

Markus Lebender

**Implementierung numerischer
verifizierender Verfahren mit graphischer
Benutzerschnittstelle unter Oberon-XSC**

Diplomarbeit

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 1998 Diplomica Verlag GmbH
ISBN: 9783832466831

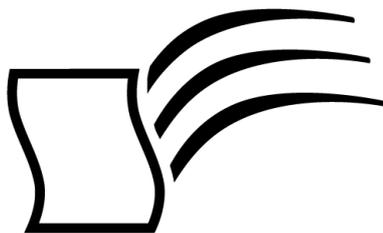
Markus Lebender

**Implementierung numerischer verifizierender Verfahren
mit graphischer Benutzerschnittstelle unter Oberon-XSC**

Markus Lebender

Implementierung numerischer verifizierender Verfahren mit graphischer Benutzerschnittstelle unter Oberon-XSC

Diplomarbeit
an der Universität Fridericiana Karlsruhe (TH)
Fachbereich Mathematik
Institut für Angewandte Mathematik
April 1998 Abgabe



Diplom.de

Diplomica GmbH _____
Hermannstal 119k _____
22119 Hamburg _____

Fon: 040 / 655 99 20 _____
Fax: 040 / 655 99 222 _____

agentur@diplom.de _____
www.diplom.de _____

ID 6683

Lebender, Markus: Implementierung numerischer verifizierender Verfahren mit graphischer Benutzerschnittstelle unter Oberon -XSC

Hamburg: Diplomica GmbH, 2003

Zugl.: Karlsruhe, Technische Universität, Diplomarbeit, 1998

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden, und die Diplomarbeiten Agentur, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Diplomica GmbH

<http://www.diplom.de>, Hamburg 2003

Printed in Germany

Inhaltsverzeichnis

1	Grundlagen	1
1.1	Kurze Einführung in die Rechnerarithmetik	1
1.1.1	Die Räume der Rechnerarithmetik	1
1.1.2	Definition der Rechnerarithmetik	3
1.2	Einführung in die Intervallrechnung	5
1.2.1	Definition und Eigenschaften der Intervallararithmetik	6
1.2.2	Maschinenintervalle	8
1.2.3	Erweiterte Intervallararithmetik	9
1.3	Einführung in Oberon	11
1.3.1	Die Programmiersprache Oberon	11
1.3.2	Das Oberon-System	15
1.3.3	Programmierung von Objekten und Gadgets	18
2	Visualisierung	21
2.1	Der FunctionViewer	21
2.1.1	Gundlegende Module	22
2.1.2	Module zur Eingabe eindimensionaler Funktionen	25
2.1.3	Module zur direkten Implementierung des FunctionViewer	31
2.2	Eine Pendelsimulation	42
2.2.1	Iteratoren	43
2.2.2	Listenverwaltung	49
2.2.3	Die Pendelsimulation	57
2.3	Benutzeroberflächen	59
2.3.1	Benutzeroberfläche zum FunctionViewer	59
2.3.2	Benutzeroberfläche für die Pendelsimulation	60
2.4	Zusammenfassung	61
3	Numerische Verfahren	62
3.1	Automatische Differentiation	62
3.1.1	Motivation	62
3.1.2	Die Taylorarithmetik	63
3.1.3	Algorithmische Beschreibung der Taylorarithmetik	69
3.1.4	Implementierung	72
3.2	Globale Optimierung	74

3.2.1	Problemstellung	74
3.2.2	Testmethoden	75
3.2.3	Teilungsmethoden	79
3.2.4	Verifikation	80
3.2.5	Algorithmische Beschreibung	80
3.2.6	Implementierung	87
3.2.7	Graphische Benutzerschnittstelle	89
A	Bedienung des Oberon-Systems	90
B	OBERON-XSC	92
C	Hilfsmodule	97
D	Testfunktionen	99

Abbildungsverzeichnis

1.1	Die Räume des numerischen Rechnens	2
1.2	Herkömmliche Definition der Rechnerarithmetik	3
1.3	Definition der Rechnerarithmetik mittels Semimorphismus	4
1.4	Die 15 Grundoperationen	5
1.5	Modulaufbau des Oberon-Systems	16
1.6	Aufbau des Gadgets-System	17
1.7	Container-Aufbau des Bildschirms	19
1.8	Koordinaten im Gadgets-System	19
1.9	Bildschirmmasken	20
2.1	Die Modulstruktur des Gadgets FunctionViewer	21
2.2	Der Funktionsplot für $\sin(x)$ und zwei Taylorkurven	42
2.3	Nachrichtenversand beim FunctionViewer	42
2.4	Modulstruktur der Pendelsimulation	43
2.5	Aufbau einer doppeltverketteten Liste	49
2.6	Aufbau einer ObjList	51
2.7	Das Panel FunctionViewers.Panel	59
2.8	Die Pendelsimulation	61
3.1	Taylorkoeffizienten für Funktionen mit $f' = 1/g$	68
3.2	Der Mittelpunkttest	76
3.3	Der Monotonietest	76
3.4	Der Konkavitätstest	77
3.5	Newtonschrift bei erweiterter Intervalldivision	78
3.6	Das Boxingverfahren	80
3.7	Benutzerpanel zur globalen Optimierung	89
B.1	Die Standardfunktionen	93
D.1	Bild der Funktion Shekel5, $x \in [-15, 15]$, $y \in [-20, 5]$	102
D.2	Funktion fd4, $x \in [-10, 10]$, $y \in [-100, 500]$	103
D.3	Funktion fd4, $x \in [-1, 3]$, $y \in [-1, 3]$	104

Verzeichnis der Algorithmen

2.1	Behandlung der <code>RealSizeMsg</code>	33
2.2	Anpassen einer <code>Drawing</code> an ein <code>Coordinates.Coordsystem</code>	38
2.3	Ablauf eines Iterationsverfahrens	46
2.4	Einfügen eines Listenelements	53
2.5	Vergleich zweier <code>PointLists.Point</code>	56
3.1	Addition für Taylor-Tupel	69
3.2	Subtraktion für Taylor-Tupel	69
3.3	Multiplikation für Taylor-Tupel	69
3.4	Division für Taylor-Tupel	70
3.5	Quadratfunktion für Taylor-Tupel	70
3.6	Quadratwurzel für Taylor-Tupel	70
3.7	Exponentialfunktion für Taylor-Tupel	70
3.8	Hilfsalgorithmus für <code>sin</code> und <code>cos</code> der Taylorarithmetik	71
3.9	<code>Sin-Funktion</code> für Taylor-Tupel	71
3.10	<code>Cos-Funktion</code> für Taylor-Tupel	71
3.11	<code>Potenzfunktion</code> für Taylor-Tupel	71
3.12	Generische Funktion für Taylor-Tupel	71
3.13	Der Monotonietest	81
3.14	Der Konkavitätstest	81
3.15	Der Erweitete Intervall Newton Schritt	82
3.16	Die Bisektion	82
3.17	Das Boxingverfahren	82
3.18	Bearbeitung des aktuellen Intervalls	83
3.19	Komprimierung einer Liste	85
3.20	Der Verifikationsschritt	85
3.21	Die globale Optimierung	86

Verzeichnis der Moduldefinitionen

LogOut	22
XStrings	23
FunctionTrees	27
FunctionParser	30
Coordinates	31
FVDrawings	34
FunctionViewers	40
Iterators	44
RungeKutta	48
ObjLists	50
PointLists	55
PendelFrames	58
FVUI	59
PendelUI	60
Tari	72
RTari	73
Gop1	87
DDFari	88
GopLists	88
lari	94
dot	96
lutil	97
Xlari	98

Vorwort

Im Titel dieser Arbeit finden sich zwei Schwerpunkte. Den einen bilden die numerischen verifizierenden Verfahren. Das sind mathematische Algorithmen, die zusätzlich zur Lösung eines Problems gleichzeitig einen Beweis der Richtigkeit dieser Lösung liefern. Der zweite Schwerpunkt ist die graphische Benutzerschnittstelle. Moderne Software ist ohne Fenster, Buttons und andere graphische Elemente nicht mehr denkbar. Zudem ermöglicht Graphik auch, mathematische Sachverhalte sichtbar zu visualisieren und dadurch zu veranschaulichen. Diese beiden Schwerpunkte werden in dieser Arbeit nun unter Oberon-XSC zusammengeführt. Denn das Oberon-System und die Programmiersprache Oberon bieten gute Möglichkeiten der Graphikprogrammierung und die XSC-Erweiterung liefert den Grundstock für die selbstverifizierenden Verfahren. Dementsprechend ist auch der Aufbau der Arbeit.

Nach einem einführenden Kapitel über die mathematischen Grundlagen der Rechnerarithmetik und der Intervallrechnung sowie einer kurzen Einführung in Oberon, folgt das Kapitel "Visualisierung". In diesem wird zunächst ein Werkzeug zur Darstellung eindimensionaler reeller Funktionen vorgestellt: Der `FunctionViewer`. Er ermöglicht z.B. die Darstellung einer Funktion sowie das gleichzeitige Anzeigen einer Funktion und ihrer Ableitungen bzw. Taylorkurven, ohne daß diese explizit angegeben oder programmiert werden müssen. Eingebunden hierbei ist auch ein Modul zur interaktiven Eingabe einfacher Funktionen, so daß nicht extra eine Oberonprozedur programmiert werden muß, die die zu zeichnende Funktion implementiert.

Als zweites Beispiel zur Visualisierung folgt dann eine Fadenpendel-Simulation, für die ein System zweier Differentialgleichungen erster Ordnung durch ein Runge-Kutta-Verfahren der Ordnung 4 gelöst wird.

Bei der Entwicklung dieser beiden Beispiel entstanden natürlich viele grundlegende allgemeine Module, z.B. die oben erwähnte interaktive Eingabe von Funktionen oder die Umsetzung von reellen Koordinaten in Oberonkoordinaten, die auch für andere Visualisierung genutzt werden können.

Das dritte Kapitel beschäftigt sich mit den numerischen Verfahren. Exemplarisch wurden hierfür die automatische Differentiation und die eindimensionale globale Optimierung gewählt. Die einzelnen Abschnitte dieses Kapitels stellen zunächst die grundlegende Aufgabenstellung dar, entwickeln anschließend den mathematischen Algorithmus und enden dann in der konkreten Implementierung. Auf die Wiedergabe von Quelltext wird in dieser Arbeit weitgehend verzichtet. Der Arbeit liegt eine Diskette mit allen Quellen der im Rahmen dieser Arbeit ent-

standenen Module bei.

Die angegebenen Moduldefinitionen entsprechen ungefähr den Definitionsdateien, die der Oberon-Browser erzeugt. Im Gegensatz zu diesen enthalten die Moduldefinitionen aber keine `IMPORT`-Zeile, die die importierten Module aufzählt. Für diese Information sei auf die Quelltexte und die Diagramme in den entsprechenden Kapiteln verwiesen.