \frac{1}{n} \sum_{i=0}^{n-1} x_{n-i} & \text{für } n < m \end{cases}$$

und

$$\bar{e}_{n,m} = \begin{cases} \frac{1}{m} \sum_{i=0}^{m-1} e_{n-i} & \text{für } n \leq m, \\ \frac{1}{n} \sum_{i=0}^{n-1} e_{n-i} & \text{für } n < m \end{cases}$$

und setzen $\mu_{n,m} = (\bar{x}_{n,m}, \bar{e}_{n,m})$. Außerdem wird in der Definition des Blackwell-Algorithmus (3) μ_n durch $\mu_{n,m}$ ersetzt. Die Punkte $\mu_{n,m}$ des so modifizierten Algorithmus laufen durch das Gitter mit Span $\frac{1}{m}$ des Einheitsquadrates ($n \geq m$). Sie können auch im oberen Dreieck S gelegen sein.

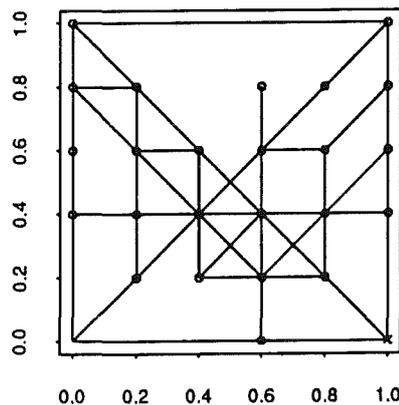


Bild 3: Pfad der Vorhersagepunkte $\mu_{n,m}$ mit $m = 5$ von 100 gleichverteilten 0-1-Daten.

3.2 Das gewichtete Mittel

Sei λ eine reelle Zahl mit $\lambda > 0$. Wir definieren

$$\bar{x}_{n,\lambda} = w^{-1} \sum_{i=1}^n e^{-\lambda(n-i)} x_i$$

und

$$\bar{e}_{n,\lambda} = w^{-1} \sum_{i=1}^n e^{-\lambda(n-i)} e_i,$$

wobei $w = \sum_{i=1}^n e^{-\lambda(n-i)}$ ist. Analog setzt man $\mu_{n,\lambda} = (\bar{x}_{n,\lambda}, \bar{e}_{n,\lambda})$ und ersetzt wiederum μ_n durch $\mu_{n,\lambda}$ in der Definition des Blackwell-Algorithmus (3). Die Punkte $\mu_{n,\lambda}$ des neuen Algorithmus berühren lediglich den Rand von S und laufen ziemlich irregulär durch das Einheitsquadrat.

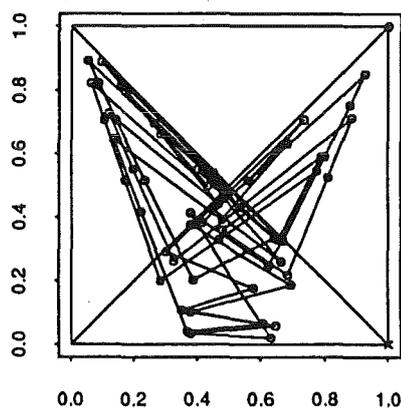


Bild 4: Pfad der Vorhersagepunkte $\mu_{n,\lambda}$ mit $\lambda = 1/2$ von 100 gleichverteilten 0-1-Daten

3.3 Die Streifenstopregel

Eine Modifikation ganz anderer Art ist die Streifenstopregel. Mit ihr wird versucht Serien von richtigen Vorhersagen zusammenzufassen, indem bei Trendwechseln das Blackwell-Verfahren erneut gestartet wird. Die Stopregel dazu ist

$$\tau = \min(\{n \geq 8 \mid \bar{e}_n < 0,5\} \cup \{n \geq 8 \mid \bar{e}_{n-1} \geq b, \bar{e}_n < b\}),$$

wobei die Schranke b den Gegebenheiten anzupassen ist. Die Vorschrift lautet, lasse das klassische Blackwell-Verfahren bis zum Zeitpunkt τ laufen und beginne bei $\tau + 1$ das Blackwell-Verfahren von neuem mit den Punkten $x'_1 = x_{\tau+1}$, $x'_2 = x_{\tau+2}$, ...

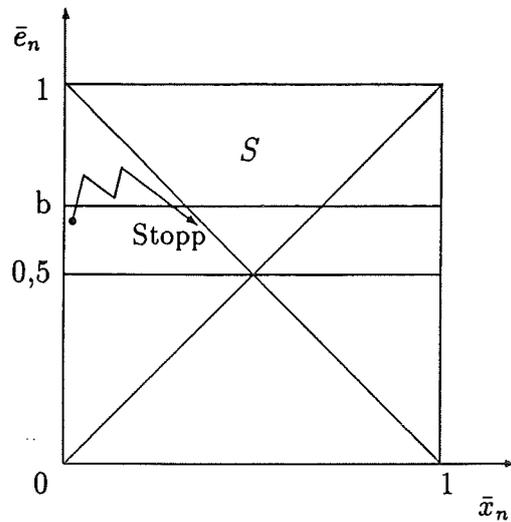


Bild 5: Streifenstoppregel

4 Einige Vorhersageergebnisse

Im folgenden wollen wir die Vorhersagequalitäten an Hand von Börsendaten untersuchen. Wir betrachten dazu zwei Datensätze ganz unterschiedlicher Märkte: zum einen stündliche Kurse von US-Dollar/DM, zum anderen tägliche Kurse des deutschen Aktienindex DAX.

Titel	N+1	Zeit Δt	Art	Zeitraum	Quelle
Dollar	2041	stündlich	Mittelkurs	02.05.93 - 29.08.93	Olsen & Associates
Dollar	2041	1h ϑ -Zeit	Mittelkurs	02.05.93 - 29.08.93	Olsen & Associates
Dax	280	täglich	Schlußkurs	14.08.92 - 23.09.93	FAZ

Tabelle 1: Datensätze (z_i) , $i = 1, \dots, N + 1$, mit Zeitabstand und Quelle

Es sei bemerkt, daß der zweite Dollar-Datensatz durch eine Zeittransformation entsteht. Die Zeitskala ist die sogenannte ϑ -Zeit. Sie ist eine auf (nahezu) gleiche Handlungsdichten gestreckte Skala und wurde von Dacorogna et al. in [2] vorgeschlagen.

Wir studieren nun die Vorhersage von Steigen und Fallen der Kurse, jeweils von einer Notierung zur nächsten. Dazu wird jede Zeitreihe (z_i) , $i = 1, \dots, N + 1$, in eine 0-1-Folge (x_i) , $i = 1, \dots, N$, umgewandelt, wobei Null für „Fallen“ und Eins für „Steigen“ steht:

$$x_i = \begin{cases} 0 & \text{für } z_{i+1} - z_i \leq 0, \\ 1 & \text{für } z_{i+1} - z_i > 0. \end{cases} \quad (5)$$

Neben dem Blackwell-Verfahren $(\bar{p}_n, n \geq 1)$ (vgl. (3)) und den bereits besprochenen Modifikationen verwenden wir auch das nichtrandomisierte Vorhersageverfahren $(p_n^0, n \geq 1)$ (vgl. (1)).

Schließlich betrachten wir die hypothetische Situation, daß die Trendänderungspunkte (change-points) bekannt sind. An diesen Punkten stoppen wir das Blackwell-Verfahren und starten es dort erneut. In jedem Datensatz wurden fünf Punkte als change-points ausgewählt. Die Vorhersageergebnisse dieser Methode gestatten uns Rückschlüsse auf den Nutzen der Analyse von change-points.

Wir beginnen mit der Diskussion der Ergebnisse der Dollardatensätze. Anhand der folgenden Abbildung sieht man, daß sich die Kursverläufe der beiden Datensätze kaum unterscheiden.

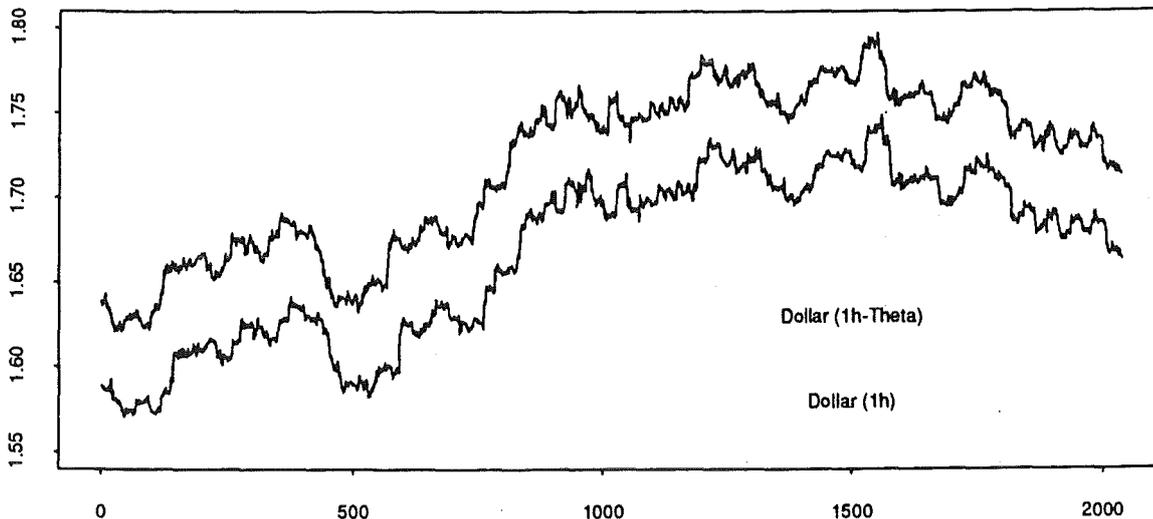


Bild 6: Kursverläufe der Datensätze Dollar (1h) und (1h ϑ) (verschoben um 0,05)

Ebenso sind die Vorhersageergebnisse nahezu gleich, die Quote der richtigen Vorhersagen \bar{e}_N der Kurse in ϑ -Zeit ist etwas höher. Interessant ist auch, daß das klassische Blackwell-Verfahren unter allen betrachteten Verfahren am besten abschneidet und daß die Prognose durch die Vorgabe der change-points nur geringfügig verbessert wird.

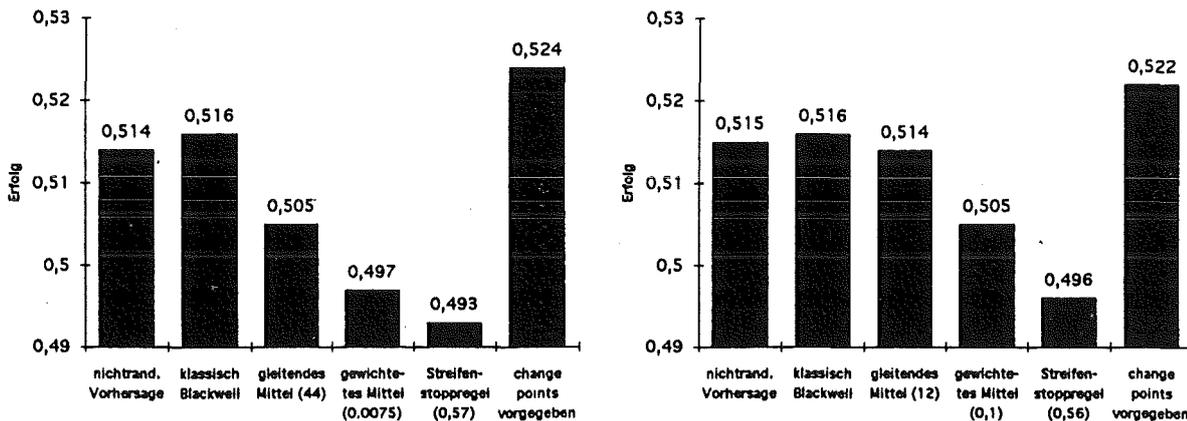


Bild 7: Vorhersageergebnisse der Datensätze Dollar (1h) und (1h ϑ)

Bei der DAX-Vorhersage sehen die erzielten Ergebnisse anders aus. Hier sind die mit gleitendem und gewichtetem Mittel modifizierten Verfahren besser als das klassische Blackwell-Verfahren. Die optimalen Parameter sind $m = 5$ und $\lambda = 0,19$.

Diese Werte verdienen besondere Beachtung. Sie zeigen, daß für die untersuchte Periode der DAX ein Gedächtnis von etwa 5 Handelstagen besitzt. Dabei ist zu beachten, daß dem Wert $\lambda = 0,19$ ein Erwartungswert von $1/\lambda = 5,26$ entspricht. (Die optimalen Parameterwerte m , λ und $1/\lambda$ sind vom Datensatz abhängige Schätzwerte.)

Bei dem DAX-Datensatz zeigt sich ebenfalls, daß die Kenntnis der change-points von Nutzen ist. Folglich erscheint es sinnvoll, nach diesen change-points mit geeigneten statistischen Mitteln zu suchen. Wir kommen auf diesen Sachverhalt im nächsten Abschnitt zurück.

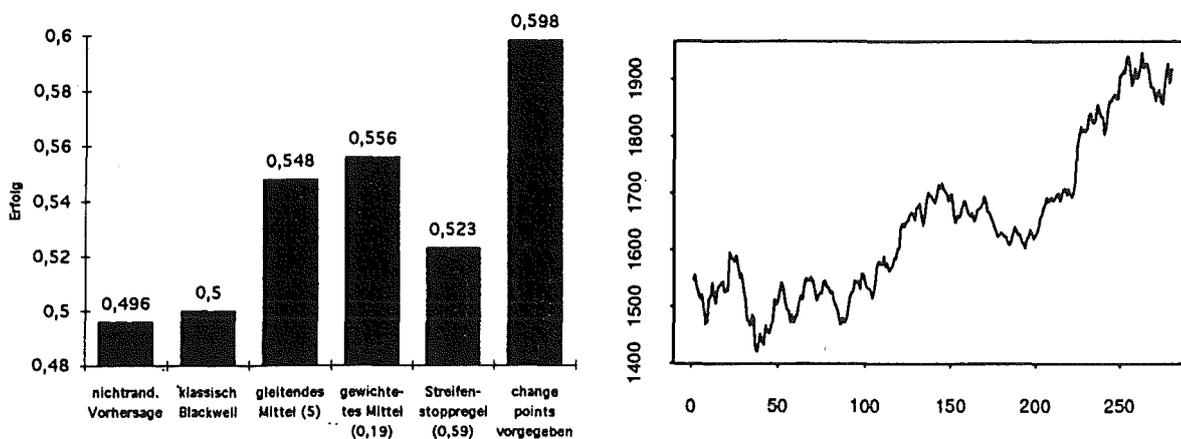


Bild 8: Vorhersageergebnis und Kursverlauf des Datensatzes DAX

Wir wollen uns nun mit der Vorhersage der 3-Glättungen der Originalzeitreihen beschäftigen. Damit zielen wir einerseits in Richtung Trendvorhersage, andererseits können wir so die Verfahren auf ihre Anpassungsfähigkeit hin überprüfen.

Die Trendzeitreihe (\tilde{z}_i) , $i = 1, \dots, N + 1$, ist gegeben durch

$$\tilde{z}_1 = z_1, \quad \tilde{z}_2 = (z_1 + z_2)/2, \quad \tilde{z}_i = \frac{1}{3} \sum_{l=i-2}^i z_l, \quad i = 3, \dots, N+1.$$

Entsprechend zur Definition (5) erklären wir die zugehörige 0-1-Folge (\tilde{x}_i) , $i = 1, \dots, N$, die Fallen oder Steigen der \tilde{z}_i -Zeitreihe angibt. Trotz der Glättung der Notierungen bleibt die Struktur des Kursverlaufes erhalten, nur das Zittern der Kurse ist geringer geworden.

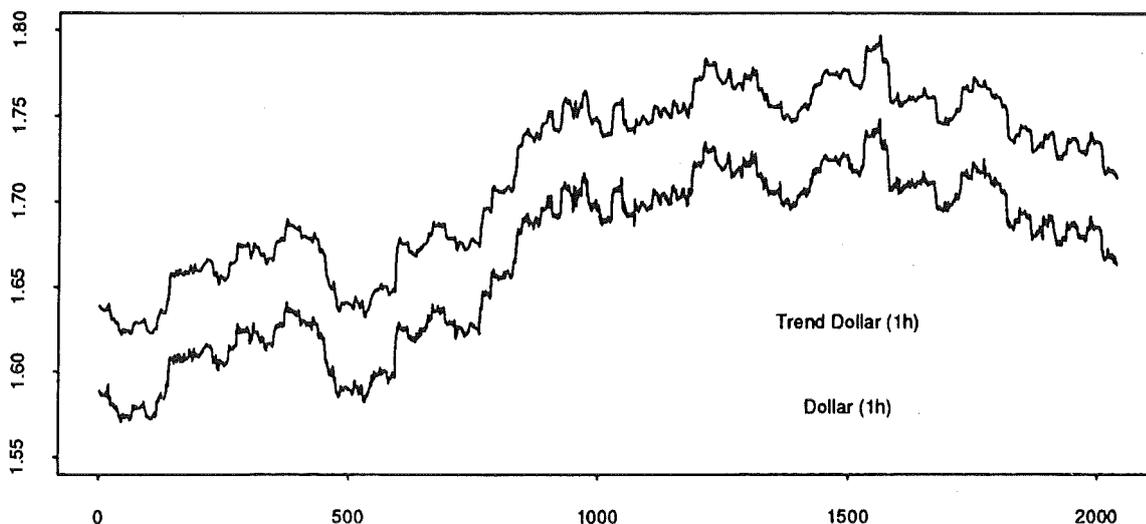


Bild 9: Original- und Trendzeitreihe des Dollar (1h) (verschoben um 0,05)

Für die 3-Glättungen der Dollardatensätze beobachtet man eine beträchtliche Steigerung der Erfolgsraten für das gleitende und das gewichtete Mittel. Sogar die Resultate für vorgegebene change-points werden übertroffen. Die beste Streifenstoppregel schneidet deutlich schlechter ab, das Niveau des Blackwell-Verfahrens bleibt auch niedrig. Die Änderungen der Ergebnisse aufgrund der ϑ -Zeitachsentransformation sind unwesentlich.

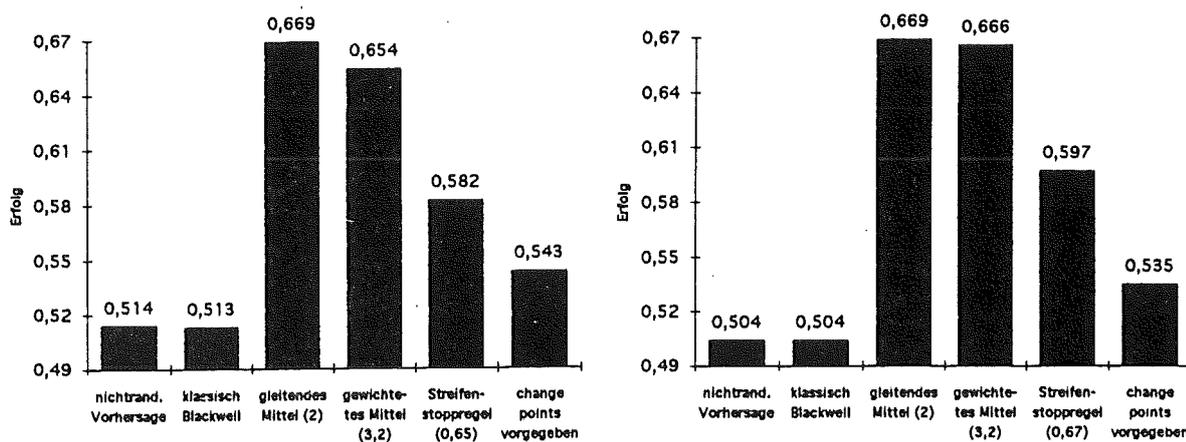


Bild 10: Vorhersageergebnisse der Trendzeitreihen des Dollar (1h) und (1h ϑ)

Bei der DAX-Vorhersage liegen die Verhältnisse ähnlich. Gleitendes und gewichtetes Mittel liefern die besten Resultate. Bei beiden Datensätzen passen sich die optimalen Parameter den veränderten Situationen an. Durch das Glätten der Zeitreihe (z_t) wird der Einfluß früherer Datenpunkte geringer. Dies bedeutet, daß m kleiner und λ größer werden.

Die guten Erfolgsraten sind insofern nicht erstaunlich, da in die Werte \tilde{z}_t die bereits bekannten Werte z_{t-1} und z_{t-2} eingehen.

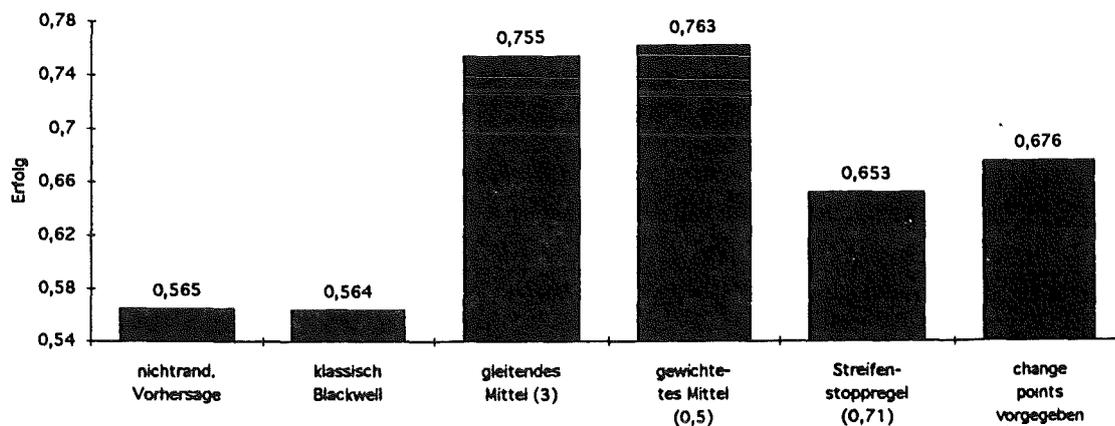


Bild 11: Vorhersageergebnisse der Trendzeitreihe des DAX

5 Eine change-point Analyse

Wir wollen nun Verfeinerungen der in Abschnitt zwei und drei eingeführten Verfahren diskutieren. Diese beruhen auf der Analyse der Trendänderungspunkte (change-points). Wie Bild 8 in Abschnitt 4 zeigt, erhöht sich die Erfolgsrate der Prognose bei fiktiver Kenntnis der change-points um ca. 8 Prozent bei den Dax-Daten.

Das folgende Bild zeigt den DAX-Chart mit den (fünf) als bekannt angenommenen change-points.

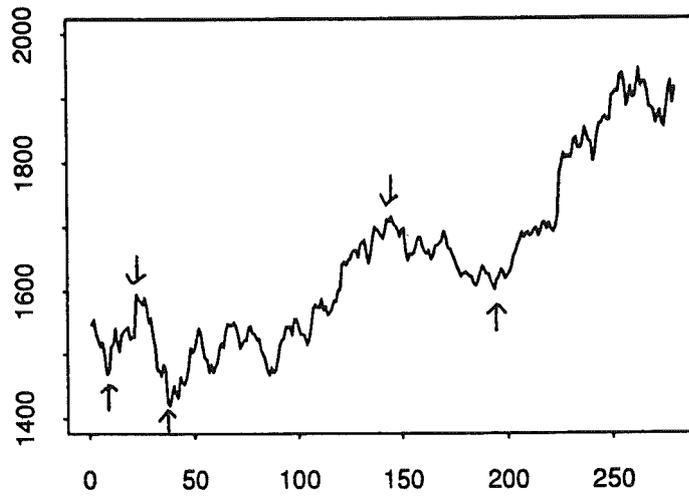


Bild 12: Kursverlauf des Datensatzes DAX mit change-points

Als Statistik zur Detektion der change-points verwenden wir die Differenz zweier gleitender Mittel:

$$I(j) := \frac{1}{3} \sum_{i=1}^3 x_{j+i-1} - \frac{1}{7} \sum_{i=1}^7 x_{j+i-1}.$$

Für den untersuchten Zeitraum besitzt der Indikator $I(j)$ die untenstehende Gestalt:

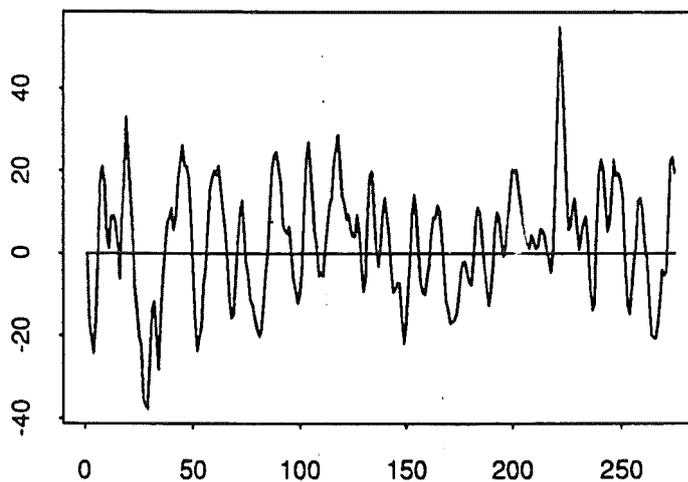


Bild 13: Indikator $I(j)$ für den Datensatz DAX

Die Verfeinerungen sehen folgendermaßen aus: Das Blackwell-Verfahren oder eine seiner Modifikationen läuft solange, bis ein Minimum oder Maximum der Funktion $I(j)$ erreicht wird. Am zweiten Punkt nach der Extremstelle wird es gestoppt und am nächsten Punkt erneut gestartet, u.s.w. Wir nennen diese Stoppanweisung die 3-7-Stoppregel. Im Vergleich des neuen Verfahrens mit den bereits bekannten erhält man das nachfolgende Bild.

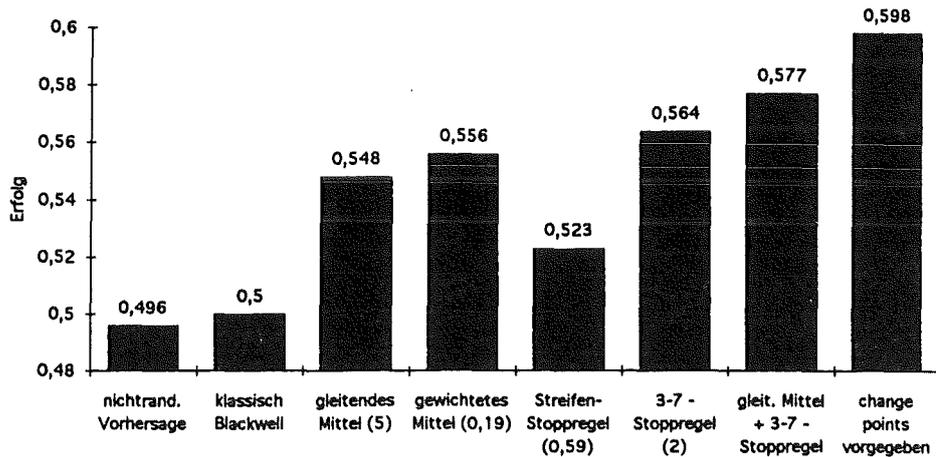


Bild 14: Vergleich der verschiedenen Vorhersageergebnisse des Datensatzes DAX

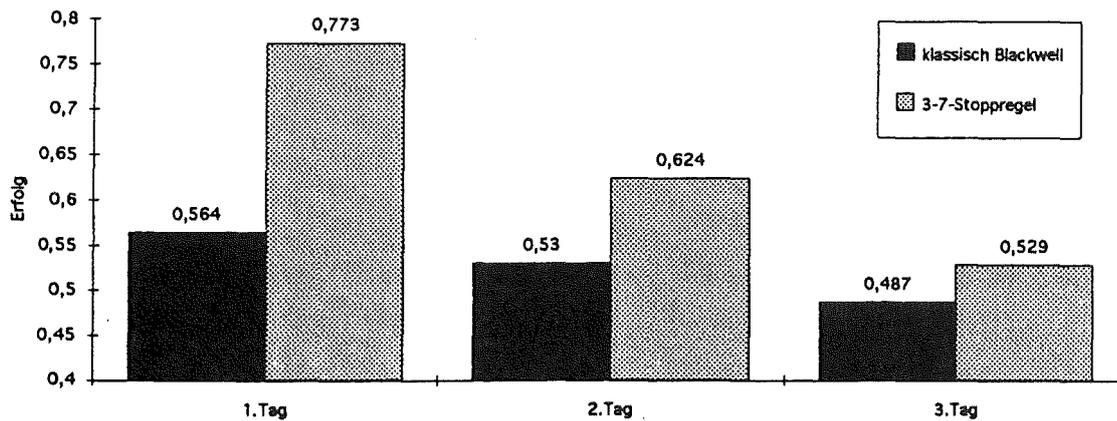


Bild 15: Vorhersageergebnisse (klassisches Blackwellverfahren und Modifikation mit 3-7-Stoppregel) der Trendzeitreihe DAX für mehrere Tage

Auch bei der Trendvorhersage sind die Verbesserungen nicht unbedeutend. Bild 15 zeigt die Prognoseergebnisse des Trends für die nächsten drei Notierungen.

6 Eine Komplexitätsanalyse

Die untersuchten Zeitreihen unterscheiden sich beträchtlich in ihrer Vorhersagbarkeit durch die verschiedenen Verfahren. Bei den Dollar-Daten ist das Blackwell-Verfahren am besten; bei den DAX-Daten das gewichtete Mittel, das gleitende Mittel ist jedoch nur unmerklich schlechter. Wir wollen nun versuchen, die Zeitreihen bezüglich ihrer Vorhersagbarkeit zu klassifizieren.

Zur Vereinfachung vergleichen wir das Blackwell-Verfahren lediglich mit seinen Gleitenden Mittel-Modifikationen. Zwei Maßzahlen sind dazu notwendig:

Definition 1 Die maximale relative Häufigkeit r einer 0-1-Folge (x_i) , $i = 1, \dots, N$, der Länge N ist gegeben durch

$$r := \max \left\{ \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{x_i=0\}}, \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{x_i=1\}} \right\}.$$

Definition 2 Die Sortierung s einer 0-1-Folge (x_i) , $i = 1, \dots, N$, der Länge N ist gegeben durch

$$s := \frac{1}{N-1} \#\{x_i \mid x_i = x_{i+1}, i = 1, \dots, N-1\}.$$

Als Komplexitätsmaß für 0-1-Folgen der Länge N wählen wir die Größe $r - s$.

Unsere intuitive Vorstellung bei dem Vergleich ist die folgende: Ist das Komplexitätsmaß klein, so soll bei den Vorhersagen eine Steigerung des Erfolges durch ein geeignetes gleitendes Mittel gegenüber dem Blackwell-Verfahren möglich sein. Ist $r - s$ groß, so soll das Blackwell-Verfahren dominieren. Die Datensätze ordnen wir in der folgenden Tabelle nach der Größe von $r - s$.

$r - s$	r	s	Titel	Blackwell	Gl. Mittel	Steigerung
-0,213	0,550	0,763	Trend Dax	0,564	0,755	0,191
-0,202	0,507	0,709	Trend Dollar (1h ϑ)	0,504	0,669	0,165
-0,187	0,511	0,698	Trend Dollar (1h)	0,513	0,669	0,156
-0,012	0,535	0,547	Dax	0,500	0,548	0,048
0,041	0,518	0,477	Dollar (1h ϑ)	0,516	0,514	-0,002
0,059	0,518	0,459	Dollar (1h)	0,516	0,505	-0,011

Tabelle 2: Zeitreihen aus Tabelle 1 mit Komplexität $r - s$ und Vergleich des Erfolgs zwischen Blackwell-Verfahren und gleitendem Mittel.

Die Ergebnisse zeigen, daß mit wachsender Komplexität $r - s$ die Verbesserungen durch das beste gleitende Mittel gegenüber dem Blackwell-Verfahren abfallen.

Schließlich bestimmen wir die Größe $r - s$ für eine Folge von unabhängigen Bernoulli-Variablen und vergleichen diese mit den 0-1-Folgen, die zu den Dollar-Datensätzen (1h, 1h ϑ) gehören. Wir simulieren dazu Daten x_i mit $P(x_i = 1) = 0,518 = 1 - P(x_i = 0)$, $i = 1, \dots, 2040$, und berechnen $r - s$. Die Ergebnisse von drei so erzeugten zufälligen Datensätzen zeigt die folgende Tabelle. Wir sehen, die Ergebnisse könnten auch von den Dollar-Datensätzen stammen.

Datensatz	$r - s$	r	s
1	0,013	0,512	0,499
2	0,025	0,520	0,495
3	0,047	0,540	0,493

Tabelle 3: Komplexität $r - s$ von x_i , $P(\{x_i = 1\}) = 0,518 = 1 - P(\{x_i = 0\})$, $i = 1, \dots, 2040$.

7 Conclusio

Beim Vergleich von stündlichen US-Dollar-Daten mit täglichen DAX-Daten anhand des Blackwell-Verfahrens und seiner Modifikationen ergeben sich beträchtliche Unterschiede in der Struktur der Datensätze.

Die DAX-Daten lassen sich leichter vorhersagen, ihre Trendänderungspunkte enthalten für die Vorhersage relevante Information. Beim Dollar haben diese Punkte keine Bedeutung.

Das Gedächtnis der DAX-Daten liegt bei etwa fünf Tagen. Die Struktur des US-Dollar ist erheblich komplexer, das einfache Blackwell-Verfahren erweist sich wie bei den u.i.v. Bernoulli-Variablen als bestmöglich. Der Dollarpreis als stochastischer Prozeß, scheint näher an einem Martingal zu liegen als der DAX.

Literatur

- [1] Blackwell, D. (1956). An analog of the minimax theorem for vector payoffs. *Pac. Jour. Math.* **6**, 1-8.
- [2] Dacorogna, M., Müller, U., Nagler, R., Olsen, R., Pictet, O. (1992). *A geographical model for the daily and weekly seasonal volatility in the FX market*. Olsen & Associates, Research Institute for Applied Economics, Zürich.
- [3] Lerche, H. R., Sarkar, J. (1994). The Blackwell Prediction Algorithm for Infinite 0-1 Sequences, and a Generalization. In: *Statistical Decision Theory and Related Topics V* (S. S. Gupta, J. O. Berger, eds.), 503-511. Springer, New York.



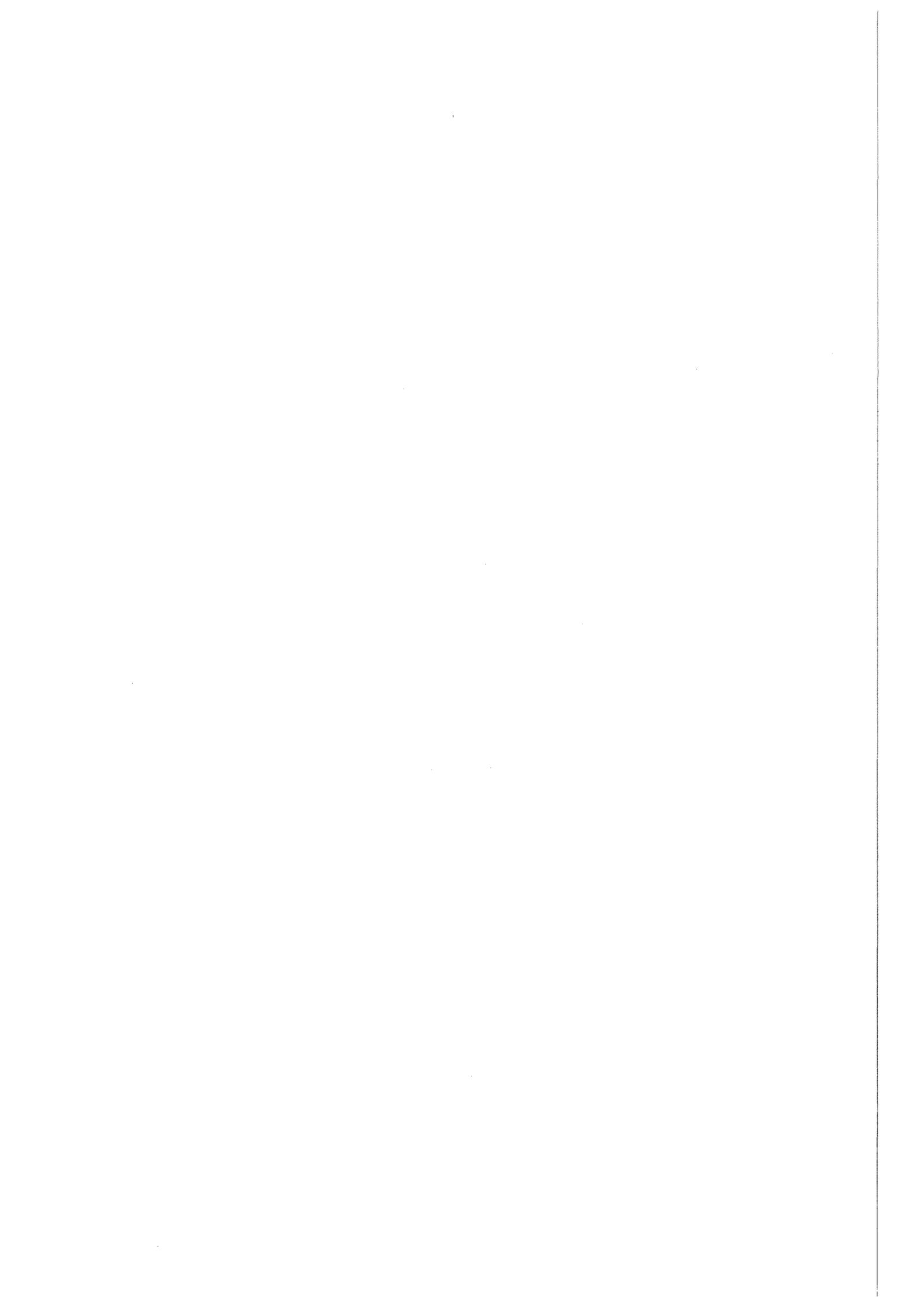
Dr. Alexander D. Tsodikov

Zur Person und zum Vortrag

Dr. Alexander D. Tsodikov ist Wissenschaftler in der Abteilung für Angewandte Mathematik an der Technischen Hochschule St. Petersburg. Seine Forschungsgebiete spannen einen weiten Bogen von der Biomathematik zur Biostatistik, insbesondere Modellierung von Krebserkrankungen, "Überlebens"-Analysis sowie optimale Regelung in Krebsüberwachung und Krebsbehandlung. Ebenso trug er wesentlich zur Entwicklung von Software für biomedizinische Studien bei.

Dr. Tsodikov erhielt sein Diplom in Angewandter Mathematik 1988 und promovierte 1991 an der Technischen Hochschule St. Petersburg. In den Jahren 1988 und 1989 arbeitete er im Zentrum für Ingenieurwesen derselben Universität. Im Jahre 1989 trat er auf Einladung des Leiters der Abteilung, Prof. A. Yakovlev, wieder der Abteilung Angewandte Mathematik bei. Seine neuesten wissenschaftlichen Arbeiten sind eng mit der Abteilung Biostatistik des Curie-Instituts in Paris verknüpft, wo er 18 Monate als Wissenschaftler arbeitete.

In seinen Vortrag "Nichtparametrische Schätzung einer Überlebens-Funktion aus unvollständigen Daten" fließen zahlreiche Erkenntnisse dieser Arbeiten ein.



NONPARAMETRIC ESTIMATION OF A SURVIVOR FUNCTION FROM INCOMPLETE DATA

Alexander D. Tsodikov

St. Petersburg State Technical University, Russia

Abstract

This paper is concerned with the estimation of a distribution function G following a discrete failure time model from incomplete samples. A maximum likelihood estimate of G with mild assumptions on the form of the likelihood is suggested. The method provides in particular the maximum likelihood estimates of the times at which the empirical distribution function has its steps as well as the estimate of the number of steps thus answering the question of how detailed an estimate should be to most closely correspond to the actual amount of information in the sample. It offers an isotonic solution when the monotony property is not inherent to the empirical distribution function. Examples for right censored, doubly censored and discrete surveillance data are given.

1 Introduction

This paper deals with nonparametric estimation of the distribution (or survivor) function of a nonnegative random variable which is not directly observable. Incomplete data samples usually arise in the reliability analysis and biological studies when a failure or disease onset is unobservable either due to its nature or to a specific study design. It is not uncommon that a sample contains no accurate times to failure, or there are quite few of them. In such cases the amount of information contained in a sample turns to be quite poor even if the sample itself is large. The estimate also termed the empirical distribution function is thought of as a step function, the step points being sometimes referred to as the grouping points. The problem of nonparametric estimation of the distribution function from incomplete samples have been studied by many authors. For particular study designs as for example the follow-up with right noninformative censoring the problem is well-studied (Cox and Oakes(1983)). Further attempts to construct a general algorithm were based on the "self-consistency" condition first formulated by Efron (1967) with respect to the product limit estimate. The concept of "self-consistency" was extended by Turnbull (1976) who developed an iterative algorithm converging to the maximum likelihood estimate of the

distribution function with arbitrary grouped, censored and truncated data. Elegant and computationally effective, however, the algorithm by Turnbull does not cover for example the samples arising from discrete surveillance where the time to failure is known with some probability to be at some point from a certain set (Tsodikov et al. (1994)).

In the present paper we develop an alternative approach based on the exhaustive search algorithm, which has the following features:

- It is rather general and covers any real-time study, where each observed event is supposed to be a consequence of the past and censoring is non-informative;
- It is not iterative and gives the global solution with finite number of steps;
- It offers an isotonic solution when the monotony property is not inherent to the empirical distribution function;
- It offers an optimal data adaptive grouping which is not taken for given but is a part of the arguments to be optimized by the maximum likelihood method;
- It allows to find out how detailed an estimate should be to most closely correspond to the actual amount of information in the sample;
- It provides a natural way of data smoothing.

A disadvantage of the approach is that it takes much computer time to get an estimate, which is a payment for generality.

2 General procedure

2.1 The problem

Let the random variable (r.v.) U be the time to failure with the distribution function $G(t)$ and the survivor function $\bar{G}(t) = 1 - G(t)$. The empirical distribution function $\hat{G}(t)$ will be thought of as a step function with the step points in the set $\mathcal{T} = \{t_i\}_{i=1}^m$ sorted in nondecreasing order. Practically, the set \mathcal{T} can be formed by including the times of all the events observed in the study. In this case m is equal to the sample size. For some designs it is known where the steps might be, and the set \mathcal{T} can be further reduced.

The actual step-points are denoted by $s_n = \{\tau_i\}_{i=1}^n$, $n \leq m$. They are supposed to satisfy the strict inequalities

$$0 < \tau_1 < \tau_2 < \dots < \tau_n \leq \tau_{n+1} \stackrel{\text{def}}{=} t_m. \quad (1)$$

Let \mathcal{D}_n stand for the set of all possible s_n satisfying (1). Similarly the values of the empirical distribution function at the step-points $G(\tau_i) \stackrel{\text{def}}{=} G_i$ satisfy the monotony property

$$0 < G_1 < G_2 < \dots < G_n \leq 1, \quad (2)$$

$e_n = \{G_i\}_{i=1}^n$, $n \leq m$ and \mathcal{E}_n is the set of all possible e_n satisfying (2). Given that $n \in \mathbf{N}$, where \mathbf{N} is the set of natural numbers bounded by m , the class $\mathcal{F} = \cup_{n=1}^m \{\mathcal{D}_n, \mathcal{E}_n\}$ will be the mutually exclusive class of all admissible step functions. Then the problem of estimation of $G(t)$ may be formulated as

$$\max_{G \in \mathcal{F}} \ell_T, \quad (3)$$

where T is the duration of the study, $T \geq t_m$, and ℓ_T is the loglikelihood function.

2.2 The algorithm

Let φ_i be the contribution to the loglikelihood of the events related to the point τ_i , i.e. the events entering the interval $(\tau_i, \tau_{i+1}]$. It is natural to assume that this contribution is dependent on the part of the distribution function G on the interval $[0, \tau_i]$. The meaning of this assumption is that any observed event is a reflection of failures which might have occurred prior to the event and so $\varphi_i = \varphi_i(s_i; e_i)$, $i = 1, \dots, n$. In other words, a failure which has not yet occurred could not influence the observed process. As a result the loglikelihood is supposed to be of the form

$$\ell = \sum_{i=0}^n \varphi_i(s_i; e_i). \quad (4)$$

Imagine that we have observed the population subject to failures for some period $x < T$. If we knew part of the solution \hat{G} on $[0, x]$ then we could get the whole estimate \hat{G} by maximizing only a part of the functional ℓ_T with respect to the part of \hat{G} on $[x, T]$

$$\max_{\tau_j, \tau_{j+1}, \dots; G_j, G_{j+1}, \dots} \sum_{i=j, j+1, \dots} \varphi_i(s_i; e_i),$$

where $j = \min\{k : \tau_k > x\}$. By introduction of a grid for the values of τ and G

and looking over all combinations of (s_i, e_i) on the grid it is possible to find the solution of (3). At the same time the structure (4) of the functional ℓ_T allows to compute only a part of the functional for each combination (s_i, e_i) on the grid. We will term this procedure an exhaustive search algorithm. If φ_i were dependent only on the first point in the past

$$\varphi_i = \varphi_i(\tau_{i-1}, \tau_i; G_{i-1}, G_i)$$

the exhaustive search procedure could be reduced to the dynamic programming one. By the dynamic programming procedure it is also possible to find an approximate solution of (3) when τ_{i-1}, G_{i-1} constitute the most part of the dependence of φ_i on the past. This happens in the majority of cases when a failure influences the observed process only in the vicinity of the failure time.

Let $\hat{s}_k, \hat{e}_k, k < n$ be the optimal solution to the problem

$$\max_{s_k, e_k, \text{ given } \tau_{k+1}, G_{k+1}} \ell_{\tau_{k+1}}(s_k; e_k), \quad (5)$$

where $\ell_{\tau_{k+1}} = \sum_{i=0}^k \varphi_i(s_i; e_i)$. Consider the optimal value $\hat{\ell}_{\tau_{k+1}}$ and the vectors \hat{s}_k and \hat{e}_k as the functions of τ_{k+1} and G_{k+1} . According to the dynamic programming procedure for \hat{s}_{k+1} and \hat{e}_{k+1} we have

$$\begin{aligned} \hat{\ell}_{\tau_{k+2}}(s_{k+1}; e_{k+1}) = \\ \max_{\tau_{k+1}, G_{k+1} \text{ given } \tau_{k+2}, G_{k+2}} \{ \hat{\ell}_{\tau_{k+1}}(\tau_{k+1}; G_{k+1}) + \varphi_{k+1}(\tau_{k+1}; G_{k+1}) \}. \end{aligned} \quad (6)$$

In the expression (6) \hat{s}_k and \hat{e}_k are omitted in the arguments of the functions $\hat{\ell}$ and φ since they are the functions of τ_{k+1} and G_{k+1} given that they are solutions of (5). Solving (6) with respect to τ_{k+1}, G_{k+1} , we obtain $\hat{\ell}_{\tau_{k+2}}$ and the vectors \hat{s}_{k+1} and \hat{e}_{k+1} as the functions of $\tau_{k+2}, G_{k+2} \dots$ etc. Starting from $k = 1$ we proceed with the above step until $k + 2 = m$ thus looking over $n = 1, 2, \dots, m$. Indeed, for any n the optimal solution \hat{s}_n, \hat{e}_n may be extracted by solving

$$\max_{\tau_n, G_n} \{ \hat{\ell}_{\tau_n}(\tau_n, G_n) + \varphi_n(\tau_n; G_n) \},$$

where in φ_n the contribution of observations right censored by T is included. It only remains for us to chose which of the values $\hat{\ell}_T(\hat{s}_n, \hat{e}_n)$ is the largest. As a result of the above forward procedure we get solutions for $n < m$ submerged into the solution for $n = m$.

2.3 Continuity issue

Denote by $\bar{\mathcal{D}}_n$ and by $\bar{\mathcal{E}}_n$ the classes analogous to \mathcal{D}_n and by \mathcal{E}_n , but subject to non-strict inequalities

$$0 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_n \leq \tau_{n+1} = t_m$$

and

$$0 \leq G_1 \leq G_2 \leq \dots \leq G_n \leq G_n \leq 1,$$

respectively. It is then evident that

$$\{\bar{\mathcal{D}}_1, \bar{\mathcal{E}}_1\} \subset \{\bar{\mathcal{D}}_2, \bar{\mathcal{E}}_2\} \subset \dots \subset \{\bar{\mathcal{D}}_m, \bar{\mathcal{E}}_m\}$$

The class $\{\mathcal{D}_k, \mathcal{E}_k\}$ may be regarded as the border of the class $\{\bar{\mathcal{D}}_{k+1}, \bar{\mathcal{E}}_{k+1}\}$, $k = 1, \dots, m-1$, and we will denote this by

$$\{\mathcal{D}_k, \mathcal{E}_k\} = \Gamma(\{\bar{\mathcal{D}}_{k+1}, \bar{\mathcal{E}}_{k+1}\}). \quad (7)$$

Suppose that $s_n = (\tau_1, \dots, \tau_n)$, $e_n = (G_1, \dots, G_n)$ and $(s_n, e_n) \in \{\mathcal{D}_n, \mathcal{E}_n\}$. We may tend (s_n, e_n) to the border $\Gamma(\{\bar{\mathcal{D}}_n, \bar{\mathcal{E}}_n\})$ by either of the two ways

$$(s_n, e_n) \longrightarrow (s'_{n-1}, e'_{n-1}) \text{ as } \tau_k \rightarrow \tau_{k+1},$$

$$s'_{n-1} = (\tau_1, \tau_2, \dots, \tau_{k-1}, \tau_{k+1}, \tau_{k+2}, \dots, \tau_n)$$

$$e'_{n-1} = (G_1, G_2, \dots, G_{k-1}, G_k + G_{k+1}, G_{k+2}, \dots, G_n)$$

or

$$(s_n, e_n) \longrightarrow (s''_{n-1}, e''_{n-1}) \text{ as } G_k \rightarrow G_{k+1},$$

$$s''_{n-1} = (\tau_1, \tau_2, \dots, \tau_{k-1}, \tau_k, \tau_{k+2}, \tau_{k+3}, \dots, \tau_n)$$

$$e''_{n-1} = (G_1, G_2, \dots, G_{k-1}, G_{k+1}, G_{k+2}, G_{k+3}, \dots, G_n)$$

for some $k < n$.

If the maximum likelihood principle is applied correctly, the following equations hold.

$$\lim_{(s_n, e_n) \rightarrow (s'_{n-1}, e'_{n-1})} \ell(s_n, e_n) = \ell(s'_{n-1}, e'_{n-1}) \quad (8^a)$$

$$\lim_{(s_n, e_n) \rightarrow (s''_{n-1}, e''_{n-1})} \ell(s_n, e_n) = \ell(s''_{n-1}, e''_{n-1}). \quad (8^b)$$

By the correct application we mean that the observed sample may be reproduced with a non zero probability by a step function G from a certain class. And the

purpose is to extract the function from this class for which the probability of the observed sample is maximal. If the equations (8) do not hold, the likelihood depends on to which class the pair (s, e) is attributed. This means that we are using not the exact likelihood but the approximate one, thus modifying the original observed sample. Consider the life table estimate to make the point clear. For this estimate the contribution to the multinomial loglikelihood associated with point τ_i has the form $m_i \log(G_i - G_{i-1}) + n_i \log(\bar{G}_i)$, where m_i and n_i are the number of uncensored and censored observations entering the interval $(\tau_{i-1}, \tau_i]$, respectively. Suppose that there is no step at the point τ_i while $m_i > 0$ and G is regarded as having a zero step at the point τ_i . Then we will have the loglikelihood equal to $-\infty$. If however we do not allow zero step-points, the likelihood will never be $-\infty$. This fact means that any discrete failure time model which does not presume a non zero step point at each uncensored member of the sample will be inconsistent with the data failing to explain how the failure did happen at a point with zero probability.

This does not necessarily mean that we should not apply the algorithms when (8) do not hold. The decision depends on whether the final solution presents a good approximation or not.

Let $\hat{\ell}_T^{(n)}$ stand for the optimal likelihood in the class $\{\bar{D}_n, \bar{E}_n\}$ and the equations (8) hold. Then

$$\hat{\ell}_T^{(1)} \leq \hat{\ell}_T^{(2)} \leq \dots \leq \hat{\ell}_T^{(m)}. \quad (9)$$

So, the nested sequence of sets provides the ordered sequence of maximum likelihoods.

The set \mathcal{F} in this case is equivalent to the set $\bar{\mathcal{F}} = \{\bar{D}_m, \bar{E}_m\}$ which regards any step function as a function having exactly m steps allowing that some of the step-probabilities be zero. This is the case for example for doubly censored data (Subsection 3.2) and is not for the follow-up with right censoring.

If (8) hold the problem (3) is equivalent to the problem

$$\max_{G \in \bar{\mathcal{F}}} \ell_T. \quad (10)$$

Talking in terms of problem (3) we mean that the search for maximum likelihood is performed by pooling (grouping) *the data*. The problem (10) on the contrary means that we simply have an optimization problem with constraints and dimension m . The constraint is that all the step-probabilities are nonnegative. The solution therefore pools the adjacent *estimates* to make them satisfy the constraint. In this context the dynamic programming algorithm gives the same solution as the pooled-adjacent-violators algorithm (Barlow et al. (1972)) in particular cases when the problem (10) allows interpretation in terms of isotonic regression as for example in the case of doubly censored data (Subsection 3.2). The equivalence of the classes \mathcal{F} and $\bar{\mathcal{F}}$ means that by grouping the data

entering the adjacent intervals we get the same their joint contribution to the likelihood as by pooling the values of G in these intervals. However, the optimization with constraints may prove to be quite a difficult task. The algorithms suggested above allow to avoid the optimization with constraints by reducing the problem to a number of unconstrained ones.

In this context it is interesting to interpret the algorithms as a backward procedure. First consider a problem of optimizing ℓ_T in the class of step functions with m steps ($n = m$). If the solution happens to be at the border of the class, i.e. if either a step-probability or the length of some interval between the adjacent step-points turns to be zero, we have to decrement n and consider the problem of optimizing ℓ_T in the class of step functions with $m - 1$ steps according to (7). This step can be repeated until for the first time for some n we have a solution not at the border. It is this n that should be taken as optimal according to (9).

Remark. The procedures can be easily modified to adjust for constraints which might be dictated by the nature of the time to failure. For example, it might be known a priori that the distribution of time to failure has an increasing failure rate. Since in the course of solution all possible combinations of $\{\tau_i\}$ and $\{G_i\}$ are looked over it is not difficult to skip the combinations which are not consistent with the constraint.

2.4 One-side algorithm

Quite often the likelihood admit analytic maximization for fixed n, s_n . Let

$$\ell'_T(s_n) \stackrel{\text{def}}{=} \max_{e_n} \ell_T(s_n, e_n), \quad (11)$$

where the right part of (11) is available in a closed form. Then instead of (3) we have

$$\max_{n \leq m} \max_{s_n} \ell'_T(s_n). \quad (12)$$

Denote the solution of (11) by $e'_n(s_n)$. Then

$$\ell'_T(s_n) = \sum_{i=1}^n \varphi'_i(s_i),$$

where

$$\varphi'_i(s_i) = \varphi_i(s_i; e'_n(s_i)).$$

The problem (12) may be solved by a reduced version of the procedure described

in subsection 2.2. The dynamic programming algorithm will have the following common step. Let \hat{s}_k , $k < n$ be the solution of the problem

$$\max_{s_k \text{ given } \tau_{k+1}} \ell'_{\tau_{k+1}}(s_k).$$

Consider the optimal value $\hat{\ell}'_{\tau_{k+1}}$ and the vector \hat{s}_k as a function of τ_{k+1} . Then for \hat{s}_{k+1} we have

$$\hat{\ell}'_{\tau_{k+2}}(s_{k+1}) = \max_{\tau_{k+1} \text{ given } \tau_{k+2}} \{\hat{\ell}'_{\tau_{k+1}}(\tau_{k+1}) + \varphi'_{k+1}(\tau_{k+1})\}.$$

For any n the optimal solution \hat{s}_n , \hat{e}_n may be extracted by solving

$$\max_{\tau_n} \{\hat{\ell}'_{\tau_n}(\tau_n) + \varphi'_n(\tau_n)\},$$

where φ'_n contains the contribution of observations right censored by T .

If the equations (8) hold this procedure can be further simplified. Suppose in this context that the problem (11) can be solved by taking a derivative of ℓ_T :

$$\frac{\partial \ell_T}{\partial G_k} = 0$$

with e_n relieved of having to satisfy (2), the solution being unique. Denote such a solution by \tilde{e}_n . We may still use the above procedure substituting \tilde{e}_n instead of e' and excluding those s_n for which \tilde{e}_n does not satisfy (2). Indeed, if \tilde{e}_n does not satisfy (2) then maximum in (11) is reached at the border and according to (8) this means that we should look for solution with s_{n-1} .

The general algorithm described in subsection (2.2) can be represented as a combination of two one-side algorithms since the problem (11) can also be solved by a one-side algorithm with respect to e_n if a simpler solution is not available. Such an algorithm, in particular, solves a conventional problem of searching for the empirical distribution function with fixed n and s_n .

3 Examples

3.1 Right censored data. The life table estimate

Suppose that we have the sample $\{t_i, \delta_i\}$, where

$$\delta_i = \begin{cases} 1, & \text{if } t_i \text{ is uncensored} \\ 0, & \text{if } t_i \text{ is censored,} \end{cases}$$

$i = 1, \dots, m$. For such data

$$\varphi_i = m_i \log(\Delta G_i) + n_i \log(\bar{G}_i),$$

where

$$m_i = \sum_{k: t_k \in (\tau_{i-1}, \tau_i]} \delta_k = m_i(\tau_{i-1}, \tau_i),$$

$$n_i = \sum_{k: t_k \in (\tau_{i-1}, \tau_i]} (1 - \delta_k) = n_i(\tau_{i-1}, \tau_i),$$

$$\Delta G_i = G_i - G_{i-1} = \bar{G}_{i-1} - \bar{G}_i, \quad \bar{G}_i = 1 - G_i.$$

This presentation means that the original sample is modified by dragging all the observations entering the interval $(\tau_{i-1}, \tau_i]$ to the point τ_i . Solution of the problem (11) in this case is known as the life table estimate and is given by

$$\tilde{G}_i = 1 - \prod_{k=1}^i \frac{n_i + N_i}{N_{i-1}}, \quad i = 1, \dots, m, \quad (13)$$

where

$$N_j = m - \sum_{k=1}^j (m_k + n_k) = N_j(\tau_1, \dots, \tau_j).$$

The modified estimate can be obtained by solving (12) with

$$\tilde{\varphi}_i(s_i) = m_i \log(\Delta \tilde{G}_i) + n_i \log(1 - \tilde{G}_i),$$

where $\tilde{G}_i, \Delta \tilde{G}_i$ are given by (13) as functions of s_i .

As it was mentioned in the Subsection 2.3. we have an incorrect problem with the life table estimate. It turns out that the optimal function \hat{G} has only one step at the end of the study which is too far from reality. So, application of the algorithm is unreasonable in this particular case.

3.2 Doubly censored data

Suppose that we have a sample of either right or left noninformatively censored data $\{t_i, \delta_i\}$, where

$$\delta_i = \begin{cases} 1, & \text{if } t_i \text{ is right censored} \\ 0, & \text{if } t_i \text{ is left censored,} \end{cases}$$

$i = 1, \dots, m$. For such data

$$\varphi_i = m_i \log(G_i) + n_i \log(\bar{G}_i),$$

where

$$m_i = \sum_{k: t_k \in [\tau_i; \tau_{i+1})} \delta_k,$$

$$n_i = \sum_{k: t_k \in [\tau_i; \tau_{i+1})} (1 - \delta_k),$$

$i = 1, \dots, n$, $\tau_n = t_m$. If we solve (11) disregarding the constraint (2), we will get the following estimate

$$\tilde{G}_i = \frac{m_i}{m_i + n_i}. \quad (14)$$

Suppose (without loss of generality) that our sample consists of untied observations. Then it is clear from (14) that should we allow a step at each t_i with the estimate \tilde{G}_i , the estimate would look like a chaotic sequence of jumps from zero to one and back since in this case $\tilde{G}_i = \delta_i$. This indicates that we have a "noisy sample" and should not take its information that "literaly". This problem was addressed as early as in 1955 by Ayer et al. The estimate was considered as a function having exactly m steps and the step-probabilities were allowed to be zero, i.e. it was considered as belonging to the class $\bar{\mathcal{F}}$ (subsection 2.3). It can be easily verified that equations (8) do hold for this study design and the general solution by dynamic programming can be interpreted as the constrained maximization of the likelihood in the class $\bar{\mathcal{F}}$. However, in Ayer et al. (1955) another (more effective) algorithm was found to solve the problem in the context of constrained optimization for this particular case. The algorithm was named for the pooled-adjacent-violators one and it has prompted the development of isotonic theory (Barlow et al. (1972)). According to this algorithm, the estimate $\tilde{G}_i = \delta_i$ is a starting point. Then the following step is repeated in arbitrary order. If some adjacent values G_i, G_{i+1} violate the monotony property, they are pooled i.e. the step-probability at the point i is set to zero and both G_i and G_{i+1} are substituted by a new G_i , the subsequent estimates being newly enumerated starting from $i + 1$. It was shown by Ayer et al. (1955) that the resultant estimate is consistent and moreover is closer in average to the true distribution than \tilde{G}_i (which is not an asymptotic property).

The present paper was to a great extent prompted by the observation that the Ayer algorithm may be interpreted in terms of grouping. The mode of grouping suggested by the Ayer algorithm was used to furnish the parametric estimate of survivor function with focus on risk prediction to analyse the data on clean-up workers of the Chernobyl accident (Krouglikov et al. (1994)).

3.3 Discrete surveillance data

Suppose that a failure can be detected only by means of some test with probability p and that such tests are performed at times $\{t_i\}_{i=1}^m$ sorted in increasing

order. In addition we might have right noninformative censoring with respect to the time to detection. Such study design arise for example when a population of initially healthy individuals is repeatedly tested to detect cancers (cancer screening or surveillance) or in reliability theory when some units are tested to detect unobservable failures which cause damage. There are sorts of control problems which can be solved to optimize a surveillance strategy. We refer the reader to Beichelt and Franken (1983), Parmigiani (1993), Tsodikov and Yakovlev (1991), Tsodikov (1992) for such examples and focus on the statistical aspect of discrete surveillance.

The sample generated by discrete surveillance study consists of

N - the initial size of the target population;

m_i - the number of failures detected at the test performed at t_i ;

n_i - the number of right censored observations in the interval $[t_{i-1}, t_i)$.

With $p = 1$ this design turns to be equivalent to that related to the life table estimate. If in addition each individual is tested only once and t_i is the time of examination of the i -th individual, the design will be reduced to the doubly censored case.

In what follows we assume that $p < 1$ and t_i is the time when the whole population is tested all-at-once and that we have a sequence of such tests $i = 1, \dots, m$. If a failure occurs in some interval $[t_{i-1}, t_i)$ it will be detected at t_i with probability p , at t_{i+1} with probability $(1-p)p$, ... etc. In other words, the time of detection conditional on the failure entering $[t_{i-1}, t_i)$ is given by $t_{i-1} + \xi$, where ξ is a random variable following the geometric scheme with parameter p truncated by the last test t_m .

Introduce the unconditional distribution function Q of the time to detection of failure, which is related to the distribution of time to failure by the following recursive relations

$$G(t_i) - G(t_{i-1}) = \frac{1}{p}(\Delta Q_i - q\Delta Q_{i-1}), \quad i = 1, \dots, n, \quad q = 1 - p.$$

In this case

$$\ell = \sum_{i=1}^m \varphi_i, \quad \varphi_i = m_i \ln(\Delta Q_i) + n_i \ln(\bar{Q}_{i-1}), \quad i = 1, \dots, m-1,$$

$$\varphi_m = m_m \ln(\Delta Q_m) + n_m \ln(\bar{Q}_{m-1}) + N_m \ln(\bar{Q}_m).$$

This design was considered in Tsodikov et al. (1994) where the empirical distribution function was obtained disregarding the constraint (2) and with $\tau_i = t_i, i = 1, \dots, m, n = m$. The reasoning was quite simple. Using the invariance

property of the ML estimate it is possible to estimate the time to detection distribution function Q as a life-table estimate \hat{Q}

$$\hat{Q}_i = 1 - \prod_{k=1}^i \frac{N_k}{N_{k-1} - n_k}.$$

Then using equations expressing G in terms of Q it is easy to extract the estimate \tilde{G} . The resultant estimate may be written in a closed form

$$\Delta \tilde{G}_i = \frac{\prod_{k=1}^{i-1} N_k}{p \prod_{k=1}^i (N_{k-1} - n_k)} \left[m_i - qm_{i-1} + \frac{qm_{i-1}n_i}{N_{i-1}} \right], \quad (15)$$

where $N_k = N - \sum_{j=1}^k (m_j + n_j)$. Since the estimate $\Delta \tilde{G}_i$ may be represented by a linear combination of the life table ones, it inherits the properties of the latter, the consistency among them. To apply the one-side algorithm we need to generalize (15) to allow for the steps to occur arbitrary on the set $\{t_i\}_{i=1}^m$. Let λ_i be the number such that $\tau_i = t_{\lambda_i}$, $i = 1, \dots, n$, $n \leq m$, $\lambda_n = m$, $\lambda_0 = 0$. Introduce the step-functions Q' such that

$$Q_{\lambda_i} = Q'_i, \quad i = 1, \dots, n-1, \quad Q'_n = Q'_m, \quad \Delta Q'_0 = 0.$$

Then the estimate \tilde{G} will be given by

$$\Delta \tilde{G}_i = \frac{1}{p} [\Delta \tilde{Q}'_i - \Delta \tilde{Q}'_{i-1} q^{\lambda_i - \lambda_{i-1}}], \quad (16)$$

$i = 1, \dots, n$, $q = 1 - p$ and the estimates $\Delta \tilde{Q}'_i$ are obtained by the following procedure. For each $j = 1, \dots, n-1$ it is necessary to solve the algebraic equation

$$\frac{1}{r_j(1-r_j)} \left[(N - \sum_{i=1}^{\lambda_{j+1}-1} m_i - \sum_{i=1}^{\lambda_{j+1}} n_i) - r_j (N - \sum_{i=1}^{\lambda_j-1} m_i - \sum_{i=1}^{\lambda_{j+1}} n_i) \right] + \sum_{k=\lambda_j}^{\lambda_{j+1}-1} \frac{n_{k+1}}{r_j - q \frac{1-q^{k-\lambda_j}}{1-q^{k-\lambda_{j+1}}}} = 0 \quad (17)$$

with respect to r_j . Then

$$\tilde{Q}'_i = 1 - \prod_{k=1}^i r_k, \quad i = 1, \dots, n, \quad r_n = \frac{N_m}{N_m + m_m}.$$

The function φ'_i giving rise to the problem (12) can be written as

$$\begin{aligned} \varphi'_i &= M_i^{(1)} \ln(\Delta Q'_i) + M_i^{(2)} \ln(q) + n_{\lambda_i+1} \ln(\bar{Q}'_i) + \\ &\left\{ \begin{array}{l} \sum_{k=\lambda_i+1}^{\lambda_{i+1}-1} n_{k+1} \ln \left[\bar{Q}'_i - \Delta Q'_i q \frac{1-q^{k-\lambda_i}}{1-q} \right], \lambda_{i+1} > \lambda_i + 1 \\ 0, \text{ otherwise,} \end{array} \right. \quad (18) \\ M_i^{(1)} &= \sum_{k=\lambda_i}^{\lambda_{i+1}-1} m_k, \quad M_i^{(2)} = \sum_{k=\lambda_i}^{\lambda_{i+1}-1} m_k(k - \lambda_i), \end{aligned}$$

$$i = 1, \dots, n-1, \quad \varphi'_n = m_m \ln(\Delta Q'_n) + N_m \ln(\bar{Q}'_n).$$

The derivation of (16), (17) and (18) is outlined in the Appendix. It is not difficult to verify that in case $\lambda_i = i$, $i = 1, \dots, m$, $n = m$ we will have (15) instead of (16).

Another particular case is of interest. If the censoring is of type I (observations are censored only by the end of the study), i.e. if $n_i = 0$, $i = 1, \dots, m$, then the roots of (17) are available in the closed form

$$r_j = \frac{N_{\lambda_{j+1}} - 1}{N_{\lambda_j - 1}}, \quad j = 1, \dots, n-1, \quad r_n = \frac{N_m}{N_{m-1}},$$

and φ'_i are reduced to

$$\begin{aligned} \varphi'_i &= M_i^{(1)} \ln(\Delta Q'_i) + M_i^{(2)} \ln(q), \quad i = 1, \dots, n-1, \\ \varphi'_n &= m_m \ln(\Delta Q'_n) + N_m \ln(\bar{Q}'_n). \end{aligned}$$

It should be noted that the equations (8) hold for $p < 1$ and do not hold for $p = 1$ (the life table case). The discrepancy arises from the fact that for $p < 1$ each detected failure (at point τ_i) might have occurred at any time prior to detection, while for $p = 1$ it must be in $(\tau_{i-1}, \tau_i]$. So if for $p = 1$ the function G does not have a non zero step at τ_i the model happens to be inconsistent with the data.

For $p < 1$ we can use the one-side form of the exhaustive search algorithm (or the dynamic programming one as an approximate solution). As in the case of doubly censored data, the solution can be interpreted in terms of constrained optimization of the likelihood in $\bar{\mathcal{F}}$. The estimate (16) can be not monotone and the constraint is essential.

It is interesting to note that the monotony property is likely to be violated by (15) if the probability of mistake q is large. Indeed, in order that $\Delta \tilde{G}_i$ be nonnegative, $i = 1, \dots, m$ we have to demand that the expression in square brackets in (15) be nonnegative. This can be written as

$$m_i \geq qm_{i-1} \frac{N_{i-1} - n_i}{N_{i-1}}, \quad i = 1, \dots, m. \quad (19)$$

and the point becomes clear. For $p = 1$ ($q = 0$) the inequalities (19) hold automatically. With increase of q the amount of information in the sample $\{m_i, n_i\}$ generally decreases and this causes violation of the monotony property by (15) or (16). As in the case of doubly censored data, the algorithm provides an isotonic solution which helps to overcome the above difficulty.

3.4 Numerical example

For an example of application of the dynamic programming algorithm we have simulated a sample $\{x_i\}$ of 50 points from the two-parameter Gamma distribution with density

$$f(t) = \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}, \quad \alpha > 1, \quad t \geq 0.$$

The shape parameter α and the scale parameter β were taken to be 2.0 and 0.1, respectively. Then the simulated observed process was imposed to generate the observed data for the doubly censored design and for the discrete surveillance one. With the doubly censored data the times $\{t_i\}_{i=1}^m$, $m = 50$ were generated from the uniform distribution on the interval $[0,30]$ and the i -th member of the initial sample was taken as right censored if $x_i > t_i$ and left censored otherwise. With the discrete surveillance the test times $\{t_i\}_{i=1}^m$, $m = 10$ were taken at each 5-th point of the initial sample. Then the detection process was organized. For each failure entering the interval $[t_{i-1}, t_i)$, $i = 1, \dots, m$ the value of ξ was generated from geometric distribution (with $p=0.5$) truncated by $m - i + 1$. The time of detection was taken to be $t_{i-1} + \xi$. If undetected until t_m the observation was declared as censored by the end of the study. In figures 1 and 2 the results of estimation of the empirical survivor function are shown for the doubly censored design and for the discrete surveillance one, respectively. The (A) part of both figures contains the true curve and the estimate $1 - \bar{G}$ having steps at each t_i disregarding the monotony constraint (2). The part (B) presents the true curve and the estimates of the empirical survivor function resulting from application of the algorithm. From these figures it is evident that in both examples the estimate should be taken less detailed than the samples which are "noisy" and therefore unable to give enough information to make the estimate \bar{G} monotone. The estimates most closely corresponding to the amount of the samples' information are produced by the algorithm (figures 1B and 2B).

4 Conclusion

The proposed approach is primarily oriented to the analysis of small (or low information) samples. An indication of such a case could be either the violation of monotony by \tilde{G} or its instability to the choice of grouping. With the large samples with enough information for the asymptotic ML theory to be applied with respect to \tilde{G} , we do not need these algorithms, because in this case \tilde{G} is monotone and stable as a consistent estimate and therefore can be made as detailed as desired. However, for example in medical applications this is not the case more often than not. The field of application of the proposed approach is not limited to searching for the empirical distribution function. It might be reasonable for example to minimize the chi-square statistics when testing a composite parametric hypothesis with respect to grouping n, s_n thus avoiding the problems associated with instability of the conventional test to the choice of grouping. However, the asymptotic distribution of the modified statistics may turn to be other than χ^2 .

Appendix

We are going to derive (16),(17) and (18). In doing so we proceed from the likelihood in the form

$$\ell = \sum_{i=1}^m [m_i \ln(\Delta Q_i) + n_i \ln(\bar{Q}_{i-1})] + N_m \ln(\bar{Q}_m), \quad (20)$$

where the time to detection distribution Q and the time to failure distribution G are linked by

$$\Delta Q_i = \bar{Q}(t_{i-1}) - \bar{Q}(t_i) = p(\bar{G}(t_{i-1}) - \bar{G}(t_i)) + q\Delta Q_{i-1}. \quad (21)$$

Since the function G actually has steps at points $\tau_i = t_{\lambda_i}$, $i = 1, \dots, n$, we have

$$\bar{G}(t_{j-1}) - \bar{G}(t_j) = 0, \quad j = \lambda_i, \lambda_i + 1, \dots, \lambda_{i+1} - 1, \quad i = 1, \dots, n - 1. \quad (22)$$

Combining (21) and (22) we get

$$\Delta Q_{\lambda_i} \stackrel{\text{def}}{=} \Delta Q'_i; \quad \Delta Q_{\lambda_i+1} = \Delta Q'_i q, \dots, \Delta Q_{\lambda_{i+1}-1} = \Delta Q'_i q^{\lambda_{i+1}-\lambda_i-1}, \quad (23)$$

$$i = 1, \dots, n - 1, \quad \Delta Q_m \stackrel{\text{def}}{=} \Delta Q'_n.$$

The likelihood (20) may be rewritten in the form

$$\ell = \sum_{i=1}^n \varphi'_i,$$

$$\varphi'_i = \sum_{k=\lambda_i}^{\lambda_{i+1}-1} \varphi_k, \quad i = 1, \dots, n - 1; \quad \varphi'_n = \varphi_{\lambda_n},$$

$$n_{m+1} \stackrel{\text{def}}{=} N_m, \quad \varphi_i = m_i \ln(\Delta Q_i) + n_{i+1} \ln(\bar{Q}_i), \quad i = 1, \dots, m.$$

Using (21),(22) and (23) after a little algebra we get expressions (18) for φ'_i in terms of $\Delta Q'_i$.

Introduce the conditional survivor probabilities

$$r_k = \frac{\bar{Q}'_k}{\bar{Q}'_{k-1}}, \quad k = 1, \dots, n. \quad (24)$$

Then it is possible to change the variables $\Delta Q'_i$ for r_i , $i = 1, \dots, n$. We have

$$\bar{Q}'_i = \prod_{k=1}^i r_k,$$

$$\Delta Q'_i = \prod_{k=1}^{i-1} r_k (1 - r_i), \quad i = 1, \dots, n, \quad \prod_1^0 = 1. \quad (25)$$

The expressions for φ'_i in terms of r_i result from substitution of (25) in (18).

$$\varphi'_i = (M_i^{(1)} + N_i^{(1)}) \sum_{j=1}^{i-1} \ln(r_j) + M_i^{(1)} \ln(1 - r_i) +$$

$$\sum_{k=\lambda_i}^{\lambda_{i+1}-1} n_{k+1} \ln \left[r_i - q \frac{1 - q^{k-\lambda_i}}{1 - q^{k-\lambda_i+1}} \right] + C,$$

$$C = M_i^{(2)} \ln(q) + N_i^{(2)} - N_i^{(1)} \ln(p),$$

$$N_i^{(1)} = \sum_{k=\lambda_i}^{\lambda_{i+1}-1} n_{k+1}; \quad N_i^{(2)} = \sum_{k=\lambda_i}^{\lambda_{i+1}-1} n_{k+1} \ln(1 - q^{k-\lambda_i+1}), \quad i = 1, \dots, n-1, \quad \sum_1^0 = 0,$$

$$\varphi'_n = (m_m + N_m) \sum_{j=1}^{n-1} \ln(r_j) + m_m \ln(1 - r_n) + N_m \ln(r_n).$$

Taking the derivative

$$\frac{\partial \ell}{\partial r_j} = \sum_{i=j+1}^n \frac{\partial \varphi'_i}{\partial r_j}$$

and making it zero we get equations (17).

The estimates $\Delta \tilde{Q}'_i$ are obtained by substituting the roots r_j of (17) in (25). The relationships (16) follow from (21) and (23) with \tilde{Q}' instead of Q' .

References

- Ayer, M., Brunk, H.D., Ewing, G.M., Reid, W.T. and E. Silverman, An empirical distribution function for sampling with incomplete information, *Annals of Mathematical Statistics*, **26** (1955) 641-647.
- Barlow, R.E., Bartholomew, D.J., Bremner, J.M. and H.D. Brunk, *Statistical Inference Under Order Restrictions*, (Wiley, New York, 1972).
- Beichelt, F. and P. Franken, *Zuverlässigkeit und Instandhaltung*, (VEB Verlag Technik, Berlin, 1983).
- Cox, D.R. and D. Oakes, *Analysis of survivor data*, (Chapman and Hall, London, 1983).
- Efron, B., The two-sample problem with censored data, in: *Proceedings of the 5-th Berkeley Symposium in Mathematical Statistics IV*, (Prentice-Hall, New York, 1967) 831-853.
- Krouglikov, I., Pilipenko, N., Tsodikov, A., and A. Yakovlev, Assessing risk with doubly censored data: An application to the analysis of radiation induced thyrothyopathy, submitted to *Biometrical Journal*.
- Parmigiani, G., On optimal screening ages, *Journal of the American Statistical Association*, **88** (1993) 622-628.
- Tsodikov, A.D., Screening under uncertainty. Games approach, *Systems Analysis. Modeling and Simulation*, **9** (1992) 259-262.
- Tsodikov, A.D., Asselain, B., Fourquet, A., Hoang, T. and A.Yu. Yakovlev, Discrete strategies of cancer post-treatment surveillance. Estimation and optimization problems, *Biometrics*, (1994) to appear.
- Tsodikov, A.D. and A.Yu. Yakovlev, On the optimal policies of cancer screening, *Mathematical Biosciences*, **107** (1991) 21-45.
- Turnbull, B.W., The empirical distribution function with arbitrarily grouped, censored and truncated data, *Journal of the Royal Statistical Society, Series B*, **38** (1976) 290-295.

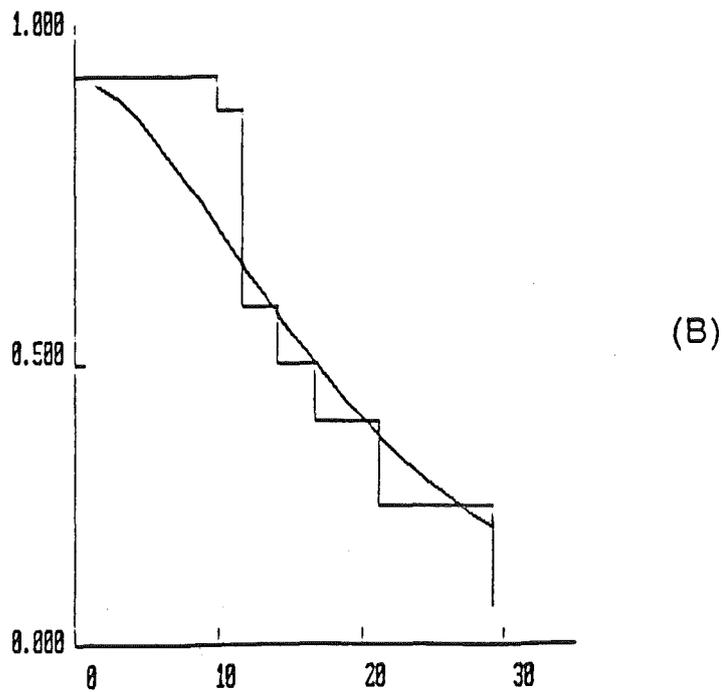
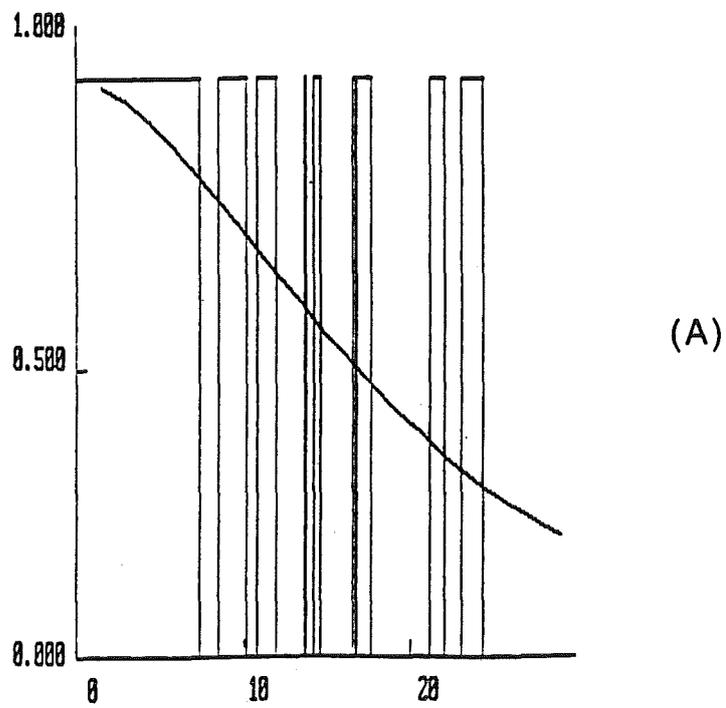
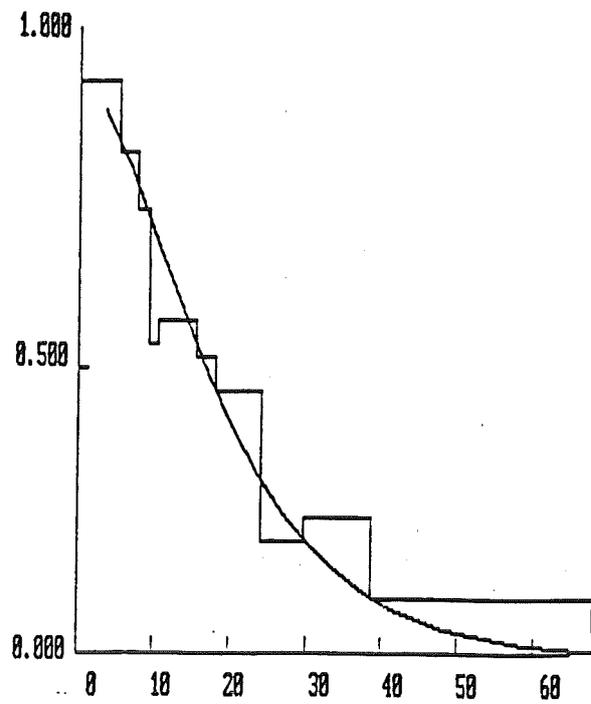
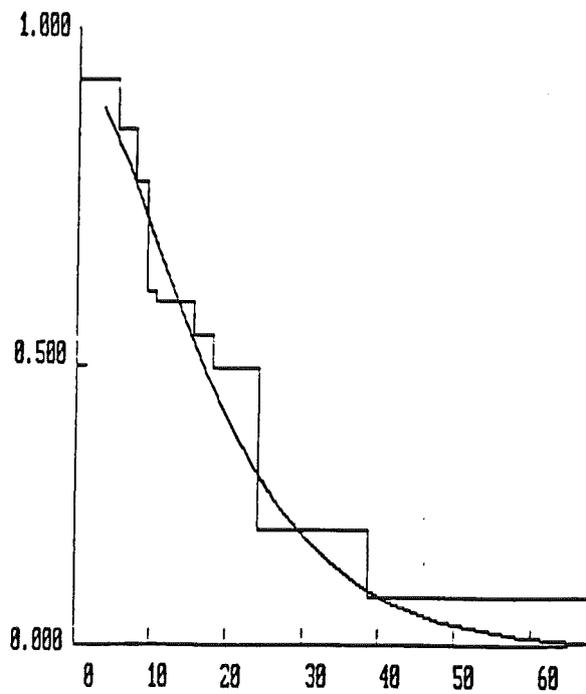


Figure 1. Survivor curves with doubly censored data. Solid line is the "true" survivor function used in computer simulations; stepwise curves are nonparametric estimates $1 - \tilde{G}$ (A) and the one resulting from application of the dynamic programming algorithm (B).

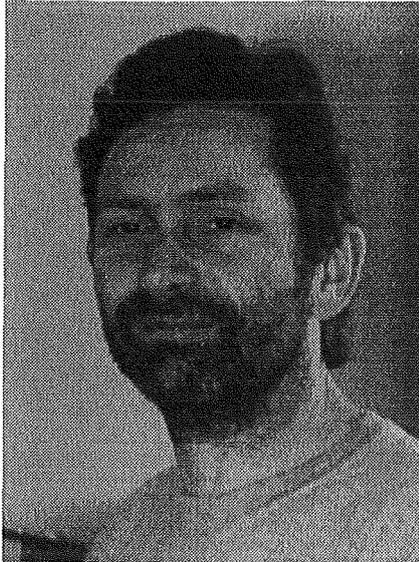


(A)



(B)

Figure 2. Survivor curves with discrete surveillance data. Solid line is the "true" survivor function used in computer simulations; stepwise curves are nonparametric estimates $1 - \tilde{G}$ (A) and the one resulting from application of the dynamic programming algorithm (B).



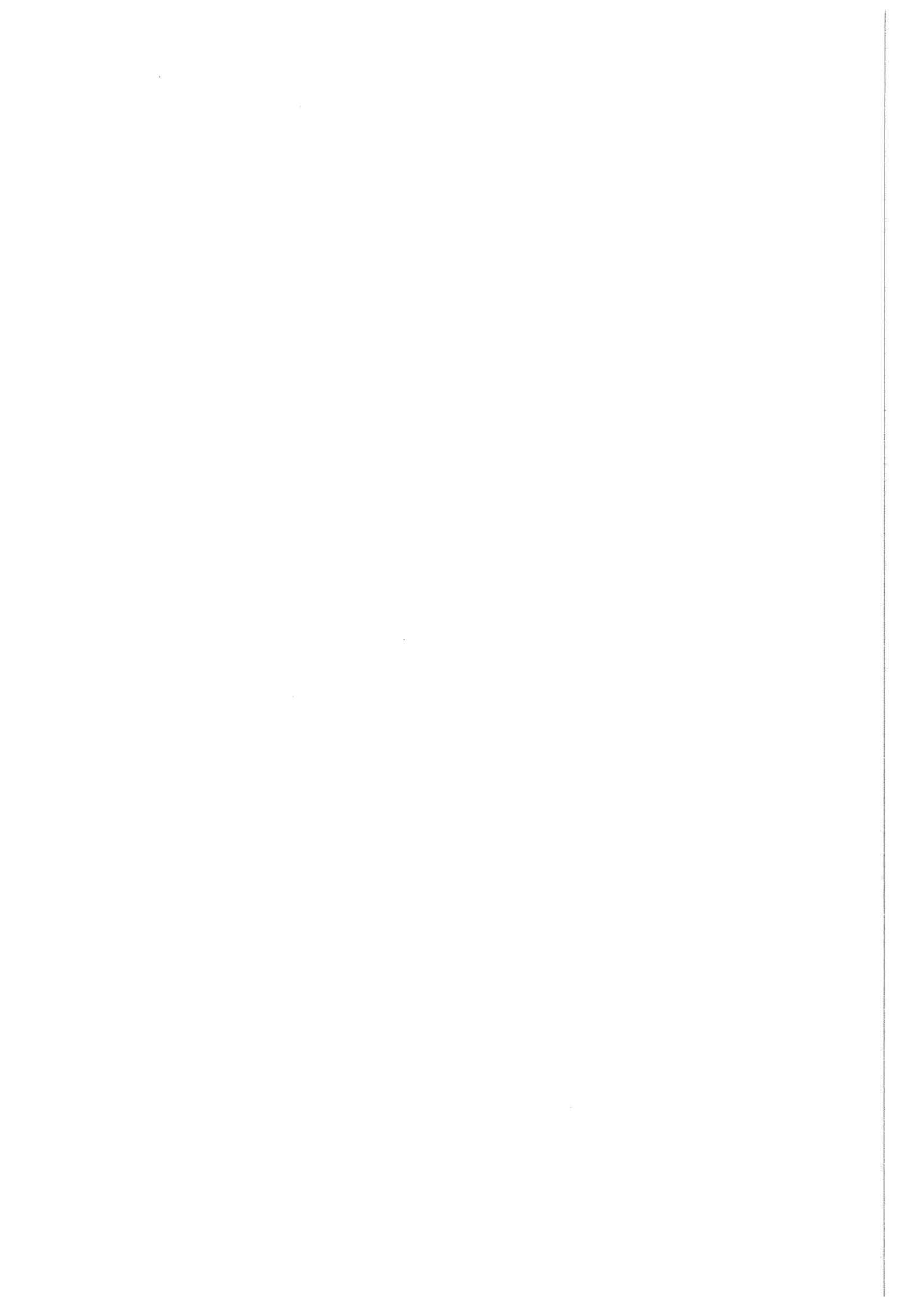
Dr. Jens Weidner:

Zum Vortrag und zur Person

Um aussagekräftige, numerische Berechnungen bei technischen Anwendungen durchführen zu können, müssen die auftretenden, komplizierten Geometrien realistisch im numerischen Modell wiedergegeben werden. Dabei ist es oftmals nicht ausreichend, die Geometrien mit monoblock-strukturierten Gittern zu erfassen, sondern man ist gezwungen zumindest auf eine block-strukturierte Gitterzerlegung überzugehen.

Im Zusammenhang mit der EASI-Kooperation zwischen dem Kernforschungszentrum Karlsruhe und der IBM Deutschland hat Dr. Weidner Methoden untersucht, um komplizierte Geometrien durch block-strukturierte Gitter zu diskretisieren, um anschließend die Potentialgleichung mit Hilfe des Schwarzschen Prinzips zu lösen.

Dr. Weidner studierte Diplom-Mathematik mit Nebenfach Physik an den Universitäten Kiel, Heidelberg, Philadelphia, Marseille und Heidelberg mit den Schwerpunkten algebraische Topologie und Operatoralgebren. 1987 promovierte er an der Fakultät für Mathematik der Universität Heidelberg. Nach drei Jahren Aufenthalt als Nachwuchswissenschaftler am mathematischen Institut wechselte er 1990 zur IBM Heidelberg.



Rechnen in Komplexen Geometrien

Dr. Jens Weidner



Heidelberg Scientific and Technical Center

Tel.: 06221/594486

e-mail: weidner@heidelberg.ibm.com

November 1994

Zusammenfassung

Macht man Simulationen in häufig veränderten Geometrien - beispielsweise bei der Optimierung der Geometrie eines Bauteils oder Apparates - so erfordert das Erzeugen geeigneter Gitter enormen Aufwand. In manchen Fällen kann man mit Gebietszerlegungsmethoden modular aus einfachen Geometrien schnell sehr komplexe Geometrien erzeugen. Die Simulation verwendet dann Algorithmen aus der parallelen Programmierung. Am Beispiel des Schwarz'schen Verfahrens soll die Methode erläutert werden.

Inhaltsverzeichnis

1	Einleitung	2
2	Das Schwarz'sche Verfahren	2
3	Etwas mathematische Theorie	4
4	Diskretisierung	5
5	Ein numerisches Beispiel - das Knie	6
6	Parallele Verarbeitung	14
7	Nachbemerkung	14

1 Einleitung

Die hier skizzierten Ideen zur Behandlung elliptischer Differentialgleichungen in komplexen Geometrien entstand in einer Kooperation des Wissenschaftlichen Zentrums Heidelberg der IBM mit dem Kernforschungszentrum Karlsruhe. Ziel war es damals, die Simulation von Hochstromdioden zu erleichtern. Wir wollen hier nur die grundlegenden Ideen skizzieren und die Anwendung der Methode an einem Beispiel vorführen.

Zunächst soll die Methode der überlappenden Gebietszerlegung erläutert werden. Anschließend an ein paar Bemerkungen zur mathematischen Theorie des Verfahrens soll dann das Diskretisierungsverfahren erläutert werden. Abschließend diskutieren wir die numerischen Ergebnisse auch im Hinblick auf Parallelverarbeitung.

2 Das Schwarz'sche Verfahren

Das nach Hermann A. Schwarz benannte Alternierende Verfahren, eine elliptische lineare Differentialgleichung auf zusammengesetzten Gebieten zu lösen, entstand ursprünglich aus dem Bedürfnis, überhaupt die Existenz von Lösungen der Laplace-Gleichung in komplexen Geometrien zu garantieren([5]). Dadurch wurde das Dirichlet'sche Prinzip umgangen, dessen Gültigkeit für die

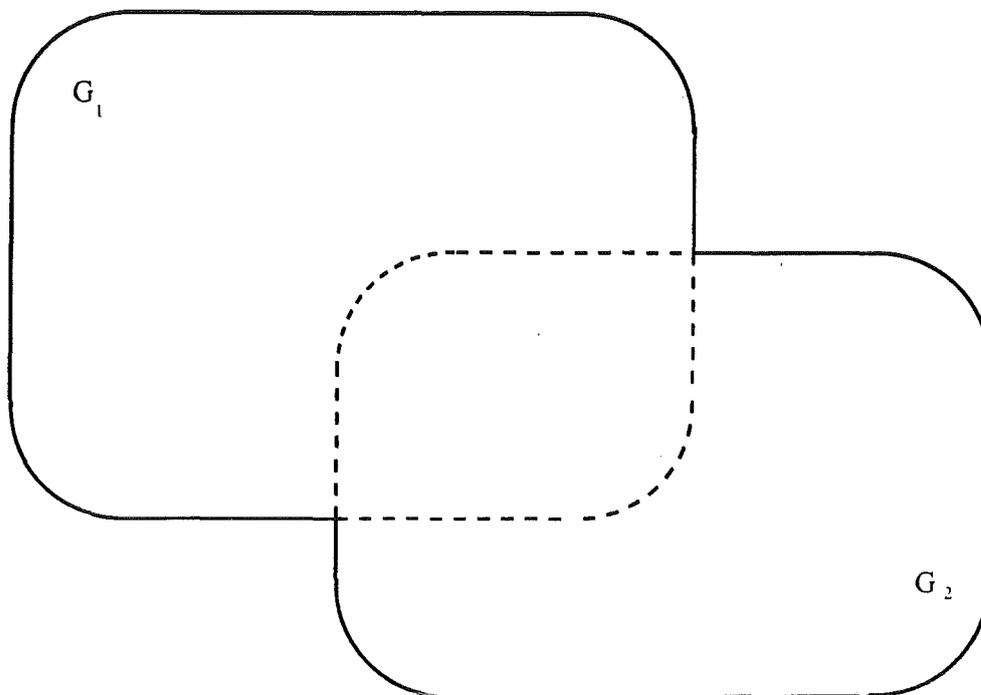


Abbildung 1: Gebietszerlegung mit zwei Gebieten

Laplace-Gleichung erst D. Hilbert nachweisen konnte. Mit dem Aufkommen der Parallelverarbeitung wurde das Verfahren erneut diskutiert ([3]). Für die Parallelverarbeitung ist das Verfahren zu langsam und zahlreiche Verbesserungen wurden vorgeschlagen ([1],[2],[4]). Weil wir uns hier nur für die Behandlung komplexer Geometrien interessieren, diskutieren wir hier nur das ursprüngliche multiplikative Verfahren. Der Einfachheit halber beschränken wir uns auf das reine Randwertproblem und auf zwei Teilgebiete.

Betrachte auf dem zusammengesetzten Gebiet $G = G_1 \cup G_2$ mit Rand ∂G das elliptische Randwertproblem bei gegebenen Funktionen f auf G und h auf ∂G :

$$\begin{aligned} \Delta u &= f \quad \text{in } G \\ u &= h \quad \text{auf } \partial G \end{aligned} \tag{1}$$

Der iterative Prozess entsteht durch sukzessives Anwenden zweier Operatoren, der sogenannten Schwarz Updates S_1 und S_2 . S_1 und S_2 operieren auf

einer geeigneten Klasse von Funktionen auf G mit Randwerten h . Dabei ist der Schwarz-Update $S_1(u) = v$ die Lösung des Problems:

$$\begin{aligned}\Delta v &= f && \text{in } G_1 \\ v &= u && \text{auf } G \setminus G_1\end{aligned}$$

und $S_2(u) = v$ die Lösung des Problems:

$$\begin{aligned}\Delta v &= f && \text{in } G_2 \\ v &= u && \text{auf } G \setminus G_2\end{aligned}$$

Wähle eine Startfunktion u_0 auf G mit den richtigen Randwerten: $u_0 = h$ auf ∂G . Setze dann:

$$\begin{aligned}u_{2n+1} &= S_1(u_{2n}) \\ u_{2n} &= S_2(u_{2n-1})\end{aligned}$$

für $n = 1, 2, 3, \dots$

Die Frage ist nun, unter welchen Bedingungen an G_1, G_2 und die Funktionen f, u_0 und h die Operatoren S_1, S_2 existieren, und ob dann die Folge der $u_n, n = 1, 2, 3, \dots$ konvergiert gegen die Lösung von (1).

3 Etwas mathematische Theorie

Wir formulieren zunächst das Problem um. Dazu multiplizieren wir die Gleichung $\Delta u = f$ mit einer Testfunktion v mit verschwindenden Randwerten, integrieren über G und machen eine partielle Integration:

$$\int_G \nabla u \cdot \nabla v = - \int_G \Delta u \cdot v = - \int_G f \cdot v$$

Der folgende Hilbertraum ist daher besonders geeignet für die Analyse: $H^1(G)$ der Raum der L^2 -Funktionen mit L^2 -Ableitungen, die auf dem Rand verschwinden. Ebenso sollte $f \in H^{-1}(G)$ sein. Entsprechend sollte h als Randwert von u in $H^{1/2}(\partial G)$ sein.

Wenn wir das Schwarz Verfahren beurteilen wollen, interessiert uns der Fehler $u_n - u_{n-2}$ benachbarter Iterationen bzw. der Fehler zur wirklichen Lösung u . Diese Differenzen liegen in $H = H_0^1(G)$ der Raum der L^2 -Funktionen mit L^2 -Ableitungen, die auf dem Rand verschwinden. H ist auch der Raum der Testfunktionen v . Der Raum H hat ein besonders einfaches Skalarprodukt:

$$\langle u, v \rangle = \int_G \nabla u \cdot \nabla v$$

Der Raum $H_+ = H_0^1(G_1)$ kann nun als Unterraum von H angesehen werden, indem man Funktionen in H_+ durch 0 auf ganz G fortsetzt. Von besonderer Bedeutung sind die beiden Projektoren P_1 bzw P_2 von H auf H_1^\perp bzw H_2^\perp :

Satz 1 Die asymptotische Konvergenzrate

$$AS = \lim_{n \rightarrow \infty} \|u_n - u\| / \|u_{n-2} - u\|$$

ist gegeben durch den Spektralradius des Operators $P_1 \cdot P_2$. Weiter ist

$$AS = \|P_1 P_2 P_1\| = \|P_1 P_2\|^2 = \|P_2 P_1\|^2$$

Satz 2 Ist M ein beschränkter Operator auf H mit der Eigenschaft $P_1 \cdot M = 0$ und $P_2 \cdot (1 - M) = 0$, so ist

$$AS \leq 1 - 1/\|M\|^2$$

Operatoren M erhält man auf geometrischem Wege. Ist zum Beispiel χ eine C^∞ -Funktion mit Werten in $[0, 1]$, $\chi = 0$ in $G \setminus G_2$ und $\chi = 1$ in $G \setminus G_1$ so kann man für M den Multiplikationsoperator mit χ wählen.

Auf diese Weise erhält man für eine große Klasse von Gebieten die Konvergenz des Multiplikativen Schwarz Verfahrens.

4 Diskretisierung

Betrachten wir nochmals die schwache Formulierung unseres Problems:

$$\int_G \nabla u \cdot \nabla v = - \int_G f \cdot v \quad \text{für alle } v \in H \quad (2)$$

Am einfachsten schränkt man sich zur Diskretisierung auf einen endlich-dimensionalen Unterraum H_d von H ein und sucht dort eine Lösung. Wählt man eine Basis von H_d , so entsteht aus (2) ein lineares Gleichungssystem mit einer positiv definiten reellen Matrix.

Es liegt jetzt nahe für H_d einen finiten Elementraum zu nehmen. Die Basisfunktionen haben dann einen "kleinen" Träger, so daß die resultierende Matrix dünn besetzt ist.

In unserer zweidimensionalen Implementierung haben wir einfache Dreieckselemente gewählt und zudem vorausgesetzt, daß die Teilgebiete logisch Rechtecke sind. Dadurch erhält die Matrix eine Bandstruktur, für deren Cholesky Faktorisierung schnelle Systemroutinen zur Verfügung stehen (z.B. in der ESSL-Bibliothek auf den IBM RS/6000 Systemen).

Die bisher beschriebene Methode setzt voraus, daß das Dreiecksgitter auf den Gebietsüberlappungen übereinstimmt. Davon kann man abweichen, muß aber dann Funktionen auf dem einen Gitter in das andere Gitter interpolieren. Das nächste Bild zeigt schematisch, wie eine solche Gebietszerlegung aussehen kann. Im nächsten Kapitel werden wir ein realistischeres Beispiel studieren.

5 Ein numerisches Beispiel - das Knie

Bei der Implementierung des Verfahrens fielen folgende Gesichtspunkte auf:

1. Die Berechnung der Interpolationsdaten für den Gebietswechsel ist sehr aufwendig. Will man daher mehrfach Simulationen in der gleichen Geometrie machen, so sollte man die Interpolationsdaten nur einmal berechnen und dann aufheben.
2. Das Verfahren ist außerordentlich stabil, man kann für den Fall, daß die Gitter auf den Durchschnitten der Gebiete übereinstimmen mit dem Abbruchkriterium für das Schwarz Verfahren in die Nähe der Maschinengenauigkeit gehen. Im allgemeinen Fall entsteht durch die Interpolation ein systematischer Fehler, weil man beim hin- und herinterpolieren einer Funktion diese nicht wieder zurückgewinnt. Dieser Interpolationsfehler begrenzt die Genauigkeit.

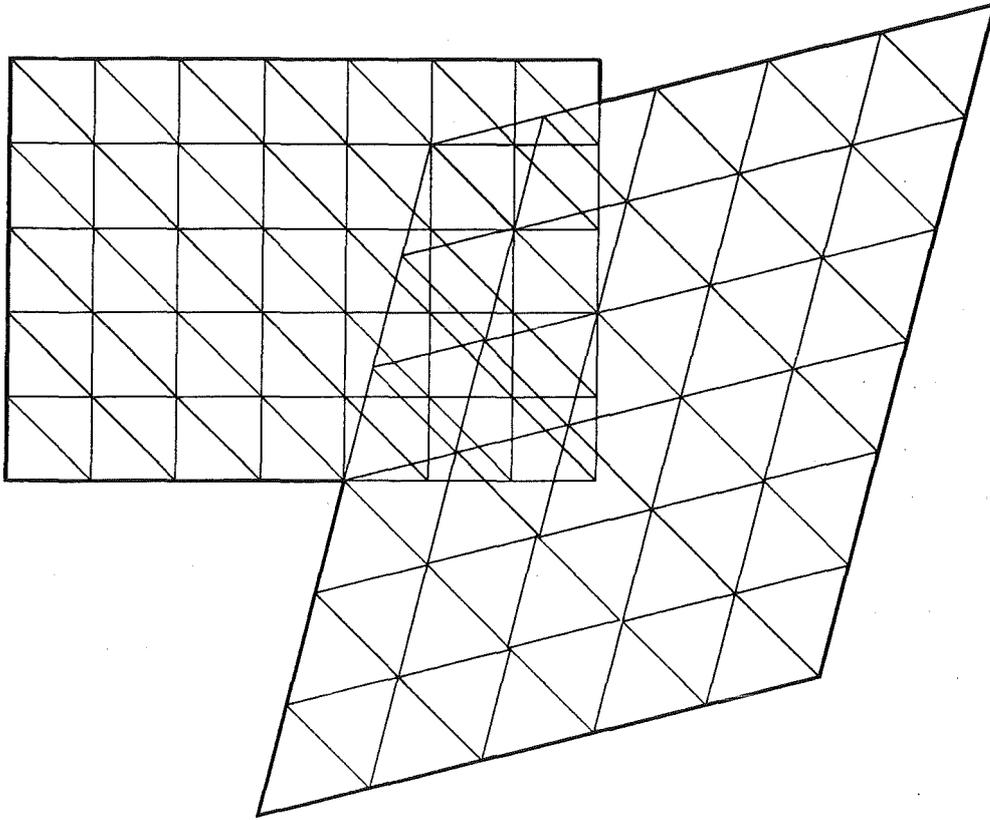


Abbildung 2: Gebietszerlegung mit finiten Elementen

3. Im Kontinuumsproblem muss die rechte Seite f der Gleichung nur sehr schwache Regularitätsforderungen erfüllen. Das zeigen auch die numerischen Experimente.
4. Ohne Schwierigkeit können mit dem gleichen Verfahren gemischte Randwertprobleme gelöst werden.
5. Weil man in verschiedenen Gebieten unterschiedliche Feinheiten der Gitter zulassen kann, kann man auf diese Weise zumindest statisch die Gittergrößen adaptieren.

Das Testbeispiel ist ein künstlich ersonnenes Problem, das aber ein für die Simulation der Hochstromdioden typisches singuläres Verhalten hat.

Das nächste Bild zeigt die Geometrie des Rechengebietes und seine Einteilung in sechs Teilgebiete. Das Problem ist ein gemischtes Randwertproblem, die verschiedenen Typen der Ränder sind dargestellt.

Gelöst werden soll in dieser Geometrie die Gleichung

$$\Delta u = f$$

wobei angenommen wird, daß es sich um ein Zylinder symmetrisches Problem handelt, also f und somit auch u nur von r - und z -Komponente abhängen. Eine weitere Besonderheit ist, daß f auf dem äußeren der beiden Dirichlet Ränder singulär wird mit $d^{-2/3}$, wenn d der Abstand zum Rand ist. Eine zusätzliche Schwierigkeit entsteht dadurch, daß die gesuchte Lösung eine Singularität in den ersten Ableitungen längs einer Linie aufweist (Knicklinie). Die numerische Simulation zeigt nun, daß man bei insgesamt ca. 6000 Gitterpunkten die exakte Lösung bis auf 10^{-5} approximiert werden kann. Dabei weist die rechte Seite der Gleichung am äußeren Rand bereits Datenunterschiede von 1000 zwischen benachbarten Zellen auf, während für die Lösung u gilt: $0 \leq u \leq 1$.

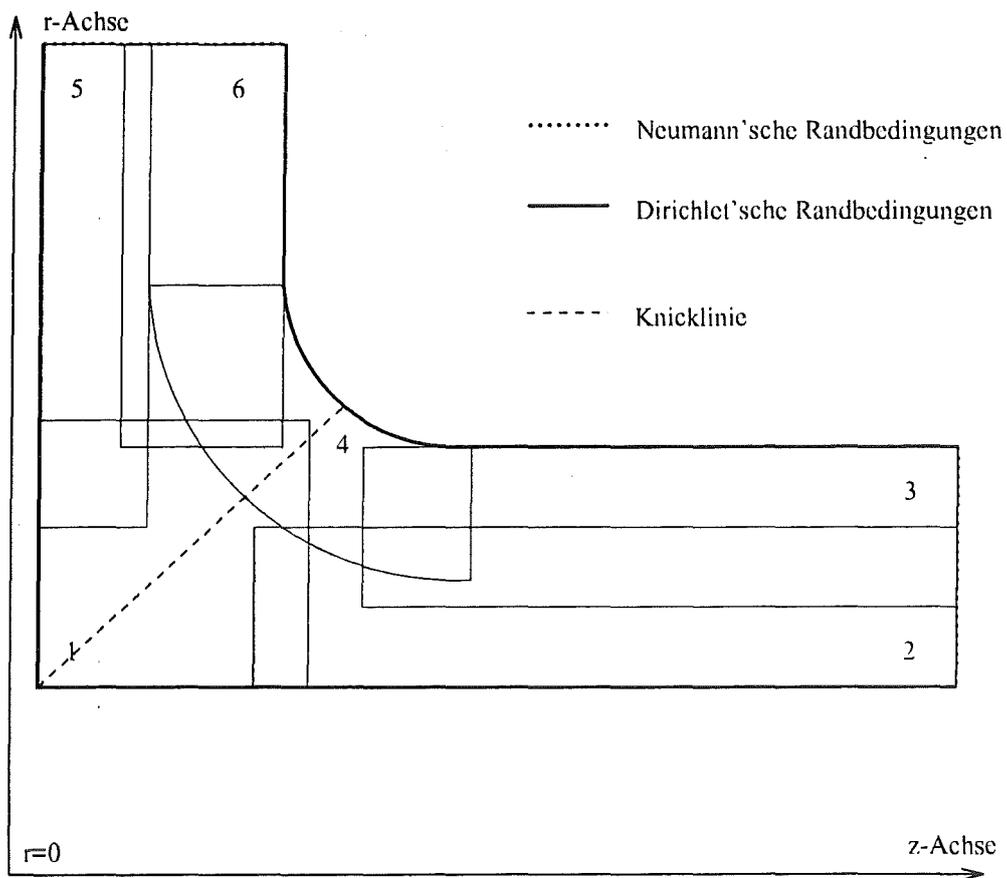


Abbildung 3: Die Zerlegung der Kniegeometrie

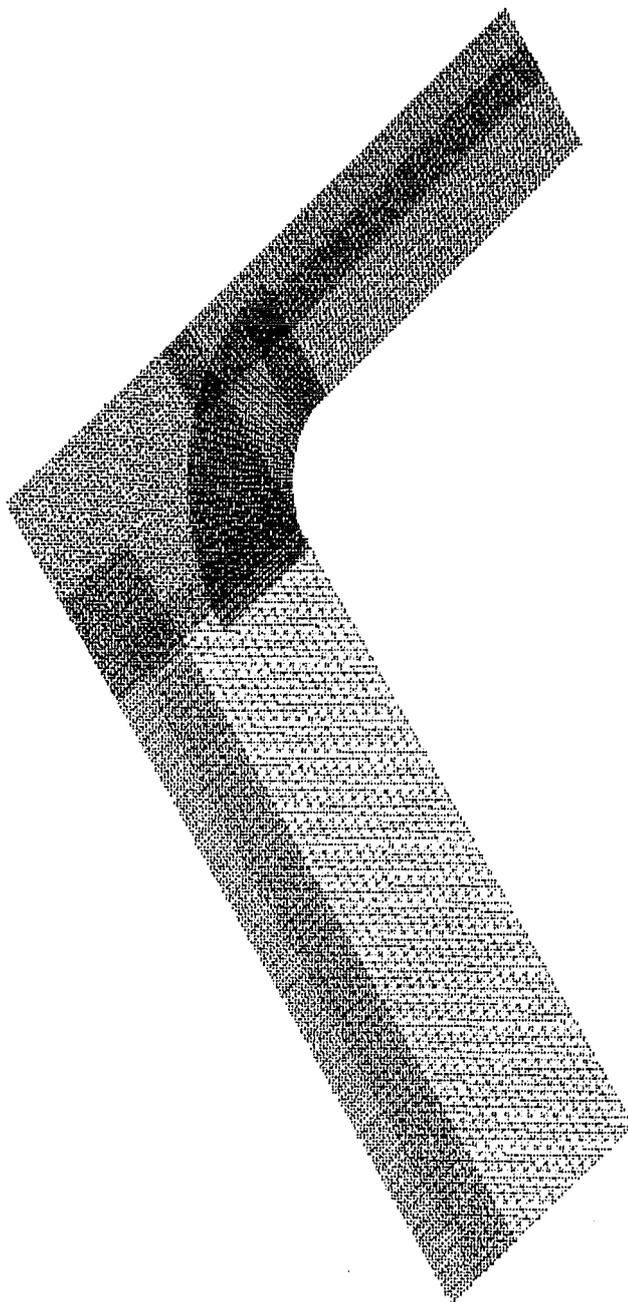


Abbildung 4: Das Gitter

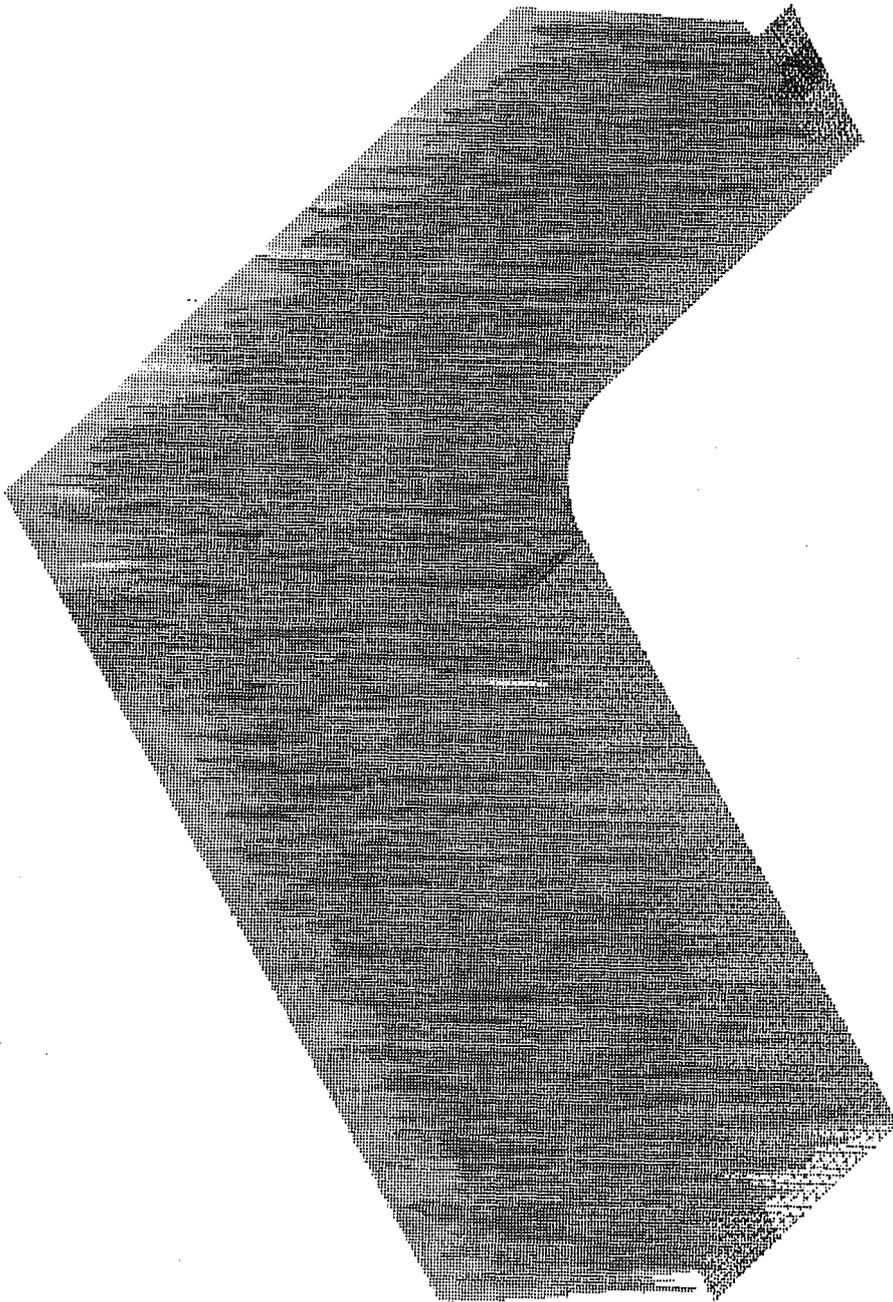


Abbildung 5: Anfängliche Zufallsverteilung

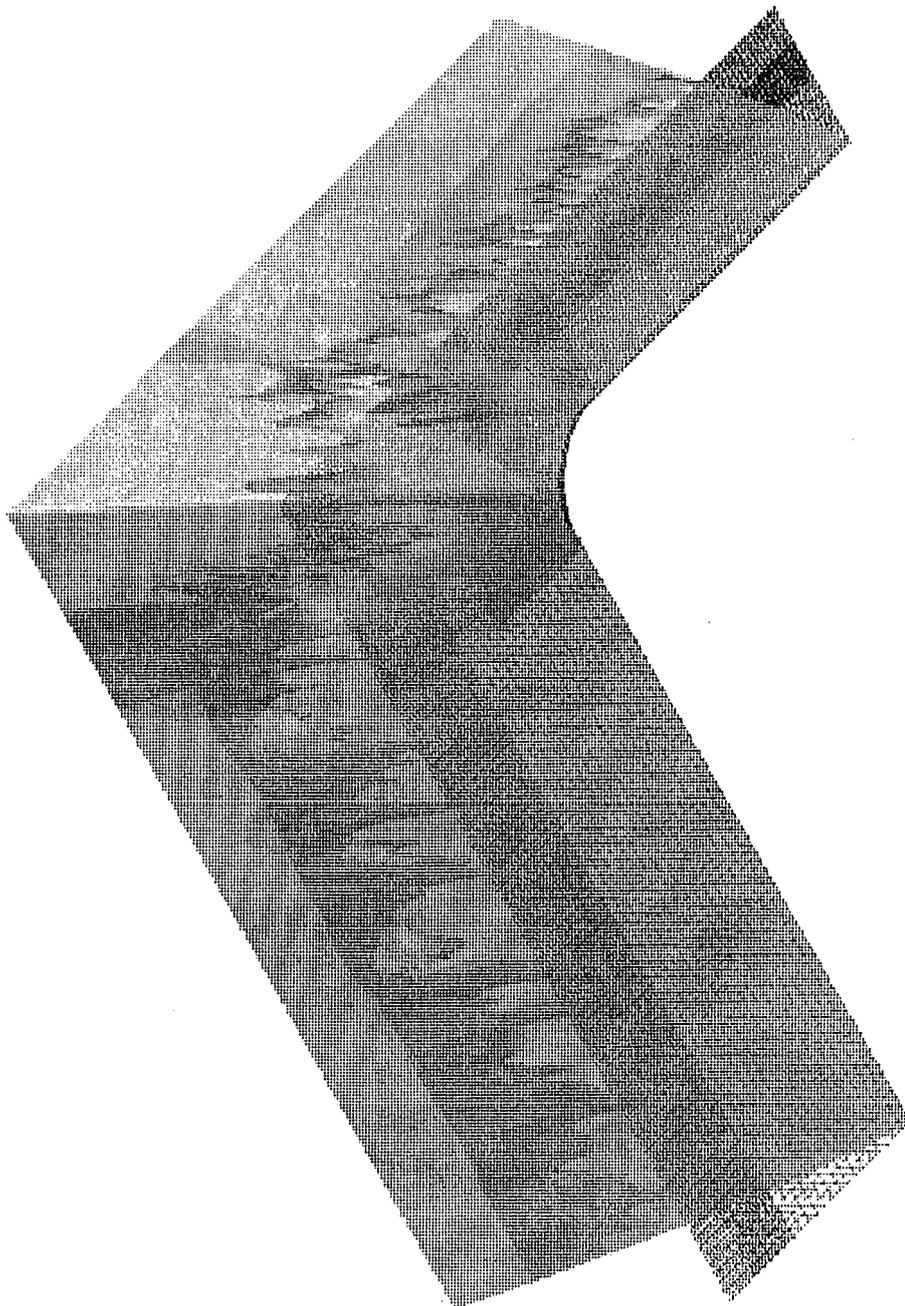


Abbildung 6: Nach dem ersten Schwarz-Schritt

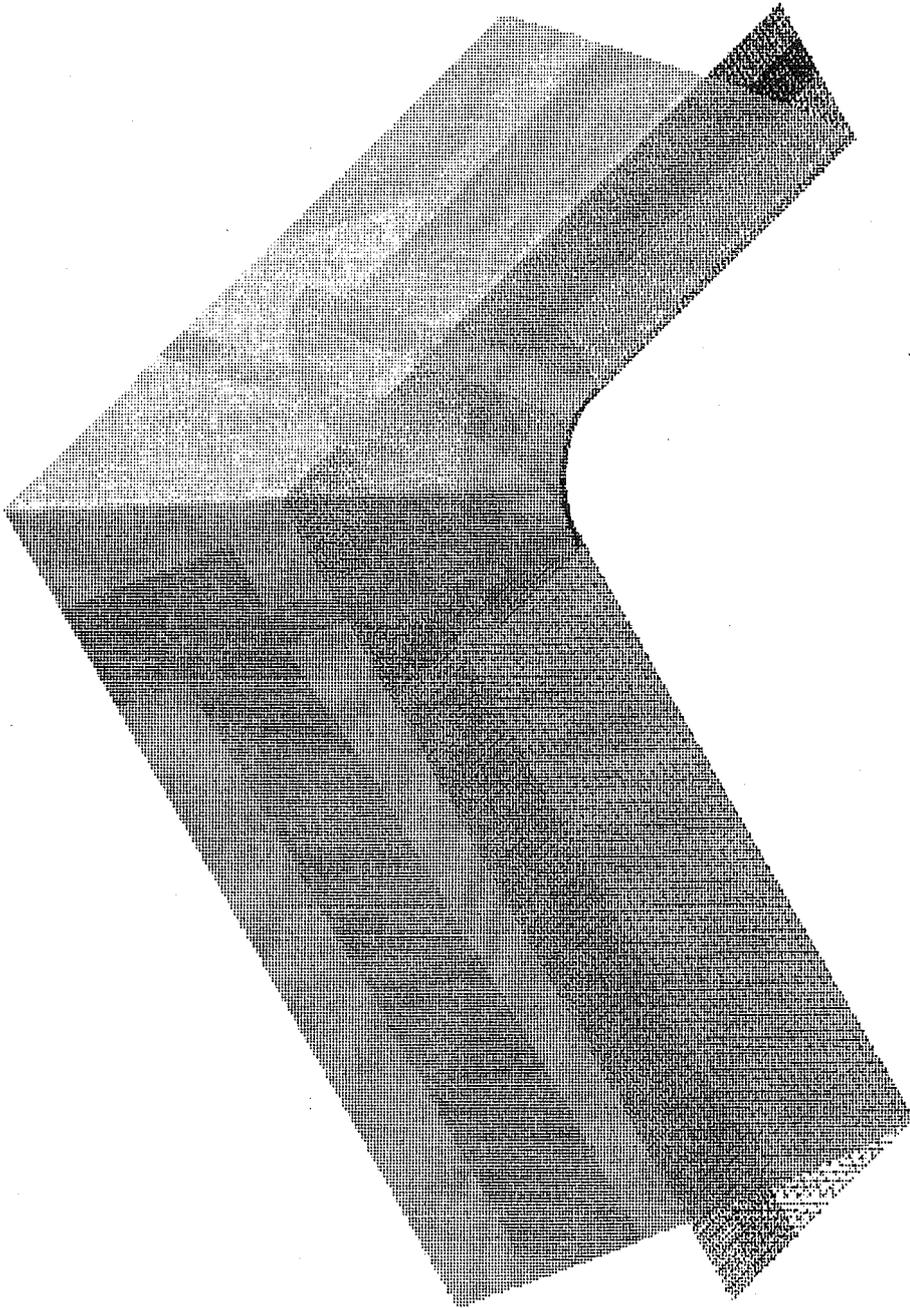


Abbildung 7: Nach dem zweiten Schwarz-Schritt

6 Parallele Verarbeitung

Um zu einem parallelen Algorithmus zu gelangen, kann man jetzt die Schwarz Updates für jedes Einzelgebiet auf einem anderen Prozessor rechnen, d.h. man weist jedem Prozessor genau ein oder auch mehrere Gebiete zu. Man kann sich überlegen, daß die dadurch entstehenden Lösungen auf den Einzelgebieten auf den Gebietsdurchschnitten im allgemeinen nicht übereinstimmen. Die Abweichungen konvergieren jedoch gegen 0. Für jeden neuen Schritt benötigt ein Prozessor jeweils nur die Dirichlet Daten auf den inneren Gebietsrändern. Das ist ein niederdimensionales Gebilde. Daher eignet sich der Algorithmus für eine Implementierung auf Parallelrechner mit verteiltem Speicher (z.B. IBM SP2). Eine solche auf Message Passing basierende Parallelisierung haben wir in Heidelberg implementiert, wobei verschiedenen Interfaces getestet wurden (EXPRESS, IBM MPL, PVM3). Wie oben schon erwähnt ist das Multiplikative Schwarz Verfahren jedoch nicht sehr effizient.

7 Nachbemerkung

Die Visualisierung des Ergebnisses wurde mit dem IBM DataExplorer gemacht (ein Graphik Software Paket von der Stange).

Die Schwarz Verfahren machen es möglich, auch in komplizierten Geometrien elliptische Differentialgleichungen robust zu lösen.

Im Kernforschungszentrum steht das Paket zur Berechnung elliptischer linearer Differentialgleichungen in komplexen Geometrien als FORTRAN Paket zur Verfügung. Das Knie ist dabei als Beispielprogramm implementiert.

Literatur

- [1] Petter E. Bjørnstad. Multiplicative and additive schwarz methods: Convergence in the two-domain case. In Tony f. Chan, Roland Glowinski, Jacques Périaux, and Olof B. Widlund, editors, *Domain Decomposition Methods*, pages 147-159, Philadelphia, 1989. SIAM.
- [2] Maksimilian Dryja and Olof B. Widlund. An additive variant of the schwarz alternating method for the case of many subregions. Technical Report 339, Courant Institute, Dept. of Computer Science, 1987.

- [3] Pierre Louis Lions. On the schwarz alternating method. i. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–43. Philadelphia. 1988. SIAM.
- [4] Christoph Pospiech. Comparison of schwarz methods for two subdomains. Technical Report 75.91.15, IBM Scientific Center, Heidelberg, to appear.
- [5] Hermann A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. Springer, Berlin/Heidelberg/New York, 1890.

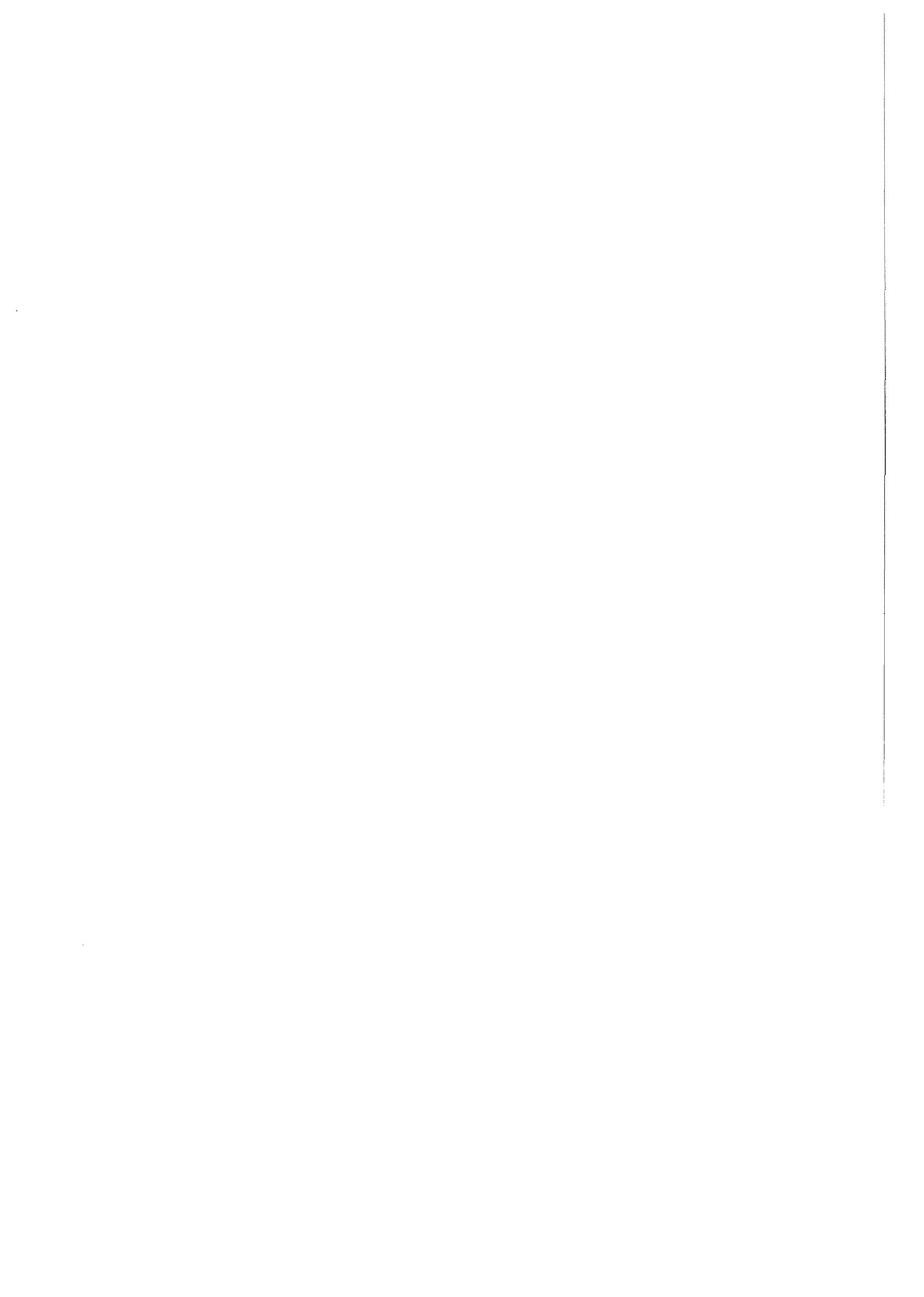


Prof. Dr. Herbert Bauer

Zum Vortrag und zur Person

Wenn man sich die Bücherflut im Bereich der Computeralgebren vergegenwärtigt, wird deutlich, welchen Stellenwert Computeralgebrasysteme in den Anwendungen besitzen. Mathematica, MatLab, Mathcad und Maple sind wohl die aktuellsten Produkte auf diesem Markt. Komplizierte Rechnungen können in symbolischer Form sehr einfach durchgeführt werden. Die Numerik kann man spezifizieren, indem man z.B. die Anzahl der Rechenstellen vorgibt. Nicht zu vergessen sind dabei die graphischen Möglichkeiten dieser Tools. Die Kombination von symbolischer und numerischer Rechnung gepaart mit den graphischen Darstellungsmöglichkeiten - auch dreidimensional - machen die Computeralgebrasysteme nicht nur attraktiv für Mathematikvorlesungen, sondern auch für andere Disziplinen, in denen mathematische Rechnungen durchgeführt werden.

Prof. Bauer studierte Diplom-Mathematik mit Nebenfach Physik an der Technischen Hochschule in Stuttgart, wo er auch an der Fakultät für Mathematik im Institut von Prof. Walcher promovierte. Von 1980 bis 1984 war er wissenschaftlicher Mitarbeiter im Rechenzentrum der DLR in Oberpfaffenhofen. Seit 1984 ist er Professor für Mathematik an der Fachhochschule für Technik und Wirtschaft in Reutlingen im Fachbereich Grundlagen.



Computeralgebra und Ingenieurmathematik - Beispiele mit Maple -

Herbert Bauer

Fachbereich Grundlagen, Fachhochschule für Technik und Wirtschaft Reutlingen

Zusammenfassung

In diesem Vortrag soll exemplarisch aufgezeigt werden, wie sich moderne Computeralgebrasysteme als Werkzeug zur Lösung von Ingenieurproblemen und als Hilfsmittel des Mathematik-Dozenten einsetzen lassen. Dazu werden ein Problem aus einer Konstruktionsabteilung und einige Beispiele aus dem Numerikteil einer Mathematik-Vorlesung für Ingenieure mit Hilfe des Computeralgebrasystems Maple V behandelt. Als besonders vorteilhaft erweist sich dabei die Vereinigung symbolischer, numerischer und graphischer Fähigkeiten in derartigen interaktiven Programmen. Dies kann die bereits heute zu hörende Meinung verständlich machen, daß in Zukunft die Computeralgebra als Softwarepaket in keinem Computer eines Ingenieurbüros mehr fehlen sollte.

Computeralgebra und Ingenieurmathematik

In dem Report Computeralgebra in Deutschland [1] findet man folgenden Versuch einer Begriffsbestimmung:

Die *Computeralgebra* ist ein Wissenschaftsgebiet, das sich mit Methoden zum Lösen mathematisch formulierter Probleme durch symbolische Algorithmen und deren Umsetzung in Soft- und Hardware beschäftigt. Sie beruht auf der exakten endlichen Darstellung endlicher oder unendlicher mathematischer Objekte und Strukturen und ermöglicht deren symbolische und formelmäßige Behandlung durch eine Maschine.

Um eine konkrete Vorstellung zu haben, was damit gemeint sein könnte, sollte man beispielsweise an das symbolische Lösen von Gleichungssystemen (Formelmanipulationen) oder das Rechnen mit Buchstaben oder rationalen Zahlen (Bruchrechnen) denken.

Für die Ingenieurmathematik bedeutet dies, daß die Computeralgebra den Teil der Mathematik betrifft, der bisher mit Papier und Bleistift erledigt wurde. Computeralgebra kann also eine Hilfe für die mathematische Formulierung bzw. die Modellierung des Problems sein. Die Modelle selbst sind in der Regel so kompliziert, daß sie prinzipiell nur numerisch gelöst werden können; somit erhält man nur Ergebnisse für spezielle Werte der Modellparameter. Gelegentlich ist aber auch nur der Aufwand zu groß, die Modelle von Hand für beliebige Werte der Modellparameter symbolisch zu lösen; dann kann ein Computeralgebrasystem diese Arbeit übernehmen und eine symbolische Lösung liefern, die häufig bessere Einsichten in die Zusammenhänge vermittelt als eine rein numerische Lösung.

Seit Ende der sechziger Jahre wurden verschiedene Computeralgebrasysteme für den Batch-Betrieb entwickelt, die meist für spezielle Anwendungsgebiete konzipiert waren, wie z.B. Hochenergiephysik, Zahlentheorie, algebraische Geometrie oder spezielle Gebiete der Algebra. Diese Systeme waren für den Ingenieur weniger geeignet, da sie nur einen kleinen Teil seiner mathematischen Tätigkeit unterstützten und umständlich zu handhaben waren.

In den achtziger Jahren wurde dann eine neue Generation von interaktiven Allzweck-Computeralgebrasystemen, wie z.B. Mathematica und Maple, entwickelt, die symbolische, numerische und graphische Fähigkeiten in sich vereinigen. Da diese Systeme nicht nur das Entwickeln eines mathematischen Modells für das Ingenieurproblem durch Formelmanipulation unterstützen, sondern darüber hinaus auch die numerische Berechnung von Lösungen und die Visualisierung der Ergebnisse durch zwei- und dreidimensionale Graphiken erlauben, versprechen diese Systeme zu einem wertvollen Hilfsmittel bei der täglichen Arbeit des Ingenieurs zu werden - ähnlich wie das früher bei Rechenschiebern und Taschenrechnern der Fall war.

Um einen Einblick in die Fähigkeiten solcher Systeme zu vermitteln und eine erste Einführung in ihren Gebrauch zu geben, sollen einige mit Maple V erstellte Worksheets (Arbeitsblätter) präsentiert werden. Nebenbei soll auch die komfortable Online-Hilfe gezeigt werden.

Beispiele mit Maple V

A. Technische Anwendungen

K-Hebel.ms optimale Dimensionierung eines Knickhebels einer
Wirbelstromrotierensonde zur Materialprüfung von Metallrohren
(symbolische Mathematik, Numerik, Graphik)

Weitere Beispiele im Verzeichnis MAPLEV3\SHARE\ENGINEER, z.B.

shear.ms Entwurf von Maschinenelementen (symbolische Math., 3d-Graphik)
stepresp.ms Antwort eines Schwingkreises auf Einschalten (analytische Lösung eines
Anfangswertproblems, 3d-Graphik)
robotarm.ms Pfadplanung für einen Roboterarm, Animation

B. Demos zur Numerik-Vorlesung für Ingenieure

1. Akkumulierte Rundungsfehler der elementaren arithmetischen Operationen (+, -, *, /) (Methode: Vergleich exakter Rechnung mit Gleitpunktrechnung mit benutzerdef. Stellenzahl)

MMA1_4.ms ¹⁾ Arithmetischer Ausdruck (Auslöschung führender Stellen) [5]
MMA1_5.ms ¹⁾ Nullstellen eines Polynoms (" " ") [4]
Hilbert.ms Lineares Gleichungssystem mit Hilbert-Matrix
SWA1_16.ms ²⁾ Regressionsfunktion $y = a + bx + cx^2$
(schlecht konditioniertes lineares Gleichungssystem)
LinReg.ms Lineare Regression (Größe der Rundungsfehler hängt von verwendeten
Formeln ab) [5]

2. Verfahrensfehler

MMA3_2.ms ¹⁾ Romberg-Verfahren u. Trapez-Regel (Euler-MacLaurinsche Summenformel)
Romberg2.ms Asymptotische Entwicklung der Spalten des Romberg-Verfahrens.
Romberg.ms Prozeduren für diese Worksheets
Diese Prozeduren sind gespeichert in Romberg.m

MMA3_4.ms ¹⁾ Euler-Verfahren (Graph. Veranschaulichung der Verfahrens-Ordnung 1)
MMA3_6.ms ¹⁾ Runge-Kutta-Verfahren (Graph. Veranschaulichung der Verf.-Ordnung 4)
ShareODE.ms Share Library file plot/ODE zugänglich machen [9, 10]
Dieses file ist gespeichert in ODE.m

C. Grenzen und Mängel

Hilb_Eig.ms Eigenwertproblem mit Hilbert-Matrix [10]
n = 4: exakte Rechnung mit algebraischen Zahlen zu aufwendig
n = 5: exakte Lösung nicht möglich;
Genauigkeit der numerischen Ergebnisse ist nicht gesichert
NLFZ_EX.ms Nullstellen einer transzendenten Funktion [3]; exakte Lösung nicht möglich;
Genauigkeit und Eindeutigkeit der numerischen Ergebnisse nicht gesichert
B11_Rump.ms Hat das Polynom positive Nullstellen? (Ein Beispiel von Rump [5])
Nachweis mit Maple aufwendig, Fehler in is(..., real) in Release 3
output.ms im Verzeichnis MAPLEV3\EXAMPLES
Erzeugen von FORTRAN- und C-Code. Numerische Rechnungen mit
Interpreter sind langsamer als compilierte Programme

¹⁾ vgl. Übungsaufgaben zu Mathematische Methoden in der Automatisierung

²⁾ vgl. Übungsaufgabe zu Statistik und Wahrscheinlichkeitsrechnung

Literatur

1. Computeralgebra in Deutschland - Bestandsaufnahme, Möglichkeiten, Perspektiven. Hrsg. Fachgruppe Computeralgebra de GI, DMV, GAMM, Passau und Heidelberg 1993
2. Geddes, K.O., Czapor, S., Labahn, G., Algorithms for Computer Algebra (Kluwer Academic Publishers) ISBN 0-7923-9259-0 (1992)
3. Hammer, R./Hocks, M./Kulisch, U./Ratz, D.: Numerical Toolbox for verified Computing I, Basic Numerical Problems (Springer-Verlag) ISBN 3-540-57118-3 (1993)
4. Kulisch, U. (Hrsg.): PASCAL-SC Manual and System-Disks (Teubner-Wiley) (1987)
5. Rump, S.M.: Wie zuverlässig sind die Ergebnisse unserer Rechenanlagen? in Jahrbuch Überblicke Mathematik 1983 (Bibliographisches Institut Mannheim), S. 163 - 168

Handbücher für Maple V

6. Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B., Watt, S.M., First Leaves: A Tutorial Introduction to Maple V (Springer-Verlag) ISBN 0-387-97621-3 (1992)
7. Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B., Watt, S.M., Maple V Library Reference Manual (Springer-Verlag) ISBN 0-387-97592-6 (1991)
8. Char, B.W., Geddes, K.O., Gonnet, G.H., Leong, B.L., Monagan, M.B., Watt, S.M., Maple V Language Reference Manual (Springer-Verlag) ISBN 0-387-97622-1 (1991)
9. Ellis, W., Johnson, E., Lodi, E., Schwalbe, D., Maple V Flight Manual (compatible with Release 2) (Brooks/Cole Publishing Co.) 0-534-21235-2 (1993)
10. Ellis, W., Johnson, E., Lodi, E., Schwalbe, D., Maple V in der mathematischen Anwendung, Flight Manual deutsche Ausgabe (International Thomson Publishing) 3-929821-21-4 (1994)
11. Redfern, D., The Maple Handbook (Springer-Verlag) ISBN 0-387-94054-5 (1993)
12. Spieth, A., Maple V Release 2 Referenzhandbuch (International Thomson Publishing) 3-929821-73-7 (1994)

Beschreibungen der Benutzer-Oberflächen

13. Maple V, Rel. 3 Motif Interface, Getting Started. Waterloo Maple Software 1994.
14. MAPLE V, Rel. 3 for DOS, Getting Started. Waterloo Maple Software 1994.
15. MAPLE V, Rel. 3 for the Macintosh, Getting Started. Waterloo Maple Software 1994.

Lehrbücher zu Maple V (Auswahl)

16. Burkhardt, W., Erste Schritte mit Maple (Springer-Verlag) ISBN 3-540-56649-X (1994)
17. Devitt, J.S., Calculus with Maple V (Brooks/Cole Publishing Co.) ISBN 0-534-16362-9 (1993)
18. Gander, W., Hrebicek, J., Solving Problems in Scientific Computing Using Maple and MATLAB (Springer-Verlag) ISBN 0-387-57329-1 (1993)
19. Gloggenießer, H., Maple V Software für Mathematiker (Markt&Technik) ISBN 3-87791-439-X (1993)
20. Heck, A., Introduction to Maple - A Computer Algebra System (Springer-Verlag) ISBN 0-387-97662-0 (1993)
21. Johnson, E., Linear Algebra with Maple V (Brooks/Cole Publishing Co.) ISBN 0-534-13069-0 (1993)
22. Kofler, M., Maple V Release 3 (Addison-Wesley Verlag Deutschland) ISBN 3-89319-765-6 (1994)
23. Kreyszig, E., Normington, E.J., Maple Computer Manual for Advanced Engineering Mathematics (John Wiley & Sons, Inc.) ISBN 0-471-31126-X (1994)
24. Vetsch, M., Die Sprache Maple. Probleme, Beispiele, Lösungen. (International Thomson Publishing) 3-929821-22-2 (1994)
25. Werner, W., Mathematik lernen mit Maple V (ELBI-Verlag GmbH) ISBN 3-929694-03-4 (1993)
26. Werner, W., Mathematikaufgaben lösen mit Maple V (ELBI-Verlag GmbH) ISBN 3-929694-04-2 (1994)

1.4. Berechnen Sie mit Ihrem Taschenrechner den Ausdruck

$$b = 9x^4 - y^4 + 2y^2 \quad \text{für } x = 10\,864.0, \quad y = 18\,817.0$$

und vergleichen Sie mit dem richtigen Ergebnis ($b = 1.0$).

1.5. Berechnen Sie die Funktionswerte des Polynoms

$$y = 2030x^4 - 5741x^3 - x^2 + 11482x - 8118$$

für $1.4140 \leq x \leq 1.4145$ (Schrittweite z.B. 0.000 002 5).
Vergleichen Sie das Ergebnis mit der Tatsache, daß ein Polynom 4. Grades höchstens 4 Nullstellen haben kann und daß daher höchstens 4 Vorzeichenwechsel auftreten können!

3.2. Die x- bzw. y-Auslenkung einer Kathodenstrahlröhre werde so durch zwei Wechselspannungen gesteuert, daß

$$x = 2 \cos t, \quad y = 2 \sin 2t \quad (\text{Einheiten: cm})$$

gilt. Berechnen Sie die Bogenlänge der entstehenden Lissajous-Figur (Skizze!)

- a) mit Hilfe des Romberg-Verfahrens
- b) mit Hilfe der Sehnen-Trapez-Regel auf 10^{-3} cm genau.

Vergleichen Sie die Ergebnisse!

Hinweis: Berechnen Sie die Bogenlänge des im 1. Quadranten gelegenen Teils der Figur auf $2,5 \cdot 10^{-4}$ cm genau.

3.4. Lösen Sie mit Hilfe des Euler-Verfahrens das Anfangswertproblem

$$y' = y - \frac{2x}{y}, \quad y(0) = 1.$$

im Intervall $0 \leq x \leq 1$

- a) mit Schrittweite $h = 0.2$,
- b) mit Schrittweite $h = 0.02$ (Werte nur für 0.2, 0.4, ... ausgeben).
- c) Um welchen Faktor ist der globale Diskretisierungsfehler an den Stellen $x = 0.2$ und $x = 1.0$ in b) kleiner als in a)?
(Hinweis: Die exakte Lösung des AWP ist $y = 2x + 1$)

3.6. Lösen Sie das Anfangswertproblem aus Aufgabe 3.4 im Intervall $0 \leq x \leq 1$ mit Hilfe des Runge-Kutta-Verfahrens 4. Ordnung mit Schrittweite $h = 0.2$. Vergleichen Sie den globalen Diskretisierungsfehler mit demjenigen in Aufgabe 3.4.

Eine Übungsaufgabe zu Statistik und Wahrscheinlichkeitsrechnung für WI

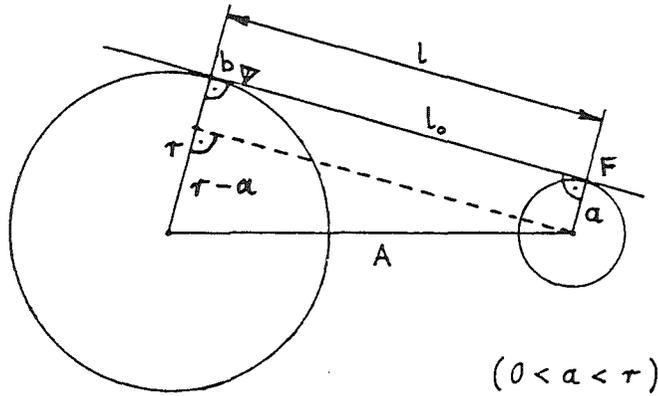
1.16. Bei einem physikalischen Versuch werden für einen frei fallenden Körper die Zeit t (in sec.) und die zurückgelegte Strecke s (in m) gemessen. Man erhält folgende Beobachtungswerte:

t	0,14	0,20	0,29	0,35	0,40	0,45	0,50	0,55	0,64	0,70
s	0,1	0,2	0,4	0,6	0,8	1,0	1,2	1,5	2,0	2,4

Es soll die KQ-Regressionsfunktion $\hat{s} = a + bt + ct^2$ bestimmt werden. (Diese Funktion entspricht den Fallgesetzen, falls kein Luftwiderstand vorhanden ist.)

Bsp 1 Wirbelstromrotiersonde (vereinfachtes Modell)

b) Knickhebel



$(0 < a < r)$

Variablen

- A = Abstand Mittelpunkt Rohr bis Drehpunkt des Hebels ($A > 0$)
- r = Außenradius des Rohres ($0 < r < A$)
- b = Abweichung der Sonde vom Berührungspunkt (b reell)
- l = Abstand Berührungspunkt des Hebels bis Lotfußpunkt F ($l > 0$)
- a = Länge des Lots vom Drehpunkt auf die Tangente ($r - A < a < r + A$)
- l_0 = Abstand Sonde bis Lotfußpunkt F ($l_0 > 0$)

Formeln

$l = l(r) = \sqrt{A^2 - (r - a)^2}$, $b = l - l_0$

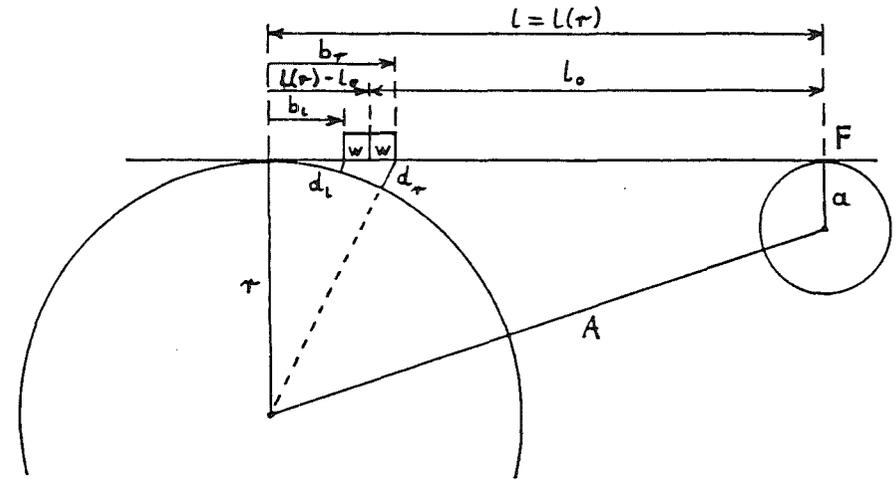
Optimale Abmessungen, so daß $\max |b| \rightarrow \text{Min.}$
 $(0 < r_1 < r_2 \leq A)$ $r_1 \leq r \leq r_2$

$a = \frac{1}{2}(r_1 + r_2)$, $l = \frac{1}{2}(A + \sqrt{A^2 - (r_2 - a)^2})$

Bsp 2a) Wirbelstromrotiersonde (realistisches Modell)

Differenz von rechtem und linkem Abstand zum Rohr möglichst klein. Exakte Modellgleichungen.

$\Delta = d_r - d_l$, $|\Delta| \rightarrow \text{Min.}$



Variablen

- A } s. Bsp. 1b)
- r }
- l }
- a }

- l_0 = Abstand Sondenmitte bis Lotfußpunkt
- w = Weite des Sensors
- b_r = rechte Abweichung = $(l(r) - l_0) + w$
- b_l = linke Abweichung = $(l(r) - l_0) - w$
- d_r = rechter Abstand = $\sqrt{r^2 + b_r^2} - r$
- d_l = linker Abstand = $\sqrt{r^2 + b_l^2} - r$
- Δ = Unterschied der Abstände = $d_r - d_l$

Optimale Dimensionierung des Knickhebels einer Wirbelstromrotiersonde zur Materialprüfung von Metallrohren

Die Qualität der Abtastung ist um so besser, je kleiner die Differenz zwischen den Abständen des linken bzw. rechten Endes des Sensors vom zu prüfenden Rohr ist. Die Fehlerfunktion Delta liefert diese Differenz in Abhängigkeit von den Variablen:

- r (Rohrradius),
- a, l0 (Abmessungen des Knickhebels)

sowie den Parametern:

A (Abstand Mittelpunkt Rohr bis Drehpunkt Hebel), w (Weite des Sensors)

$$\Delta := (r, a, l_0) \rightarrow \sqrt{r^2 + (A^2 - (r-a)^2)^{1/2} - l_0 + w} - \sqrt{r^2 + (A^2 - (r-a)^2)^{1/2} - l_0 - w}$$

$$\Delta := (r, a, l_0) \rightarrow \sqrt{r^2 + \left(\sqrt{A^2 - (r-a)^2} - l_0 + w\right)^2} - \sqrt{r^2 + \left(\sqrt{A^2 - (r-a)^2} - l_0 - w\right)^2}$$

Optimierungs-Aufgabe:

Das Maximum des Betrags der Fehlerfunktion Delta soll fuer den Anwendungsbereich $r_1 \leq r \leq r_2$ minimiert werden.

Zu gegebenem A, r1, r2 und w sollen jeweils die optimalen Werte fuer a und l0 sowie der max. Betrag des Fehlers del bestimmt werden.

Ein vereinfachtes Modell, in dem anstelle von Delta die Abweichung der Sensormitte vom Beruehrpunkt von Knickhebel und Rohr minimiert wird, besitzt folgende optimale Loesung:

$$a_v := (r_1 + r_2)/2; l_{0_v} := (A + (A^2 - (r_2 - a_v)^2)^{1/2}) / 2;$$

$$a_v := \frac{1}{2} r_1 + \frac{1}{2} r_2$$

$$l_{0_v} := \frac{1}{2} A + \frac{1}{2} \sqrt{A^2 - \left(\frac{1}{2} r_2 - \frac{1}{2} r_1\right)^2}$$

Damit erhaelt man eine erste Naecherung fuer die Loesung der Optimierungs-Aufgabe.

Als Beispiel werden A=100, r1=30, r2=50, w=5/2 vorgegeben (alle Angaben in mm).

$$A := 100; r_1 := 30; r_2 := 50; w := 5/2;$$

$$a_v := a_v; l_{0_v} := \text{evalf}(l_{0_v});$$

$$a_v = 40$$

$$l_{0_v} = 99.74937186$$

Schaubild der Fehlerfunktion Delta fuer a = a_v, l0 = l0_v (Uebersicht):

$$\text{plot}(\Delta(r, a_v, l_{0_v}), r=0..100, -5..1);$$

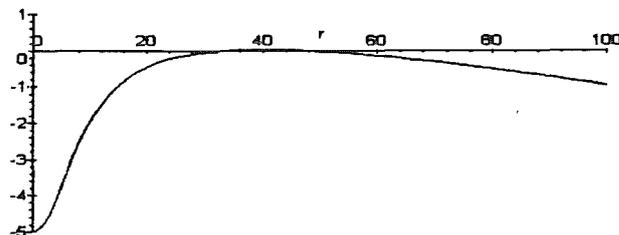
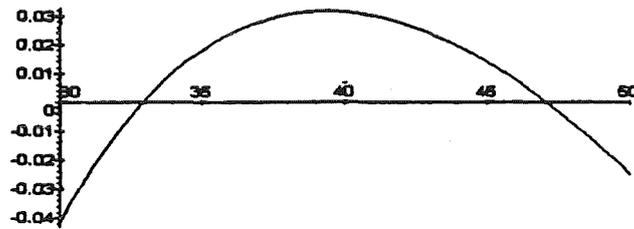


Schaubild der Fehlerfunktion Delta (fuer den Anwendungsbereich):

$$\text{plot}(\Delta(r, a_v, l_{0_v}), r=r_1..r_2);$$



Dem Schaubild entnimmt man den max. Betrag des Fehlers Δ_v

> $\Delta_v := 0.042$;

$$\Delta_v := .042$$

Maximale Werte des Betrags des Fehlers treten am Rand auf fuer $r=r_1$ und $r=r_2$ sowie dazwischen fuer $r=r_3$.

Fuer die optimale Loesung hat die Fehlerfunktion $\Delta(r, \dots)$ an den Stellen $r = r_1, r_3, r_2$ den selben Betrag und alternierende Vorzeichen. An der Stelle r_3 ist die Ableitung $D\Delta$ von Δ nach r gleich 0.

> $D\Delta := D[1](\Delta)$;

$$D\Delta := (r, a, l_0) \rightarrow \frac{1}{2} \frac{2r + \frac{(\sqrt{A^2 - (r-a)^2} - l_0 + w)(-2r + 2a)}{\sqrt{A^2 - (r-a)^2}}}{\sqrt{r^2 + (\sqrt{A^2 - (r-a)^2} - l_0 + w)^2}} - \frac{1}{2} \frac{2r + \frac{(\sqrt{A^2 - (r-a)^2} - l_0 - w)(-2r + 2a)}{\sqrt{A^2 - (r-a)^2}}}{\sqrt{r^2 + (\sqrt{A^2 - (r-a)^2} - l_0 - w)^2}}$$

Damit erhaelt man 3 Gleichungen fuer die 3 Unbekannten a, l_0, r_3 :

> $gl_1 := \Delta(r_1, a, l_0) + \Delta(r_3, a, l_0) = 0$;

> $gl_2 := \Delta(r_2, a, l_0) + \Delta(r_3, a, l_0) = 0$;

> $gl_3 := D\Delta(r_3, a, l_0) = 0$;

$$gl_1 := \sqrt{900 + \left(\sqrt{10000 - (30-a)^2} - l_0 + \frac{5}{2}\right)^2} - \sqrt{900 + \left(\sqrt{10000 - (30-a)^2} - l_0 - \frac{5}{2}\right)^2} + \sqrt{r_3^2 + \left(\sqrt{10000 - (r_3-a)^2} - l_0 + \frac{5}{2}\right)^2} - \sqrt{r_3^2 + \left(\sqrt{10000 - (r_3-a)^2} - l_0 - \frac{5}{2}\right)^2} = 0$$

$$gl_2 := \sqrt{2500 + \left(\sqrt{10000 - (50-a)^2} - l_0 + \frac{5}{2}\right)^2} - \sqrt{2500 + \left(\sqrt{10000 - (50-a)^2} - l_0 - \frac{5}{2}\right)^2} + \sqrt{r_3^2 + \left(\sqrt{10000 - (r_3-a)^2} - l_0 + \frac{5}{2}\right)^2} - \sqrt{r_3^2 + \left(\sqrt{10000 - (r_3-a)^2} - l_0 - \frac{5}{2}\right)^2} = 0$$

$$gl_3 := \frac{1}{2} \frac{2r_3 + \frac{(\%1 - l_0 + \frac{5}{2})(-2r_3 + 2a)}{\sqrt{10000 - (r_3-a)^2}}}{\sqrt{r_3^2 + \left(\%1 - l_0 + \frac{5}{2}\right)^2}} - \frac{1}{2} \frac{2r_3 + \frac{(\%1 - l_0 - \frac{5}{2})(-2r_3 + 2a)}{\sqrt{10000 - (r_3-a)^2}}}{\sqrt{r_3^2 + \left(\%1 - l_0 - \frac{5}{2}\right)^2}} = 0$$

$$\%1 := \sqrt{10000 - (r_3 - a)^2}$$

Exakte Loesung des nichtlinearen Gleichungssystems (gelingt nicht):

```
> # Lsg := solve({g1, g2, g3}, {a, l0, r3});
```

Lsg :=

Numerische Loesung des nichtlinearen Gleichungssystems:

```
> Lsg := fsolve({g1, g2, g3}, {a, l0, r3}, {a=0.9*a_v..a_v, l0=0.9*A..A, r3=r1..r2});
```

Lsg := {l0 = 99.75138323, a = 39.36918315, r3 = 38.73518823 }

Maximaler Betrag des Fehlers

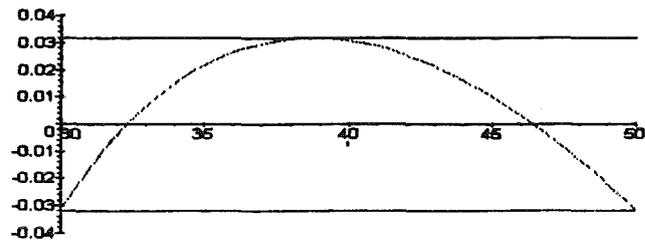
```
> del := subs( Lsg, Delta(r3, a, l0));
```

del := .03176570

Die Korrektheit dieses Resultats zeigt folgende Graphik:

```
> Delta0 := subs( Lsg, Delta(r, a, l0));
```

```
> plot( {Delta0, -del, del}, r=r1..r2, -0.04..0.04 );
```



```
> restart;
```

```
>
```

Die Berechnung eines arithmetischen Ausdrucks mit gewöhnlicher Gleitpunktarithmetik kann zu ungenauen oder völlig falschen Ergebnissen führen (Quelle: Rump, S.164, 2.).

> b := 9*x^4 - y^4 + 2*y^2;

$$b := 9x^4 - y^4 + 2y^2$$

Berechnung des Ausdrucks für $x = 10864$, $y = 18817$ mit exakter ganzzahliger Arithmetik:

> subs({x=10864, y=18817}, b);

1

Berechnung desselben Ausdrucks mit Gleitpunktarithmetik:

> Digits := 10:

> subs({x=10864.0, y=18817.0}, b);

$$.81589780 \cdot 10^7$$

> convert(" ", rational);

8158978

Das ungenaue Ergebnis beim Rechnen mit Gleitpunktarithmetik mit 10-stelliger Mantisse wird durch die Auslöschung führender Stellen bei der Subtraktion gerundeter Zwischenergebnisse verursacht:

> term1 := subs({x=10864, y=18817}, 9*x^4);

> term2 := subs({x=10864, y=18817}, - y^4);

$$term1 := 125372283822342144$$

$$term2 := -125372284530501121$$

> tzwi := term1 + term2;

> term3 := subs({x=10864, y=18817}, 2*y^2);

$$tzwi := -708158977$$

$$term3 := 708158978$$

> tzwi_fpa := subs({x=10864.0, y=18817.0}, 9*x^4 - y^4);

> term3_fpa := subs({x=10864.0, y=18817.0}, 2*y^2);

$$tzwi_fpa := -.7 \cdot 10^9$$

$$term3_fpa := .7081589780 \cdot 10^9$$

Abhängigkeit des Ergebnisses von der verwendeten Stellenzahl:

> Digits;

10

> Digits := 15:

> subs({x=10864.0, y=18817.0}, b);

-22.00

> Digits := 17:

> subs({x=10864.0, y=18817.0}, b);

-2.00

> Digits := 18:

> subs({x=10864.0, y=18817.0}, b);

1.00

Literatur

Rump, S.M.: Wie zuverlässig sind die Ergebnisse unserer Rechenanlagen? in Jahrbuch Überblicke Mathematik 1983, S.163 - 168, Bibliographisches Institut, Mannheim 1983.

Um Nullstellen eines Polynoms p zu berechnen, muß man in der Lage sein, Funktionswerte des Polynoms zu berechnen. Verwendet man Gleitpunktarithmetik mit endlicher Mantissenlänge, so sind evtl. nicht einmal die Vorzeichen korrekt.

Horner-Schema eines Polynoms p (Quelle: PASCAL-SC Manual and System Disks):

```
> p := (((2030*x - 5741)*x - 1)*x + 11482)*x - 8118;
      p := (((2030 x - 5741) x - 1) x + 11482) x - 8118
```

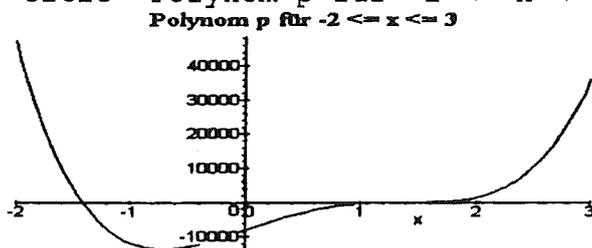
Dasselbe Polynom p entwickelt nach Potenzen von x :

```
> expand(p);
```

$$2030 x^4 - 5741 x^3 - x^2 + 11482 x - 8118$$

Funktionsverlauf und Nullstellen mit exakter rationale Arithmetik:

```
> s := solve( p=0, x ); evalf({s});
> plot(p, x=-2..3, title='Polynom p für -2 <= x <= 3');
```

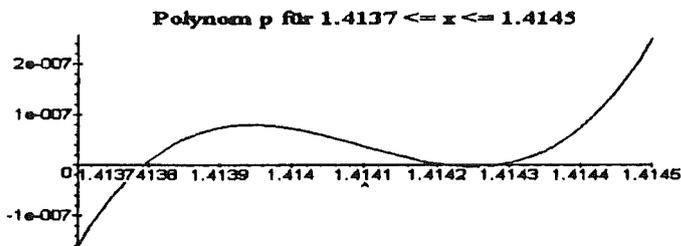


$$s := \frac{99}{70}, \frac{41}{29}, \sqrt{2}, -\sqrt{2}$$

{1.414285714, 1.413793103, 1.414213562, -1.414213562}

```
> Digits := 12;
```

```
> plot( p, x=1.4137..1.4145, title='Polynom p für 1.4137 <= x <= 1.4145');
> 145');
```



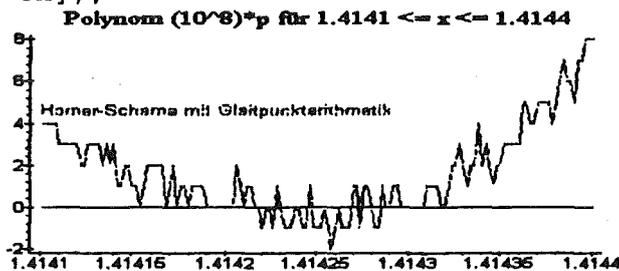
Liste der Funktionswerte von p gemäß Horner-Schema für Gleitpunktarithmetik mit 12-stelliger Mantisse (Schrittweite 0.000 002):

```
> Digits;
```

12

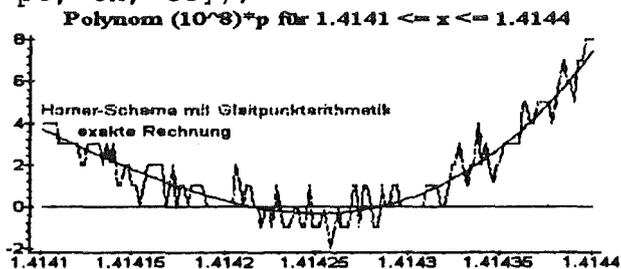
```
> lh := [[1.4141 + n*0.000002,
> subs( x=1.4141 + n*0.000002, (10^8)*p )] $n=0..150];
> ph := plot( lh, style=LINE, color=RED, title='Polynom (10^8)*p für
> 1.4141 <= x <= 1.4144', axes=FRAME );
> lN := [[1.4141, 0], [1.4144, 0]];
> pN := plot( lN, style=LINE, color=BLACK );
> with(plots):
> th := textplot([1.4141, 4.2, 'Horner-Schema mit Gleitpunktarithmet
> ik'], color=RED, align={ABOVE,RIGHT});
```

```
> display([ph, pN, th]);
```



Liste der exakten Funktionswerte (Schrittweite 0.000 002) und
Grafik für Werte nach Horner mit Gleitpunktarithmetik bzw. für exakte Werte:

```
> le := [[14141/10000 + n/500000,  
> subs( x=14141/10000 + n/500000, (10^8)*p)] $n=0..150]:pe := plot(  
> le , style=LINE, color=BLUE ):  
> te := textplot([1.41412, 3.2, `exakte Rechnung`], color=BLUE, alig  
> n={ABOVE,RIGHT}):  
> display([ph, pN, pe, th, te]);
```



Literatur

Kulisch, U. (Hrsg.): PASCAL-SC Manual and System-Disks, Teubner-Wiley 1987

Die Lösung eines linearen Gleichungssystems $Ax=b$ kann sehr ungenau werden, wenn die Koeffizientenmatrix A schlecht konditioniert ist, und man mit endlicher Stellenzahl rechnet (Gleitpunktarithmetik).

```
> with (linalg):
Warning: new definition for norm
Warning: new definition for trace
```

n = Anzahl der Gleichungen bzw. Unbekannten:

```
> n := 8;
n := 8
```

Koeffizientenmatrix A: Hilbert-Matrix

```
> A := hilbert(n);
A :=
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} \\ \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} \\ \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} \end{bmatrix}$$

Konditionszahl:

```
> cond(A);
33872791095
```

Rechte Seite b: 1, 2, ..., n

```
> b := randvector(n);
b := [-85 -55 -37 -35 97 50 79 56]
```

x = Exakte Lösung des Gleichungssystems

```
> x := linsolve(A,b);
```

Digits = Anzahl der Stellen der Mantisse von Gleitpunktzahlen:

```
> Digits := 10;
> Af := map(convert, A, float);
> bf := map(convert, b, float);
Digits := 10
bf := [-85. -55. -37. -35. 97. 50. 79. 56.]
```

xf = Lösung des Gleichungssystems mit Gleitpunktarithmetik

```
> xf := linsolve(Af,bf);
```

Vergleich von	x,	xf,	x-xf
> augment(map(convert, x, float), xf, add(x,xf,1,-1));			
	.21990128 10 ⁸	.2211482000 10 ⁸	-124692.00
	-.1158700536 10 ¹⁰	-.1167512907 10 ¹⁰	.8812371 10 ⁷
	.1494886176 10 ¹¹	.1508004102 10 ¹¹	-.13117926 10 ⁹
	-.8018131044 10 ¹¹	-.8094887900 10 ¹¹	.76756856 10 ⁹
	.2143977066 10 ¹²	.2165755523 10 ¹²	-.21778457 10 ¹⁰
	-.3017325271 10 ¹²	-.3049334560 10 ¹²	.32009289 10 ¹⁰
	.2137939724 10 ¹²	.2161388091 10 ¹²	-.23448367 10 ¹⁰
	-.6010439292 10 ¹¹	-.6078127665 10 ¹¹	.67688373 10 ⁹
>			

Die Bestimmung einer KQ-Regressionsfunktion $s = a + b t + c t^2$
 zu gegebenen Beobachtungspaaren $(t[i], s[i])$ führt auf ein System von Normalgleichungen
 $A * x = d$ für die gesuchten Koeffizienten $x = (a, b, c)$

```
> with (linalg):
> A := array(symmetric, 1..3, 1..3, [[N], [sum(t[i], i=1..N), sum(t[i]^2, i=1..N)], [sum(t[i]^2, i=1..N), sum(t[i]^3, i=1..N), sum(t[i]^4, i=1..N)]]);
> d := vector(3, [sum(s[i], i=1..N), sum(t[i]*s[i], i=1..N), sum(t[i]^2*s[i], i=1..N)]);
Warning: new definition for norm
Warning: new definition for trace
```

$$A := \begin{bmatrix} N & \sum_{i=1}^N t_i & \sum_{i=1}^N t_i^2 \\ \sum_{i=1}^N t_i & \sum_{i=1}^N t_i^2 & \sum_{i=1}^N t_i^3 \\ \sum_{i=1}^N t_i^2 & \sum_{i=1}^N t_i^3 & \sum_{i=1}^N t_i^4 \end{bmatrix}$$

$$d := \begin{bmatrix} \sum_{i=1}^N s_i & \sum_{i=1}^N t_i s_i & \sum_{i=1}^N t_i^2 s_i \end{bmatrix}$$

```
t-Werte:
> t := [0.14, 0.20, 0.29, 0.35, 0.40, 0.45, 0.50, 0.55, 0.64, 0.70];
t := [.14, .20, .29, .35, .40, .45, .50, .55, .64, .70]
```

```
s-Werte:
> s := [0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.5, 2.0, 2.4];
s := [.1, .2, .4, .6, .8, 1.0, 1.2, 1.5, 2.0, 2.4]
```

```
> N := nops(t);
N := 10
```

Koeffizientenmatrix A (mit rationaler Darstellung Ae):

```
> A := map(value, A);
> Ae := map(convert, A, rational, exact);
```

$$A := \begin{bmatrix} 10 & 4.22 & 2.0808 \\ 4.22 & 2.0808 & 1.129652 \\ 2.0808 & 1.129652 & .65254788 \end{bmatrix}$$

$$Ae := \begin{bmatrix} 10 & \frac{211}{50} & \frac{2601}{1250} \\ \frac{211}{50} & \frac{2601}{1250} & \frac{282413}{250000} \\ \frac{2601}{1250} & \frac{282413}{250000} & \frac{16313697}{25000000} \end{bmatrix}$$

Rechte Seite b (mit rationaler Darstellung be):

> d := map(value, d);

> de := map(convert, d, rational, exact);

$d := [10.2 \quad 5.535 \quad 3.19655]$

$de := \left[\frac{51}{5} \quad \frac{1107}{200} \quad \frac{63931}{20000} \right]$

x = (a, b, c) = Exakte Lösung des Gleichungssystems

xe = Gleitpunktapproximation der exakten Lösung

> x := linsolve(Ae, de); xe := map(convert, "float);

$x := \left[\frac{28567}{17000695} \quad \frac{147020}{64602641} \quad \frac{315859750}{64602641} \right]$

$xe := [.001680343068 \quad .002275758355 \quad 4.889269929]$

xf = Lösung des Gleichungssystems mit Gleitpunktarithmetik

Digits = Anzahl der Stellen der Mantisse von Gleitpunktzahlen

> Digits := 7;

> xf := linsolve(A, d);

$xf := [.001695 \quad .002195 \quad 4.889364]$

> Digits := 6;

> xf := linsolve(A, d);

$xf := [.00148 \quad .00334 \quad 4.88805]$

> Digits := 5;

> xf := linsolve(A, d);

$xf := [.0054 \quad -.0180 \quad 4.9126]$

Rechnet man mit weniger als 6 Stellen, so wird das Ergebnis sehr ungenau. Ursache?

> Digits := 10: cond(A);

3754.926236

Das lineare Gleichungssystem ist schlecht konditioniert, d.h. kleine relative Fehler der Koeffizienten des Gleichungssystems bewirken große relative Fehler des Ergebnisses. Rundungsfehler haben dieselbe Auswirkung!

Bezogen auf den größten Koeffizienten sind die Fehler allerdings selbst bei 5-stelliger Rechnung tragbar.

>

Beispiel 10 aus

Rump, S.M.: Wie zuverlässig sind die Ergebnisse unserer Rechenanlagen? in Jahrbuch Überblicke Mathematik 1983, S.163 - 168, Bibliographisches Institut, Mannheim 1983.

Lineare Regressionsfunktion $y = a + b x$ für drei Punkte auf einer Geraden

```
> X := [5201477, 5201478, 5201479]; Y := [99999, 100000, 100001];
      X := [ 5201477, 5201478, 5201479 ]
      Y := [ 99999, 100000, 100001 ]
```

Exakte Rechnung (Varianz und Kovarianz mit Formeln zur Rechenvereinfachung):

```
> LinRegRF := proc(X, Y) local xbar, ybar, s2x, covxy, n, i, a, b; # Formeln zur Rechenvereinfachung
> n := nops(X);
> xbar := sum( X[i], i=1..n ) / n; # arith. Mittel von X
> ybar := sum( Y[i], i=1..n ) / n; # arith. Mittel von Y
> s2x := sum( X[i]^2, i=1..n ) / n - xbar^2; # Varianz von X
> covxy := sum( X[i]*Y[i], i=1..n ) / n - xbar*ybar; # Kovarianz von X und Y
> b := covxy/s2x; a := ybar - b*xbar; # Regressionskoeffizienten
> a + b*x; # lineare yx-Regressionsfunktion
> end;
```

```
> yDach := LinRegRF(X, Y);
      yDach := -5101478 + x
```

```
> subs( x=5201480, yDach );
      100002
```

Gleitpunktarithmetik (Varianz und Kovarianz mit Formeln zur Rechenvereinfachung):

Digits := 10, 12, 14, 16

```
> Digits := 10;
> Xf := evalf(X); Yf := evalf(Y); yDach := LinRegRF(Xf, Yf);
      Xf := [ .5201477 107, .5201478 107, .5201479 107 ]
      Yf := [ 99999., 100000., 100001. ]
      yDach := -160073.9000 + .050000000000 x
```

```
> subs( x=5201480, yDach );
      100000.1000
```

Gleitpunktarithmetik (Varianz und Kovarianz mit Formeln gemäß Definition):

```
> LinReg := proc(X, Y) local xbar, ybar, s2x, covxy, n, i, a, b; # Formeln gemäß Definition
> n := nops(X);
> xbar := sum( X[i], i=1..n ) / n; # arith. Mittel von X
> ybar := sum( Y[i], i=1..n ) / n; # arith. Mittel von Y
> s2x := sum( (X[i] - xbar)^2, i=1..n ) / n; # Varianz von X
> covxy := sum( (X[i] - xbar)*(Y[i] - ybar), i=1..n ) / n; # Kovarianz von X und Y
> b := covxy/s2x; a := ybar - b*xbar; # Regressionskoeffizienten
> a + b*x; # lineare yx-Regressionsfunktion
> end;
```

Digits := 10, 8, 6

```
> Digits := 10;
> Xf := evalf(X); Yf := evalf(Y); yDach := LinReg(Xf, Yf);
      Xf := [ .5201477 107, .5201478 107, .5201479 107 ]
      Yf := [ 99999., 100000., 100001. ]
      yDach := -.5101470275 107 + .9999985150 x
```

```
> subs( x=5201480, yDach );
      100002.001
```

Gegeben ist die Parameterdarstellung einer Kurve (Lissajous-Figur).

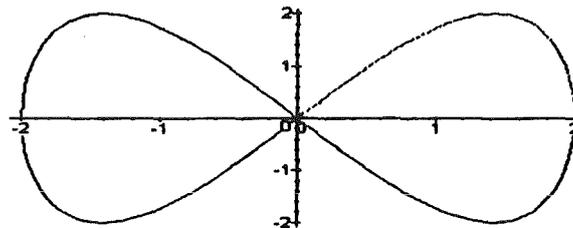
Gesucht ist die Bogenlänge $l/4$ (des im 1. Quadranten gelegenen Teils der Figur).

> $x := 2 \cdot \cos(t)$; $y := 2 \cdot \sin(2 \cdot t)$;

$$x := 2 \cos(t)$$

$$y := 2 \sin(2 t)$$

> p1 := plot([x,y, t=0..Pi/2], color=RED);
 > p2 := plot([x,y, t=Pi/2..2*Pi], color=BLACK);
 > with(plots): display({p1,p2});



Bogenlänge

> $l/4 = \text{Int}(\text{sqrt}(\text{'diff}(x, t)^2 + \text{'diff}(y, t)^2$), $t=0..Pi/2$);

$$\frac{1}{4} l = \int_0^{\frac{1}{2} \pi} \sqrt{\left(\frac{\partial}{\partial t} x\right)^2 + \left(\frac{\partial}{\partial t} y\right)^2} dt$$

Numerische Integration mit dem Romberg-Verfahren

> f := unapply(sqrt(diff(x, t)^2 + diff(y, t)^2), t);
 > a := 0; b := Pi/2;
 > Digits := 8; read 'romberg.m';

$$f := t \rightarrow 2 \sqrt{\sin(t)^2 + 4 \cos(2 t)^2}$$

$$a := 0$$

$$b := \frac{1}{2} \pi$$

$$\text{Digits} := 8$$

> SchemaRom(4);

n	$T(n, 0)$	$T(n, 1)$	$T(n, 2)$	$T(n, 3)$	$T(n, 4)$
0	6.6540003	-----	-----	-----	-----
1	4.4377210	3.6989613	-----	-----	-----
2	4.6962593	4.7824386	4.8546705	-----	-----
3	4.7144586	4.7205250	4.7163974	4.7142026	-----
4	4.7147160	4.7148016	4.7144201	4.7143888	4.7143894

$|T(4,3) - T(4,4)| < 0.000001$ legt nahe, daß $l/4 = 4.7143894$ mit Fehler < 0.000001

Der tatsächliche Fehler ist aber wesentlich größer wie Numerische Integration durch Maple (Clenshaw-Curtis) zeigt

> readlib('evalf/int');
 > IC := 'evalf/int'(f(t), t=a..b, 10);
 > Fehler_Romberg := IC - TRom(4,4);

$$IC := 4.714715654$$

$$\text{Fehler_Romberg} := .0003263$$

Dagegen ist der Fehler für die Sehnen-Trapez-Regel wesentlich kleiner

> Fehler_Sehnen_Trapez := IC - TRom(4,0);

$$\text{Fehler_Sehnen_Trapez} := -3 \cdot 10^{-6}$$

Woran liegt das?

Ist die Funktion $f(t)$ $(2m+2)$ -mal stetig differenzierbar auf $[a,b]$, so besitzt die Sehnen-Trapez-Regel $T(h)$ folgende asymptotische Entwicklung (Euler-MacLaurinsche Summenformel s. Schwarz [1] S.324, (8.11) und (8.12) oder Stoer [2] S.104ff, 3.2)

> $T(h) = \text{Int}(f(t), t='a'..'b') + \text{Sum}(\text{bernoulli}(2*k)/(2*k)! * ((D@@(2*k-1))(f)'(b) - (D@@(2*k-1))(f)'(a)) * h^{(2*k)}, k=1..m) + O(h^{(2*m+2)});;$

$$T(h) = \int_a^b f(t) dt + \left(\sum_{k=1}^m \frac{\text{bernoulli}(2k) (D^{(2k-1)}(f)(b) - D^{(2k-1)}(f)(a)) h^{(2k)}}{(2k)!} \right) + O(h^{(2m+2)})$$

Für dieses Beispiel verschwinden sämtliche Glieder unter dem Summenzeichen, da aus der Spiegelsymmetrie des Integranden bzgl. $t = a = 0$ und $t = b = \text{Pi}/2$ (siehe Grafik!) folgt, daß alle seine ungeradzahligten Ableitungen ungerade sind und deshalb alle Ausdrücke folgender Gestalt verschwinden

> $(D@@(2*k-1))(f)'(b), (D@@(2*k-1))(f)'(a);$

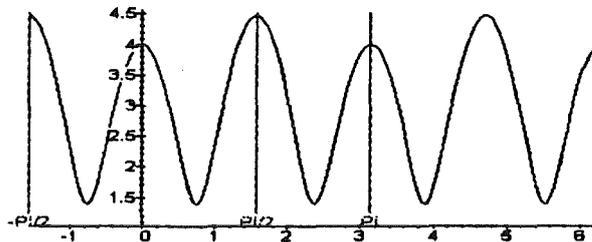
$$D^{(2k-1)}(f)(b), D^{(2k-1)}(f)(a)$$

> p3:= plot(f, -Pi/2..2*Pi):

> p4:= plot({ [Pi/2,t=1.2..4.5], [Pi,t=1.2..4.5], [-Pi/2,t=1.2..4.5] }, color=RED);

> tp := textplot({ [Pi/2, 1.1, 'Pi/2'], [Pi, 1.1, 'Pi'], [-Pi/2, 1.1, '-Pi/2'] }, color=RED);

> display([p3, p4, tp]);



Beweis der behaupteten Symmetrie

> $f(t)=f(-t);$

$$2 \sqrt{\sin(t)^2 + 4 \cos(2t)^2} = 2 \sqrt{\sin(t)^2 + 4 \cos(2t)^2}$$

> $f(\text{Pi}/2+t)=f(\text{Pi}/2-t);$

$$2 \sqrt{\cos(t)^2 + 4 \cos(2t)^2} = 2 \sqrt{\cos(t)^2 + 4 \cos(2t)^2}$$

Damit folgt, daß für dieses Beispiel der Verfahrensfehler der Sehnen-Trapez-Regel rascher gegen Null geht als jede Potenz von h . Die pro Schritt gewonnene Anzahl geltender Stellen nimmt zu:

> Digits := 20: read `romberg.m`;

> for n from 0 to 7 do TRom(n,0) od;

6.6540000191101564350

4.4377207440946697793

4.6962591010940812856

4.7144584725225250175

4.7147158360078806601

4.7147156484750410114

4.7147156484719601600

4.7147156484719601599

Durch das Romberg-Verfahren kann die Konvergenzgeschwindigkeit nicht verbessert werden, tatsächlich wird sie sogar verschlechtert. Die Voraussetzungen für die Schätzung des Fehlers durch die Differenz von im Romberg-Schema nebeneinander stehenden Gliedern ist auch nicht mehr erfüllt.

Probe für das Verschwinden der Glieder unterm Summenzeichen in der asymptotischen Entwicklung

$> [(D^{2k-1}f)(b), (D^{2k-1}f)(a)] \quad k = 1..3;$
 $[0, 0], [0, 0], [0, 0]$

Literatur

1. Stoer, J.: Einführung in die Numerische Mathematik I. Springer, Berlin 1979.
 2. Schwarz, H.R.: Numerische Mathematik. Teubner, Stuttgart 1986
-

Numerische Integration mit dem Romberg-Verfahren
Asymptotische Entwicklung der Spalten des Romberg-Rechenschemas
Schätzwert des Fehlers (Abbruchkriterium)

Ist die Funktion $f(t)$ $(2m+2)$ -mal stetig differenzierbar auf $[a,b]$, so besitzt die Sehnen-Trapez-Regel $T(h)$ folgende asymptotische Entwicklung (Euler-MacLaurinsche Summenformel s. Schwarz [1] S.324, (8.11) und (8.12) oder Stoer [2] S.104ff, 3.2)
 $T(h) = \text{Int}(f(t), t='a'..'b') + \text{Sum}(\text{bernoulli}(2*k)/(2*k)! * (D^{(2*k-1)}(f)(b) - D^{(2*k-1)}(f)(a)) * h^{(2*k)}, k=1..m) + O(h^{(2*m+2)});$

$$T(h) = \int_a^b f(t) dt + \left(\sum_{k=1}^m \frac{\text{bernoulli}(2k) (D^{(2k-1)}(f)(b) - D^{(2k-1)}(f)(a)) h^{(2k)}}{(2k)!} \right) + O(h^{(2m+2)})$$

Mit dem Integranden f (Funktion $f := t \rightarrow f(t)$) und den Integrationsgrenzen $a .. b$
 $> \# f := t \rightarrow (4*\sin(t)^2 + 16*\cos(2*t)^2)^{(1/2)}; a := 0; b := 1;$
 $> f := t \rightarrow (2*\pi)^{(-1/2)} * \exp(-t^2/2); a := 0; b := 2.45;$
 $> \# f := t \rightarrow \text{if } t=0 \text{ then } 1 \text{ else } \sin(t)/t \text{ fi}; a := 0; b := 0.8;$

Rechengenauigkeit

$> \text{Digits} := 10;$
 ergibt sich für
 $> m := 5;$
 beispielsweise folgende asymptotische Entwicklung für die Sehnen-Trapez-Regel

$> \text{read 'romberg.m';}$
 $> T(h) = \text{Int}(f(t), t='a'..'b') + \text{evalf}(SRom(0)) + O(h^{(2*m+2)});$

$$f := t \rightarrow \frac{1}{2} \frac{\sqrt{2} e^{-\frac{1}{2}t^2}}{\sqrt{\pi}}$$

$a := 0$
 $b := 2.45$

$$T(h) = \int_a^b f(t) dt - .004050126522 h^2 + .0002026750814 h^4 + .00001445669158 h^6 - .6066156845 10^{-6} h^8 - .5211502714 10^{-6} h^{10} + O(h^{12})$$

Für $h = (b-a)/2^n$ erhält man eine 0. Folge von Näherungswerten $T(n,0)$ des Integrals.
 Durch Elimination des niedrigsten Fehlerterms aus $T(n,0)$ und $T(n-1,0)$ erhält man eine 1. Folge von verbesserten Näherungswerten $T(n,1)$, hieraus entsprechend eine 2. Folge von verbesserten Näherungswerten $T(n,2)$, usw.
 Diese Folgen bilden jeweils eine Spalte des Romberg-Rechenschemas (hier mit den Zeilen 0 bis $m-1$)

$> \text{SchemaRom}(m-1);$

n	$T(n, 0)$	$T(n, 1)$	$T(n, 2)$	$T(n, 3)$	$T(n, 4)$
0	.5130050525	-----	-----	-----	-----
1	.4872779601	.4787022626	-----	-----	-----
2	.4913670309	.4927300546	.4936652408	-----	-----
3	.4924791263	.4928498246	.4928578093	.4928449929	-----
4	.4927623365	.4928567400	.4928572010	.4928571912	.4928572388

Für die Spalten des Schemas erhält man folgende asymptotische Entwicklungen

$> \text{for } p \text{ from } 0 \text{ to } m-2 \text{ do } T('n',p) = \text{Int}(f(t), t='a'..'b') + \text{evalf}(SRom(p)) + O(h^{(2*m+2)}) \text{ od};$
 $> h = (b-a)/2^n;$

$$T(n, 0) = \int_a^b f(t) dt - .004050126522 h^2 + .0002026750814 h^4 + .00001445669158 h^6 \\ - .6066156845 \cdot 10^{-6} h^8 - .5211502714 \cdot 10^{-6} h^{10} + O(h^{12})$$

$$T(n, 1) = \int_a^b f(t) dt - .0008107003257 h^4 - .0002891338316 h^6 + .00005095571750 h^8 \\ + .0001771910923 h^{10} + O(h^{12})$$

$$T(n, 2) = \int_a^b f(t) dt + .0009252282613 h^6 - .0008152914799 h^8 - .01190724140 h^{10} + O(h^{12})$$

$$T(n, 3) = \int_a^b f(t) dt + .002484697844 h^8 + .1814436785 h^{10} + O(h^{12})$$

$$h = 2.45 \frac{1}{2^n}$$

Die Differenz nebeneinander stehender Glieder im Romberg-Schema liefert einen Schätzwert für den Fehler, falls die höheren Glieder in der asymptotischen Entwicklung gegenüber dem niedrigsten Glied vernachlässigbar klein sind.

Zum Vergleich Numerische Integration durch Maple (Clenshaw-Curtis)

```
> readlib('evalf/int');
> 'evalf/int'( f(x), x=a..b, 12 );
```

.492857189265

Beschreibung der Prozeduren:

Die Prozedur TRom(n,p) berechnet Näherungswerte für das Integral der Funktion

x -> f(x) von a bis b. (options remember)

TRom(n,0) ... Sehnen-Trapez-Regel für 2^n Intervalle (n = 0, 1, 2, ...)

Für p > 0 Richardson-Extrapolation:

TRom(n,p) = (4^p TRom(n,p-1) - TRom(n-1,p-1)) / (4^p - 1) (n = p, p+1, p+2, ...)

Die Prozedur SchemaRom(m) druckt das Romberg-Rechenschema aus

bis einschließlich TRom(m,m).

Die Prozedur SRom(p) berechnet ausgehend von der asymptotischen Entwicklung der

Sehnen-Trapez-Regel (p = 0) die asymptotischen Entwicklungen der Spalten im Romberg-Schema

(p = 0, 1, ...) bis zum Glied mit h^(2*m).

Hierbei ist h = (b - a)/2^n.

```
> restart;
```

```
>
```

Numerische Integration mit dem Romberg-Verfahren

Die Prozedur TRom(n,p) berechnet Näherungswerte für das Integral der Funktion $x \rightarrow f(x)$ von a bis b.

TRom(n,0) ... Sehnen-Trapez-Regel für 2^n Intervalle ($n = 0, 1, 2, \dots$)

Für $p > 0$ Richardson-Extrapolation:

$TRom(n,p) = (4^p TRom(n,p-1) - TRom(n-1,p-1)) / (4^p - 1)$ ($n = p, p+1, p+2, \dots$)

```
> TRom := proc(n,p) local h, i; options remember;
>   if not type(p, integer) or p < 0 then
>     ERROR( `2. Argument muß nichtnegative ganze Zahl sein` ) fi;
>   if not type(n, integer) or n < p then
>     ERROR( `1. Argument muß ganze Zahl und größer gleich 2. Argument sein` ) fi;
>   if p > 0 then TRom(n,p) := (4^p*TRom(n,p-1) - TRom(n-1,p-1))/(4^p - 1)
>   elif n > 0 then h := (b - a)/2^n:
>     TRom(n,p) := evalf( TRom(n-1,p)/2 + h*sum( f( a+(2^i-1)*h ), i=1..2^(n-1) ) )
>   else TRom(n,p) := evalf( (b - a)/2 * ( f(a) + f(b) ) )
>   fi
> end:
```

Die Prozedur SchemaRom(m) druckt das Romberg-Rechenschema aus bis einschließlich TRom(m,m).

```
> SchemaRom := proc(m) local n, p;
>   Rechenschema := array(1..m+2, 1..m+2);
>   Rechenschema[1,1] := `n`;
>   for p from 0 to m do Rechenschema[1,p+2] := T(`n`,p) od:
>   for n from 0 to m do Rechenschema[n+2,1] := n od:
>   for p from 0 to m do
>     for n from 0 to p-1 do
>       Rechenschema[n+2,p+2] := `-----`
>     od:
>     for n from p to m do
>       Rechenschema[n+2,p+2] := TRom(n,p)
>     od
>   od;
>   print( Rechenschema );
> end:
```

Die Prozedur SRom(p) berechnet ausgehend von der asymptotischen Entwicklung der Sehnen-Trapez-Regel ($p = 0$) die asymptotischen Entwicklungen der Spalten im Romberg-Schema ($p = 0, 1, \dots$) bis zum Glied mit $h^{(2*m)}$.

Hierbei ist $h = (b - a)/2^n$.

```
> SRom := proc(p) options remember;
>   if not type(p, integer) or p < 0 or p > m then
>     ERROR( `Argument muß ganze Zahl zwischen 0 und m sein` ) fi;
>   if p > 0 then SRom(p) := (4^p*SRom(p-1) - subs(h=2*h, SRom(p-1)))/(4^p - 1)
>   else SRom(p) := convert(evalf(Sum( bernoulli(2*k)/(2*k)! * ( D@@(2*k-1))(f('b')) - (D@@(2*k-1))(f('a'))
> * h^(2*k), k=1..m)), rational);
>   fi
> end:
Warning, `Rechenschema` is implicitly declared local
```

Ein Anfangswertproblem (AWP) soll mit Hilfe des Euler-Verfahrens numerisch gelöst werden. Wendet man das Euler-Verfahren mit unterschiedlichen Schrittweiten h an und vergleicht die Fehler, so sieht man, daß der globale Fehler ungefähr proportional zu h ist. Es handelt sich also um ein Verfahren 1. Ordnung.

Differentialgleichung 1. Ordnung (Dgl)

> Dgl := diff(y(x), x) = y - 2*x/y;

$$Dgl := \frac{\partial}{\partial x} y(x) = y - 2 \frac{x}{y}$$

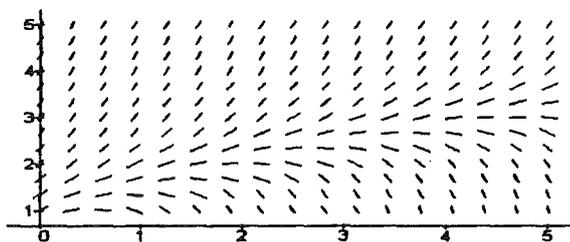
Anfangsbedingung: $y(0) = 1$

Richtungsfeld

```
> with(DEtools): with(plots):
> p1 := dfieldplot( Dgl, [x, y], x=0..5, y=1..5, arrows=LI
> NE, grid = [16, 12] );
> display(p1);
```

Die Linienelemente sind etwas zu lang. Deshalb wird das Richtungsfeld nochmals gezeichnet mit Hilfe von "fieldplot" aus der Share Library (file: plots/ODE). (Dieser file kann mit dem Worksheet ShareODE.ms zugänglich gemacht werden.)

```
> # eq := (x, y) -> rhs(Dgl); #So funktioniert's nicht!?
> eq := unapply(rhs(Dgl), x, y);
> read `ODE.m`:
> p1 := directionfield(eq, 0..5, 1..5):
> display(p1);
```



$$eq := (x, y) \rightarrow y - 2 \frac{x}{y}$$

Folgende Funktion ist die exakte Lösung des AWP

```
> f := (2*x+1)^(1/2);
```

$$f := \sqrt{2x+1}$$

Probe durch einsetzen in die Dgl und vereinfachen

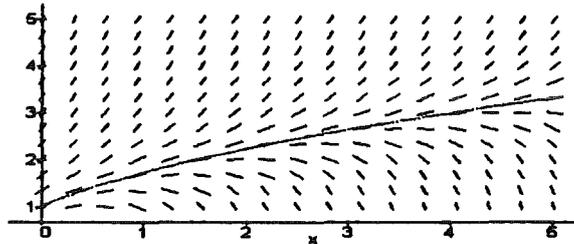
```
> subs( y(x)=f, y=f, Dgl );
> simplify(");
```

$$\frac{\partial}{\partial x} \sqrt{2x+1} = \sqrt{2x+1} - 2 \frac{x}{\sqrt{2x+1}}$$

$$\frac{1}{\sqrt{2x+1}} = \frac{1}{\sqrt{2x+1}}$$

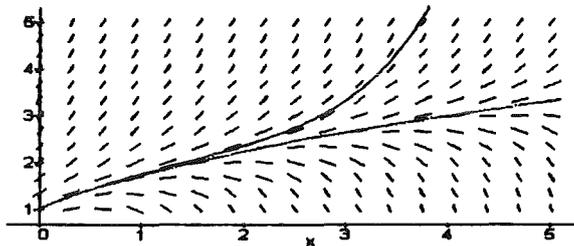
Richtungsfeld mit exakter Lösung des AWP (rot)

```
> p2 := plot( f, x=0..5, color=red );
> display( [p1, p2] );
```



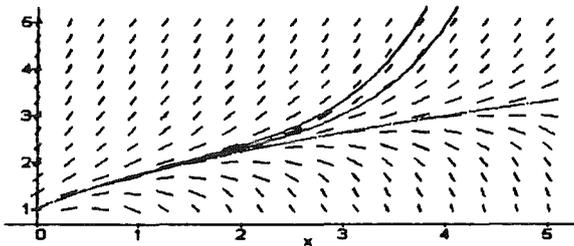
Richtungsfeld mit exakter Lösung des AWP (rot) sowie einer Näherungslösung mit Hilfe des Euler-Verfahrens mit Schrittweite $h = 0.04$

```
> p3 := DEplot1( Dgl, [x, y], x=0..5, {[0, 1]}, y=1..5, ar
> rows=NONE, method='euler', stepsize=0.04 );
> display( [p1, p2, p3] );
```



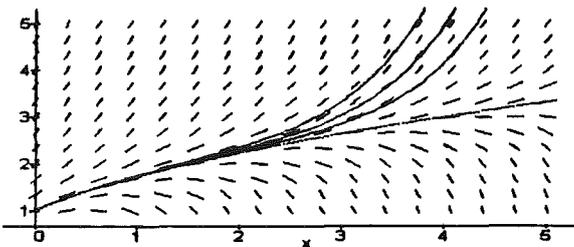
Richtungsfeld mit exakter Lösung des AWP (rot) sowie Näherungslösungen mit Hilfe des Euler-Verfahrens mit Schrittweiten $h = 0.04, 0.02$

```
> p4 := DEplot1( Dgl, [x, y], x=0..5, {[0, 1]}, y=1..5, ar
> rows=NONE, method='euler', stepsize=0.04, iterations=2):
> display( [p1, p2, p3, p4] );
```



Richtungsfeld mit exakter Lösung des AWP (rot) sowie Näherungslösungen mit Hilfe des Euler-Verfahrens mit Schrittweiten $h = 0.04, 0.02, 0.01$

```
> p5 := DEplot1( Dgl, [x, y], x=0..5, {[0, 1]}, y=1..5, ar
> rows=NONE, method='euler', stepsize=0.04, iterations=4):
> display( [p1, p2, p3, p4, p5] );
```



Ein Anfangswertproblem (AWP) soll mit Hilfe des Runge-Kutta-Verfahrens numerisch gelöst werden. Wendet man das Runge-Kutta-Verfahren mit unterschiedlichen Schrittweiten h an und vergleicht die Fehler, so sieht man, daß der globale Fehler ungefähr proportional zu h^4 ist. Es handelt sich also um ein Verfahren 4. Ordnung.

Differentialgleichung 1. Ordnung (Dgl)

```
> Dgl := diff(y(x), x) = y - 2*x/y;
```

$$Dgl := \frac{\partial}{\partial x} y(x) = y - 2 \frac{x}{y}$$

Anfangsbedingung: $y(0) = 1$

Richtungsfeld

```
> with(DEtools): with(plots):  
> eq := unapply(rhs(Dgl), x, y);  
> read `ODE.m`:
```

$$eq := (x, y) \rightarrow y - 2 \frac{x}{y}$$

Folgende Funktion ist die exakte Lösung des AWP

```
> f := (2*x+1)^(1/2);
```

$$f := \sqrt{2x+1}$$

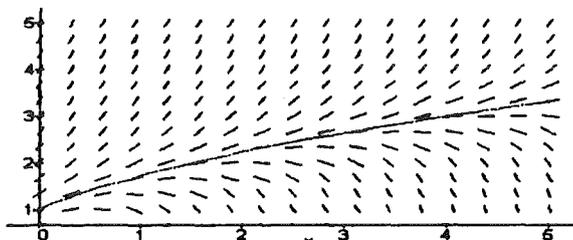
Probe durch einsetzen in die Dgl und vereinfachen

```
> simplify( subs( y(x)=f, y=f, Dgl ) );
```

$$\frac{1}{\sqrt{2x+1}} = \frac{1}{\sqrt{2x+1}}$$

Richtungsfeld mit exakter Lösung des AWP (rot)

```
> p2 := plot( f, x=0..5, color=red );  
> display( [p1, p2] );
```



Richtungsfeld mit exakter Lösung des AWP (rot) sowie einer Näherungslösung mit Hilfe des Runge-Kutta-Verfahrens mit Schrittweite $h = 0.1$

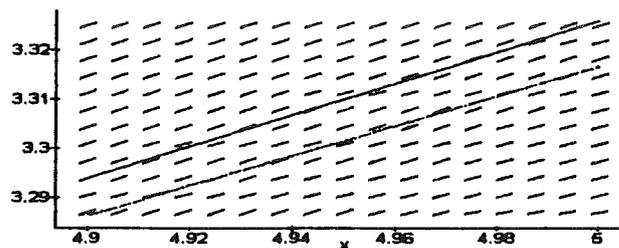
```
> p3 := DEplot1( Dgl, [x, y], x=0..5, {[0, 1]}, y=1..5, ar  
> rows=NONE, method='rk4', stepsize=0.1 );  
> display( [p1, p2, p3] );
```

Exakte Lösung und Näherungslösung sind in der Grafik nicht zu unterscheiden.

Deshalb: Ausschnitt aus der Grafik mit $4.9 \leq x \leq 5.0$, $3.287 \leq y \leq 3.325$

```
> p4 := directionfield( eq, 4.9..5, 3.287..3.325, grid=[16  
> , 11] );  
> p5 := plot( f, x=4.9..5, color=red );  
> p6 := DEplot( Dgl, [x, y], x=4.9..5, {[0, 1]}, arrows=NO
```

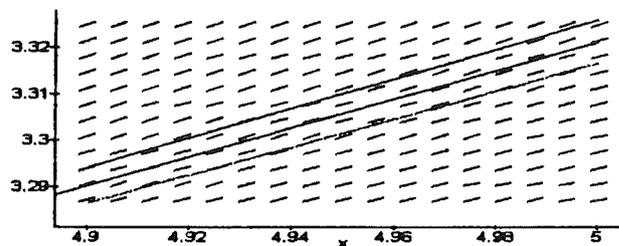
```
> NE, method='rk4', stepsize=0.1 );
> display( [p4, p5, p6] );
```



Richtungsfeld mit exakter Lösung des AWP (rot) sowie Näherungslösungen mit Hilfe des Runge-Kutta-Verfahrens mit Schrittweiten $h_1 = 0.1$, $h_2 = 0.08409$

(Damit gilt $(h_2^4) : (h_1^4) = 0.5$)

```
> p7 := DEplot( Dgl, [x, y], x=4.9..5, {[0, 1]}, arrows=NO
> NE, method='rk4', stepsize=0.08409 );
> display( [p4, p5, p6, p7] );
> h2^4/h1^4 = 0.08409^4/0.1^4;
```

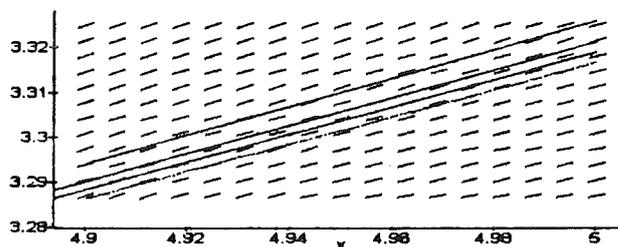


$$\frac{h_2^4}{h_1^4} = .5000085261$$

Richtungsfeld mit exakter Lösung des AWP (rot) sowie Näherungslösungen mit Hilfe des Runge-Kutta-Verfahrens mit Schrittweiten $h_1 = 0.1$, $h_2 = 0.08409$, $h_3 = 0.07071$

(Damit gilt $(h_3^4) : (h_2^4) = 0.5$)

```
> p8 := DEplot( Dgl, [x, y], x=4.9..5, {[0, 1]}, arrows=NO
> NE, method='rk4', stepsize=0.07071 );
> display( [p4, p5, p6, p7, p8] );
> h3^4/h2^4 = 0.07071^4/0.08409^4;
```



$$\frac{h_3^4}{h_2^4} = .4999722946$$

Der file plots/ODE aus der Share Library wird folgendermaßen für Maple V zugänglich gemacht (vgl. Ellis,W., Johnson,E., Lodi,E., Schwalbe,D., Maple V in der mathematischen Anwendung, Flight Manual deutsche Ausgabe (International Thomson Publishing) 1994, S. 133, Kapitel 5 Differentialgleichungen)

```
> with(share);
Unable to find the share library.
```

[]

Funktioniert obiges Vorgehen nicht, dann libname abfragen

```
> libname;
```

```
C:\MAPLEV3\UPDATE, C:\MAPLEV3\LIB
```

und sharename selbst definieren: dabei ist lib durch share zu ersetzen

```
> sharename:=`c:/maplev3/share`; # Hier wurde der in Release 3 übliche Pfad eingesetzt
> with(share);
```

```
sharename := c:/maplev3/share
```

```
See ?share and ?share,contents for information about the share library
```

[]

Einlesen von plots/ODE aus der Share Library

```
> # readshare(ODE, plots); # Funktioniert erst ab Release 3
> read ``.sharename.`\plots\ODE`; # Funktioniert für Release 2 und 3
> # read ``.sharename.`/plots/ODE`; # Vereinfachte Schreibweise
    [ directionfield, phaseplot, impeuler, rungekutta, rungekuttahf ]
    directionfield is the new name for fieldplot in Release 2.
```

Jetzt können die Funktionen aus plots/ODE benutzt werden.

Falls einem der Zugriff auf die Share Library zu umständlich ist, kann man den Zustand dieses Worksheets als ODE.m sichern. Dann kann man künftig mit

```
> read `ODE.m`
```

die Funktionen aus plots/ODE verfügbar machen.

```
> save `ODE.m`;
```

```
>
```

Ellis/ Johnson/ Lodi/ Schwalbe: Maple V in der mathematischen Anwendung (Maple Flight Manual)

4.7 Diagonalisierung und Ähnlichkeit
4.7.5 Zusätzliche Übungen

Aufgabe 3

Gegeben sei eine n-reihige Hilbert-Matrix H.

- a) Ist H diagonalisierbar?
- b) Bestimme eine Diagonalisierungsmatrix P mit $D = P^{-1} * H * P = \text{Diagonalmatrix!}$
- c) Berechne P^{-1} und das Matrizenprodukt $P^{-1} * H * P!$

```
> with(linalg):
Warning: new definition for      norm
Warning: new definition for      trace
```

```
Setze n = 3, 4, 5
> n := 4;
> H := hilbert(n);
```

$$n := 4$$

$$H := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

```
a) Vielfachheit gleich Ordnung der Eigenwerte ==> H diagonalisierbar
> eigsys := eigenvecs(H);
```

$$\text{eigsys} := \left[\%1, 1, \left\{ \left[\frac{12434}{45} \%1 - \frac{5590}{3} \%1^2 + \frac{11}{1080} + 1120 \%1^3 \right. \right. \right. \\ \left. \left. \left. 7288 \%1^2 - 4368 \%1^3 - \frac{82621}{75} \%1 - \frac{139}{450} \quad 1 \quad \frac{497609}{450} \%1 - \frac{21616}{3} \%1^2 - \frac{511}{675} + 4312 \%1^3 \right] \right\} \right]$$

$$\%1 := \text{RootOf}(6048000 _Z^4 - 10137600 _Z^3 + 1603680 _Z^2 - 10496 _Z + 1)$$

```
b) Diagonalisierungsmatrix P: Spalten von P = Eigenvektoren von H
> t := \%1; # falls n > 2
> # t := op(1, eigsys); # falls n = 2
```

$$t := \text{RootOf}(6048000 _Z^4 - 10137600 _Z^3 + 1603680 _Z^2 - 10496 _Z + 1)$$

all = alle Eigenwerte von H = alle Lösungen von RootOf(...)
Für n > 4 nicht durch Radikale ausdrückbar, daher erhält man dann nur Gleitpunktnäherungen, so daß ab hier nur noch mit Gleitpunktarithmetik gerechnet wird!

```
> Digits := 20: all := allvalues(t);
> P := augment('subs( t=all[i], op(eigsys[3]) )' $ i=1..n);
```

$$P := \left[\frac{1459051}{12600} + \frac{6217}{850500} \sqrt{\%2} + \frac{6217}{850500} \%3 - \frac{5590}{3} \%4^2 + 1120 \%4^3, \right.$$

$$\frac{1459051}{12600} + \frac{6217}{850500} \sqrt{\%2} - \frac{6217}{850500} \%3 - \frac{5590}{3} \%5^2 + 1120 \%5^3,$$

$$\frac{1459051}{12600} - \frac{6217}{850500} \sqrt{\%2} + \frac{6217}{850500} \%6 - \frac{5590}{3} \%7^2 + 1120 \%7^3,$$

$$\frac{1459051}{12600} - \frac{6217}{850500} \sqrt{\%2} - \frac{6217}{850500} \%6 - \frac{5590}{3} \%8^2 + 1120 \%8^3$$

$$\left[7288 \%4^2 - 4368 \%4^3 - \frac{346453}{750} - \frac{11803}{405000} \sqrt{\%2} - \frac{11803}{405000} \%3,$$

$$7288 \%5^2 - 4368 \%5^3 - \frac{346453}{750} - \frac{11803}{405000} \sqrt{\%2} + \frac{11803}{405000} \%3,$$

$$7288 \%7^2 - 4368 \%7^3 - \frac{346453}{750} + \frac{11803}{405000} \sqrt{\%2} - \frac{11803}{405000} \%6,$$

$$7288 \%8^2 - 4368 \%8^3 - \frac{346453}{750} + \frac{11803}{405000} \sqrt{\%2} + \frac{11803}{405000} \%6 \left. \right]$$

[1, 1, 1, 1]

$$\left[\frac{520453}{1125} + \frac{71087}{2430000} \sqrt{\%2} + \frac{71087}{2430000} \%3 - \frac{21616}{3} \%4^2 + 4312 \%4^3,$$

$$\frac{520453}{1125} + \frac{71087}{2430000} \sqrt{\%2} - \frac{71087}{2430000} \%3 - \frac{21616}{3} \%5^2 + 4312 \%5^3,$$

$$\frac{520453}{1125} - \frac{71087}{2430000} \sqrt{\%2} + \frac{71087}{2430000} \%6 - \frac{21616}{3} \%7^2 + 4312 \%7^3,$$

$$\frac{520453}{1125} - \frac{71087}{2430000} \sqrt{\%2} - \frac{71087}{2430000} \%6 - \frac{21616}{3} \%8^2 + 4312 \%8^3 \left. \right]$$

$$\%1 := \frac{3805076179}{6751269000000} + \frac{1}{2250423000} I \sqrt{13863954291}$$

$$\%2 := \frac{187760700 \%1^{1/3} + 357210000 \%1^{2/3} + 2444281}{\%1^{1/3}}$$

$$\%3 := \left(- \frac{-375521400 \%1^{1/3} \sqrt{\%2} + 357210000 \sqrt{\%2} \%1^{2/3} + 2444281 \sqrt{\%2} - 4971476736000 \%1^{1/3}}{\%1^{1/3} \sqrt{\%2}} \right)$$

1/2

$$\%4 := \frac{44}{105} + \frac{1}{37800} \sqrt{\%2} + \frac{1}{37800} \%3$$

$$\%5 := \frac{44}{105} + \frac{1}{37800} \sqrt{\%2} - \frac{1}{37800} \%3$$

$$\%6 := \left(- \frac{-375521400 \%1^{1/3} \sqrt{\%2} + 357210000 \sqrt{\%2} \%1^{2/3} + 2444281 \sqrt{\%2} + 4971476736000 \%1^{1/3}}{\%1^{1/3} \sqrt{\%2}} \right)$$

1/2

$$\%7 := \frac{44}{105} - \frac{1}{37800} \sqrt{\%2} + \frac{1}{37800} \%6$$

$$\%8 := \frac{44}{105} - \frac{1}{37800} \sqrt{\%2} - \frac{1}{37800} \%6$$

Vereinfachung bringt nichts:

> # Pc := map(evalc, P);

> # Pcs := map(simplify, Pc);

c) HP := H * P

> HP := evalm(H &* P);

IP := P⁽⁻¹⁾

Dauert zu lang für n = 4 (Auf PC 486DX33 nach einer Stunde noch kein Ergebnis!)

> IP := inverse(P);

IPHP := P⁽⁻¹⁾ * H * P

> IPHP := evalm(IP &* HP);

> map(evalf, IPHP): evalf(" , 4);

Vergleiche die Eigenwerte

> evalf(all): evalf(" , 4);

Für n = 4 ist die exakte Berechnung der Inversen zu aufwendig, daher Gleitpunktarithmetik:

> Digits := 10;

> PN := map(x -> Re(evalf(x)), P);

Digits := 10

$$PN := \begin{bmatrix} 2.458339 & -1.142268652 & 1.787786342 & .03688529157 \\ 1.4016675 & .7270756 & -7.402298780 & -.415339920 \\ 1. & 1. & 1. & 1. \\ .78210 & 1.00877129 & 6.368305075 & -.6501807216 \end{bmatrix}$$

> evalm(inverse(PN) &* H &* PN);

$$\begin{bmatrix} 1.500214280 & .73 \cdot 10^{-8} & -.160 \cdot 10^{-7} & -.2791 \cdot 10^{-7} \\ .185466 \cdot 10^{-5} & .1691412204 & -.1 \cdot 10^{-9} & -.188 \cdot 10^{-8} \\ -.11217118 \cdot 10^{-5} & -.3384 \cdot 10^{-9} & .006738273728 & -.89450 \cdot 10^{-8} \\ -.180402448 \cdot 10^{-5} & .399971 \cdot 10^{-8} & .74906 \cdot 10^{-8} & .00009670227791 \end{bmatrix}$$

Vergleiche die Eigenwerte

> evalf(all): evalf(");

$$1.500214280 - .1327808532 \cdot 10^{-11} I, .1691412203 + .4302606422 \cdot 10^{-11} I,$$

$$.006738281730 + .3554955644 \cdot 10^{-9} I, .000096693670 - .3584703622 \cdot 10^{-9} I$$

Reine Gleitpunktarithmetik ist schneller

> HN := map(evalf, H);

> eigsys := eigenvects(HN);

> R := augment('op(eigsys[i][3])' \$ 'i'=1..n);

> evalm(inverse(R) &* HN &* R);

$$\begin{bmatrix} 1.500214280 & 0 & -.3 \cdot 10^{-9} & .8 \cdot 10^{-9} \\ -.325 \cdot 10^{-9} & .006738273573 & -.21 \cdot 10^{-10} & -.50 \cdot 10^{-10} \\ -.50 \cdot 10^{-9} & -.8 \cdot 10^{-10} & .1691412201 & .20 \cdot 10^{-9} \\ .199805 \cdot 10^{-8} & -.384 \cdot 10^{-11} & .20784 \cdot 10^{-9} & .00009670231489 \end{bmatrix}$$

>

Nullstellen einer transzendenten Funktion

```
> f := exp(-3*x) - (sin(x))^3;
```

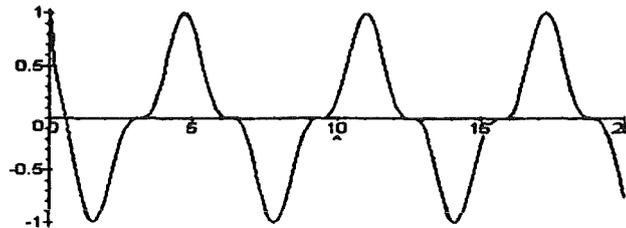
$$f := e^{(-3x)} - \sin(x)^3$$

Exakte Bestimmung der Nullstellen nicht möglich!

```
> Nst := solve(f=0, x);
```

Nst :=

```
> plot(f, x=0..20);
```



Berechnung von Näherungswerten $Nst.i$ für Nullstellen in der Nähe von $\pi \cdot (i-1)$ und der Funktionswerte $Fktswert.i$ an diesen Stellen ($i = 1, 2, \dots, n$):

```
> n := 7;
```

```
> for i from 1 to n
```

```
>   do Nst.i := fsolve( f=0, x, Pi*(i-1.5)..Pi*(i-0.5) );
```

```
>     Fktswert.i := evalf(subs(x=", f), 12);
```

```
>   od;
```

Nst1 := .5885327440

Fktswert1 := -.24 10⁻¹⁰

Nst2 := 3.096363932

Fktswert2 := -.24021 10⁻¹¹

Nst3 := 6.285049273

Fktswert3 := .399520 10⁻¹⁴

Nst4 := 9.424697255

Fktswert4 := .5108968 10⁻¹⁷

Nst5 := 12.56637410

Fktswert5 := .616059541 10⁻¹⁹

Nst6 := 15.70796312

Fktswert6 := .18415044058 10⁻²¹

Nst7 := 18.84955593

Fktswert7 := -.329560906293 10⁻²⁴

Es ist nicht klar, wie genau die Nullstellen berechnet wurden und ob weitere Nullstellen in ihrer Nähe liegen!

Literatur:

Hammer, R./ Hocks, M./ Kulisch, U./Ratz, D.: Numerical Toolbox for verified Computing I, Basic Numerical Problems (Springer-Verlag) ISBN 3-540-57118-3 (1993), p. 101, 6.3.2 Example.

