

Christoph W. Ueberhuber

# Numerical **1** Computation

Methods, Software, and Analysis



Springer

Christoph W. Ueberhuber

# NUMERICAL COMPUTATION 1

Methods, Software, and Analysis

With 157 Figures



Springer

Christoph W. Ueberhuber

Technical University of Vienna

Wiedner Hauptstrasse 8-10/115

A-1040 Vienna

Austria

e-mail: cwu@uranus.tuwien.ac.at

Title of the German original edition: *Computer-Numerik 1*. Published by Springer, Berlin Heidelberg 1995.

ROBERT LETTNER created the picture on the cover of this book. He is the person in charge of the central printing and reprography office at the Academy of Applied Arts in Vienna, Austria. In the past few years Robert Lettner, inspired by advancements in digital image processing, has developed new forms of aesthetics.

Sequences of pictures recently created by Robert Lettner include

*Das Spiel vom Kommen und Gehen (Coming and Going)*,

*Dubliner Thesen zur informellen Geometrie (Dublin Theses on Informal Geometry)*,

*Bilder zur magischen Geometrie (Magical Geometry)*.

The illustration on the cover of this book was taken from the sequence

*Das Spiel vom Kommen und Gehen*.

Mathematics Subject Classification: 65-00, 65-01, 65-04, 65Dxx, 65Fxx, 65Hxx, 65Y10, 65Y15, 65Y20

#### Library of Congress Cataloging-in-Publication Data

Ueberhuber, Christoph W. 1946-

[Computer Numerik. English] Numerical computation : methods, software, and analysis /

Christoph W. Ueberhuber. p. cm.

Includes bibliographical references and index.

ISBN 3-540-62058-3 (v. 1 : soft : acid-free paper). --

ISBN 3-540-62057-5 (v. 2 : soft : acid-free paper)

1. Numerical analysis -- Data processing. I. Title.

QA297.U2413 1997

96-46772

519.4'0285'53 --dc21

CIP

ISBN 3-540-62058-3 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1997

Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: By the author using L<sup>A</sup>T<sub>E</sub>X.

SPIN 10543814

41/3143 - 5 4 3 2 1 0 - Printed on acid-free paper

46

**Springer**

*Berlin*

*Heidelberg*

*New York*

*Barcelona*

*Budapest*

*Hong Kong*

*London*

*Milan*

*Paris*

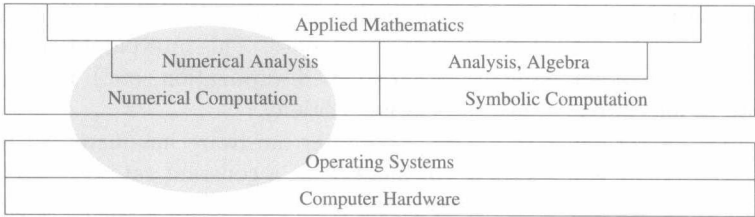
*Santa Clara*

*Singapore*

*Tokyo*

# Preface

This book deals with various aspects of scientific numerical computing. No attempt was made to be complete or encyclopedic. The successful solution of a numerical problem has many facets and consequently involves different fields of computer science. Computer numerics—as opposed to computer algebra—is thus based on applied mathematics, numerical analysis and numerical computation as well as on certain areas of computer science such as computer architecture and operating systems.



Each chapter begins with sample situations taken from specific fields of application. Abstract and general formulations of mathematical problems are then presented. Following this abstract level, a general discussion about principles and methods for the numerical solution of mathematical problems is presented. Relevant algorithms are developed and their efficiency and the accuracy of their results is assessed. It is then explained as to how they can be obtained in the form of numerical software. The reader is presented with various ways of applying the general methods and principles to particular classes of problems and approaches to extracting practically useful solutions with appropriately chosen numerical software are developed. Potential difficulties and obstacles are examined, and ways of avoiding them are discussed.

The volume and diversity of all the available numerical software is tremendous. When confronted with all the available software, it is important for the user to have a broad knowledge of numerical software in general and to know where this software can be optimally applied in order for him to be able to choose the most appropriate software for a specific problem. Assistance in this complicated matter is offered in this book.

This book gives a comprehensive survey of the high quality software products available and the methods and algorithms they are based on. Positive properties and inherent weaknesses of specific software products are pointed out. Special subsections in this book, devoted to software, provide relevant information about commercially available software libraries (IMSL, NAG, etc.) as well as public domain software (such as the NETLIB programs) which can be downloaded from the Internet.

This book addresses people interested in the numerical solution of mathematical problems, who wish to make a good selection from the wealth of available software products, and who wish to utilize the functionality and efficiency of modern numerical software. These people may be students, scientists or engineers. Accordingly, this monograph may be used either as a textbook or as a reference book.

The German version *Computer-Numerik* was published in 1995 by Springer-Verlag, Heidelberg. The English version, however, is not merely a translation; the book has also been revised and updated.

## Synopsis

**Volume I** starts with a short introduction into scientific model building, which is the foundation of all numerical methods which rely on finite models that replace infinite mathematical objects and techniques. This unavoidable finitization (found in floating-point numbers, truncation, discretization, etc.) is introduced in Chapter 2 and implications are discussed briefly.

The peak performance of modern computer hardware has increased remarkably in recent years and continues to increase at a constant rate of nearly 100 % every year. On the other hand, there is a steadily increasing gap between the potential performance and the empirical performance values of contemporary computer systems. Reasons for this development are examined and remedial measures are presented in Chapter 3.

Chapter 4 is dedicated to the objects of all numerical methods—numerical data—and to the operations they are used in. The main emphasis of this chapter is on standardized floating-point systems as found on most computers used for numerical data processing. It is explained as to how portable programs can be written to adapt themselves to the features of a specific number system.

Chapter 5 deals with the foundations of algorithm theory, in so far as this knowledge is important for numerical methods. Floating-point operations and arithmetic algorithms, which are the basic elements of all other numerical algorithms, are dealt with extensively.

Chapter 6 presents the most important quality attributes of numerical software. Particular attention is paid to techniques which provide for the efficient utilization of modern computer systems when solving large-scale numerical problems.

Chapter 7 gives an overview of readily available commercial or public domain software products. Numerical programs (published in TOMS or in other journals or books), program packages (LAPACK, QUADPACK etc.), and software libraries are dealt with. Particular emphasis is placed on software available on the Internet (NETLIB, ELIB etc.).

Chapter 8 deals with modeling by approximation, a technique important in many fields of numerical data processing. The applications of these techniques range from data analysis to the modeling processes that take place in numerical programs (such as the process found in numerical integration programs in which

the integrand function is replaced by a piecewise polynomial).

The most effective approach to obtaining model functions which approximate given data is interpolation. Chapter 9 gives the theoretical background necessary to understanding particular interpolation methods and programs. In addition, the algorithmic processing of polynomials and various spline functions is presented.

**Volume II** begins with Chapter 10 which contains best approximation techniques for linear and nonlinear data fitting applications.

Chapter 11 is dedicated to a very important application of approximation: the Fourier transform. In particular, the fast Fourier transform (FFT) and its implementations are dealt with.

The subject of Chapter 12 is numerical integration. Software products which solve univariate integration problems are systematically presented. Topical algorithms (such as lattice rules) are covered in this chapter to enable the user to produce tailor-made integration software for solving multivariate integration problems (which is very useful since there is a paucity of ready made software for these problems).

The solution of systems of linear equations is undoubtedly the most important field of computer numerics. Accordingly, it is the field in which the greatest number of software products is available. However, this book deals primarily with LAPACK programs. Chapter 13 answers many questions relevant to users: How should algorithms and software products appropriate for a given problem be chosen? What properties of the system matrix influence the choice of programs? How can it be ascertained whether or not a program has produced an adequate solution? What has to be done if a program does not produce a useful result?

Chapter 14 deals with nonlinear algebraic equations. The properties of these equations may differ greatly, which makes it difficult to solve them with black box software. Chapter 15 is devoted to a very special type of nonlinear equations: algebraic eigenproblems. There is a multitude of numerical software available for these problems. Again, this book deals primarily with LAPACK programs.

Chapter 16 takes a closer look at the topics found in previous chapters, especially problems with large and sparse matrices. Problems of this type are not covered using black box software; a solution method has to be selected individually for each of them. Accordingly, this chapter gives hints on how to select appropriate algorithms and programs.

Monte Carlo methods, which are important for solving numerical problems as well as for empirical sensitivity studies, are based on random numbers. The last chapter of the book, therefore, gives a short introduction into the world of random numbers and random number generators.

## Acknowledgments

Many people have helped me in the process of writing this book. I want to thank all of them cordially.

Arnold Krommer, with whom I have worked closely for many years, has had a very strong influence on the greater part of this book. Arnold's expertise and



knowledge has particularly enhanced the chapter on numerical integration, a field of interest to both of us.

Roman Augustyn and Wilfried Gansterer have contributed significantly to the chapter on computer hardware.

Winfried Auzinger and Hans Stetter reviewed parts of early manuscripts. Their comments have helped to shape the final version.

Many students at the Technical University of Vienna have been of great assistance. Their suggestions and their help with the proofreading contributed in turning my lecture notes on numerical data processing and a subsequent rough draft into a manuscript. In particular the contributions of Bernhard Bodenstorfer, Ernst Haunschmid, Michael Karg and Robert Matzinger have improved and enriched the content of this book. I wish to express my thanks to all of them.

Wolfgang Moser contributed considerably to the process of translating the German version of this book. He organized the technical assistance of many students who produced a rough draft of the English version. He also translated several chapters himself.

Yolanda Goss was a competent and helpful collaborator in linguistic matters. She brought the rough draft of the English manuscript a giant step nearer to the final version. We even managed to squeeze some fun out of the dry business of improving the grammar, style and form of the draft version.

Robert Lynch (Purdue University) has made a great many valuable suggestions for improving the comprehensibility and the formulation of several chapters.

Robert Bursill (University of Sheffield) and Ronnie Sircar (Stanford University) have proofread and revised the manuscript competently.

Jeremy Du Croz (NAG Ltd.) and Richard Hanson (Visual Numerics Inc.) have checked the software sections and made numerous suggestions for improvements.

Christoph Schmid and Thomas Wihan deserve particular appreciation for the appealing text lay out and the design of the diagrams and illustrations. Christoph Schmid created the final L<sup>A</sup>T<sub>E</sub>X printout with remarkable dedication.

In addition, I would like to acknowledge the financial support of the Austrian Science Foundation.

January 1997

CHRISTOPH W. UEBERHUBER

# Contents

<b>1</b>	<b>Scientific Modeling</b>	<b>1</b>
1.1	Reality Versus Model	1
1.2	The Model Subject and the Model	3
1.2.1	The Purpose of a Model	3
1.3	The Model Subject and Reality	4
1.4	Model Building	5
1.4.1	Problem Specification	5
1.4.2	Creating a Structural Concept	5
1.4.3	Choosing and Designing a Model	6
1.4.4	Establishing Parameter Values	8
1.4.5	Tests and Validation	8
<b>2</b>	<b>Fundamental Principles of Numerical Methods</b>	<b>9</b>
2.1	From Application Problems to their Numerical Solution	9
2.1.1	Case Study: Pendulum	10
2.1.2	Qualitative and Quantitative Problems	14
2.2	Numerical Problems	16
2.2.1	Numerical Problems	19
2.2.2	Categories of Numerical Problems	21
2.2.3	The Accuracy of Numerical Results	23
2.3	Types of Errors in Numerics	28
2.3.1	Model Errors	29
2.3.2	Data Errors	29
2.3.3	Algorithm Errors	30
2.3.4	Rounding Errors	34
2.3.5	Error Hierarchy	37
2.4	The Condition of Mathematical Problems	39
2.4.1	Perturbed and Unperturbed Problems	39
2.4.2	The Absolute Condition Number	39
2.4.3	The Relative Condition Number	40
2.4.4	Narrowing the Problem Class	42
2.4.5	Calculating Condition Numbers Using Differentiation	42
2.4.6	Case Study: Quadratic Equation	43
2.4.7	Ill-Conditioned Problems	44
2.4.8	Ill-Posed Problems	45
2.5	The Condition of Application Problems	46
2.6	The Mathematical Elements of Condition Estimation	48
2.6.1	The Condition of Direct Mathematical Problems	48
2.6.2	The Condition of Inverse Mathematical Problems	54
2.7	Validation of Numerical Computations	58

2.7.1	The Uncertainty of Numerical Computations . . . . .	58
2.7.2	Validation of Mathematical Models . . . . .	59
2.7.3	Sensitivity Analysis and Error Estimation . . . . .	60
2.7.4	Validation of Numerical Software . . . . .	65
<b>3</b>	<b>Computers for Numerical Data Processing . . . . .</b>	<b>68</b>
3.1	Processors . . . . .	70
3.1.1	Pipelining . . . . .	71
3.1.2	Superpipeline Architecture . . . . .	75
3.1.3	Superscalar Architectures . . . . .	75
3.1.4	Vector Processors . . . . .	77
3.2	Memory . . . . .	78
3.2.1	Memory Hierarchy . . . . .	79
3.2.2	Addressing Schemes . . . . .	81
3.2.3	Registers . . . . .	82
3.2.4	Cache Memory . . . . .	83
3.2.5	Virtual Memory . . . . .	88
3.2.6	Interleaved Memory . . . . .	89
3.3	Performance Quantification . . . . .	90
3.3.1	The Notion of Performance (Power) . . . . .	91
3.3.2	Time as a Factor in Performance . . . . .	92
3.4	Analytical Performance Assessment . . . . .	94
3.4.1	Peak Performance . . . . .	94
3.4.2	Instruction Performance . . . . .	95
3.4.3	Performance of Vector Processors . . . . .	97
3.4.4	The Influence of Memory on Performance . . . . .	100
3.5	Empirical Performance Assessment . . . . .	101
3.5.1	Temporal Performance . . . . .	101
3.5.2	Empirical Instruction Performance . . . . .	101
3.5.3	Empirical Floating-Point Performance . . . . .	102
3.5.4	Empirical Performance of Vector Processors . . . . .	104
<b>4</b>	<b>Numerical Data and Numerical Operations . . . . .</b>	<b>106</b>
4.1	Mathematical Data . . . . .	106
4.1.1	Fundamental Mathematical Data . . . . .	106
4.1.2	Algebraic Data . . . . .	106
4.1.3	Analytic Data . . . . .	107
4.2	Numerical Data on Computers . . . . .	108
4.2.1	Fundamental Numerical Data . . . . .	108
4.2.2	Algebraic Data . . . . .	110
4.2.3	Analytic Data . . . . .	110
4.2.4	Numerical Data Types . . . . .	110
4.3	Operations on Numerical Data . . . . .	111
4.3.1	Arithmetic Operations . . . . .	111
4.3.2	Algebraic Operations . . . . .	112

4.3.3	Array Processing in Fortran 90 . . . . .	113
4.3.4	Analytic Operations . . . . .	116
4.4	Number Systems on Computers . . . . .	117
4.4.1	INTEGER Number Systems . . . . .	118
4.4.2	Fixed-Point Number Systems . . . . .	120
4.4.3	Floating-Point Number Systems . . . . .	122
4.5	Structure of Floating-Point Systems . . . . .	126
4.5.1	The Number of Floating-Point Numbers . . . . .	126
4.5.2	Largest and Smallest Floating-Point Numbers . . . . .	127
4.5.3	Distances Between Floating-Point Numbers . . . . .	128
4.5.4	Relative Distances Between Floating-Point Numbers . . . . .	129
4.5.5	Case Study: $\mathbb{F}(10, 6, -9, 9, \text{true})$ . . . . .	131
4.6	Standardization of Floating-Point Number Systems . . . . .	132
4.6.1	The IEEE Standard for Floating-Point Numbers . . . . .	132
4.6.2	Hidden Bit . . . . .	134
4.6.3	Infinity, NaNs, and Signed Zero . . . . .	134
4.6.4	Impact on Programming Languages . . . . .	134
4.7	Arithmetics for Floating-Point Systems . . . . .	135
4.7.1	Rounding Functions . . . . .	136
4.7.2	Rounding Error . . . . .	139
4.7.3	Rounding and Arithmetic Operations . . . . .	142
4.7.4	Implementation of a Floating-Point Arithmetic . . . . .	146
4.7.5	Multiple-Precision Software . . . . .	147
4.8	Inquiry Functions and Manipulation of Numbers in Fortran 90 . . . . .	151
4.8.1	Parameters for Floating-Point Numbers . . . . .	151
4.8.2	Characteristics of Floating-Point Numbers . . . . .	152
4.8.3	The Distance Between Floating-Point Numbers, Rounding . . . . .	154
4.8.4	Manipulation of Floating-Point Numbers . . . . .	155
4.8.5	Parameters in INTEGER Numbers . . . . .	156
4.8.6	Case Study: Multiple Products . . . . .	156
4.9	Operations with Algebraic Data . . . . .	159
4.10	Operations with Arrays . . . . .	162
4.10.1	BLAS . . . . .	163
4.11	Operations with Analytic Data . . . . .	164
4.11.1	Representation of Functions . . . . .	164
4.11.2	Implementation of Functions . . . . .	165
4.11.3	Operations with Functions . . . . .	167
4.11.4	Functions as Results . . . . .	171
5	<b>Numerical Algorithms . . . . .</b>	<b>172</b>
5.1	The Intuitive Notion of an Algorithm . . . . .	172
5.2	Properties of Algorithms . . . . .	174
5.2.1	Abstraction . . . . .	175
5.2.2	Generality . . . . .	175

5.2.3	Finiteness . . . . .	175
5.2.4	Termination . . . . .	175
5.2.5	Deterministic Algorithms . . . . .	177
5.2.6	Determinate Algorithms . . . . .	177
5.3	Existence of Algorithms . . . . .	178
5.3.1	Precise Definitions of “Algorithm” . . . . .	179
5.3.2	Computable Functions . . . . .	179
5.4	Practical Solvability of Problems . . . . .	181
5.5	Complexity of Algorithms . . . . .	181
5.5.1	Abstract Computer Models . . . . .	183
5.5.2	Theoretical Execution Cost . . . . .	184
5.5.3	Asymptotic Complexity of Algorithms . . . . .	185
5.5.4	Problem Complexity . . . . .	186
5.5.5	Case Study: Multiplication of Matrices . . . . .	187
5.5.6	Practical Effort Determination . . . . .	191
5.6	Representation of Algorithms . . . . .	194
5.6.1	Fortran 90 . . . . .	194
5.6.2	Pseudocode . . . . .	195
5.7	Influence of Rounding Errors on Numerical Algorithms . . . . .	196
5.7.1	Arithmetic Algorithms . . . . .	196
5.7.2	Implementation of Arithmetic Algorithms . . . . .	199
5.7.3	Error Propagation . . . . .	200
5.7.4	Error Propagation Analysis . . . . .	201
5.8	Case Study: Floating-Point Summation . . . . .	210
5.8.1	Pairwise Summation . . . . .	212
5.8.2	Compensated Summation . . . . .	213
5.8.3	Comparison of Summation Methods . . . . .	214
5.8.4	Ascending Summation . . . . .	215
<b>6</b>	<b>Numerical Programs . . . . .</b>	<b>219</b>
6.1	The Quality of Numerical Programs . . . . .	219
6.1.1	Reliability . . . . .	219
6.1.2	Portability . . . . .	224
6.1.3	Efficiency . . . . .	226
6.2	Reasons for Poor Efficiency . . . . .	229
6.2.1	Lack of Parallelism . . . . .	229
6.2.2	Lack of Locality of Reference . . . . .	232
6.2.3	Inadequate Memory Access Patterns . . . . .	233
6.2.4	Overhead . . . . .	234
6.3	The Measurement of Performance Indices . . . . .	235
6.3.1	Measurement of the Workload . . . . .	236
6.3.2	Measurement of the Time . . . . .	238
6.3.3	Profiling . . . . .	243
6.3.4	Spotting Performance Decreasing Factors . . . . .	246
6.4	Performance Optimization . . . . .	247

6.5	Architecture Independent Optimizations . . . . .	248
6.6	Loop Optimizations . . . . .	249
6.6.1	Loop Unrolling . . . . .	249
6.6.2	Unrolling Nested Loops . . . . .	252
6.6.3	Loop Fusion . . . . .	253
6.6.4	The Elimination of Loop Branches . . . . .	254
6.6.5	Associative Transformations . . . . .	256
6.6.6	Loop Interchanges . . . . .	257
6.7	Blocked Memory Access . . . . .	258
6.7.1	Hierarchical Blocking . . . . .	261
6.8	Case Study: Multiplication of Matrices . . . . .	262
6.8.1	Loop Interchanges . . . . .	263
6.8.2	Loop Unrolling . . . . .	264
6.8.3	Blocking . . . . .	266
6.8.4	Block Copying . . . . .	267
6.8.5	Blocking, Copying and Loop Unrolling . . . . .	269
6.8.6	Optimizing System Software . . . . .	269
<b>7</b>	<b>Available Numerical Software . . . . .</b>	<b>272</b>
7.1	The Cost of Software . . . . .	272
7.2	Sources of Numerical Software . . . . .	274
7.2.1	Numerical Application Software . . . . .	274
7.2.2	Printed Programs . . . . .	275
7.2.3	Numerical Software Libraries . . . . .	276
7.2.4	Numerical Software Packages . . . . .	278
7.2.5	References to Software in this Book . . . . .	280
7.3	Software and the Internet . . . . .	281
7.3.1	The Internet . . . . .	282
7.3.2	Communication in the Internet, E-Mail . . . . .	284
7.3.3	Forums for Discussion on the Internet . . . . .	285
7.3.4	Resource Sharing on the Internet . . . . .	285
7.3.5	Finding Software in the Internet . . . . .	291
7.3.6	Internet Front Ends . . . . .	292
7.3.7	NETLIB . . . . .	293
7.3.8	eLib . . . . .	295
7.3.9	GAMS . . . . .	295
7.4	Interactive Multifunctional Systems . . . . .	295
7.4.1	Exploratory Systems . . . . .	295
7.4.2	Numerical Systems . . . . .	296
7.4.3	Symbolic Systems . . . . .	297
7.4.4	Simulation Systems . . . . .	297
7.5	Problem Solving Environments . . . . .	298
7.5.1	Available Problem Solving Environments . . . . .	299
7.6	Case Study: Software for Elliptic PDEs . . . . .	300
7.6.1	Problem Classification . . . . .	301

7.6.2	Software Packages for Elliptic Problems . . . . .	302
7.6.3	Numerical Program Library Sections . . . . .	305
7.6.4	TOMS Programs . . . . .	306
<b>8</b>	<b>Using Approximation in Mathematical Model Building . . . . .</b>	<b>308</b>
8.1	Analytic Models . . . . .	308
8.1.1	Elementary Functions as Models . . . . .	309
8.1.2	Algorithms Used as Mathematical Models . . . . .	311
8.2	Information and Data . . . . .	312
8.2.1	Obtaining Algebraic Data from Discrete Information . . . . .	312
8.2.2	Obtaining Analytic Data from Continuous Information . . . . .	312
8.2.3	The Discretization of Continuous Information . . . . .	314
8.2.4	The Homogenization of Discrete Data . . . . .	316
8.3	Discrete Approximation . . . . .	317
8.3.1	Discrete Approximation Data . . . . .	317
8.3.2	Interpolation . . . . .	319
8.3.3	Case Study: Water Resistance of Vessels . . . . .	319
8.4	Function Approximation . . . . .	321
8.4.1	Nonadaptive Discretization . . . . .	322
8.4.2	Adaptive Discretization . . . . .	322
8.4.3	Homogenization Algorithms . . . . .	323
8.4.4	Additional Information . . . . .	325
8.5	Choosing a Model Function . . . . .	326
8.5.1	Uniform or Piecewise Approximation . . . . .	326
8.5.2	Linear Approximation . . . . .	327
8.5.3	Nonlinear Approximation . . . . .	329
8.5.4	Global or Local Approximation . . . . .	330
8.5.5	Continuity and Differentiability . . . . .	332
8.5.6	Condition . . . . .	333
8.5.7	Invariance under Scaling . . . . .	334
8.5.8	Constraints . . . . .	335
8.5.9	Appearance . . . . .	337
8.6	Choice of the Distance Function . . . . .	339
8.6.1	Mathematical Foundations . . . . .	339
8.6.2	Norms for Finite-Dimensional Spaces . . . . .	341
8.6.3	Norms for Infinite-Dimensional Spaces . . . . .	343
8.6.4	Weighted Norms . . . . .	343
8.6.5	Hamming Distance . . . . .	344
8.6.6	Robust Distance Functions . . . . .	344
8.6.7	Orthogonal Approximation . . . . .	347
8.7	Transformation of the Problem . . . . .	350
8.7.1	Ordinate Transformation . . . . .	350
8.7.2	Curves . . . . .	351

<b>9 Interpolation</b>	<b>353</b>
9.1 Interpolation Problems	353
9.1.1 Choosing a Class of Functions	354
9.1.2 Determination of the Parameters of an Interpolation Function	354
9.1.3 Manipulation of the Interpolation Function	355
9.2 Mathematical Foundations	356
9.2.1 The General Interpolation Problem	356
9.2.2 Interpolation of Function Values	360
9.3 Univariate Polynomial Interpolation	361
9.3.1 Univariate Polynomials	361
9.3.2 Representation Forms of Univariate Polynomials	362
9.3.3 The Calculation of Coefficients	367
9.3.4 The Evaluation of Polynomials	369
9.3.5 Error and Convergence Behavior	373
9.3.6 Algorithm Error in Polynomial Interpolation	374
9.3.7 The Convergence of Interpolation Polynomials	377
9.3.8 The Conditioning of Polynomial Interpolation	386
9.3.9 Choosing Interpolation Nodes	390
9.3.10 Hermite Interpolation	393
9.4 Univariate, Piecewise, Polynomial Interpolation	394
9.4.1 The Accuracy of the Approximation	396
9.4.2 Evaluation	397
9.5 Polynomial Splines	397
9.5.1 Oscillations and Sensitivity to Perturbations	399
9.5.2 The Representation of Polynomial Splines	401
9.6 B-Splines	403
9.6.1 Choice of the B-Spline Break Points	406
9.6.2 B-Splines in Graphical Data Processing	407
9.6.3 Software for B-Splines	411
9.7 Cubic Spline Interpolation	412
9.7.1 Boundary Conditions	413
9.7.2 Extremal Property	414
9.7.3 Error and Convergence Behavior	415
9.7.4 The Calculation of Coefficients	415
9.7.5 The Evaluation of Spline Functions	416
9.7.6 Condition	416
9.8 Splines Without Undesirable Oscillations	419
9.8.1 Exponential Splines	421
9.8.2 $\nu$ -Splines	422
9.8.3 The Akima Sub spline Interpolation	426
9.9 Multivariate Interpolation	429
9.9.1 Tensor Product Interpolation	430
9.9.2 Triangulation	431
9.10 Multivariate Polynomial Interpolation	433



9.11 Multivariate (Sub-) Spline Interpolation . . . . . 435

9.11.1 Tensor Product Spline Functions . . . . . 437

9.11.2 Polynomial Interpolation on Triangles . . . . . 437

9.12 Related Problems and Methods . . . . . 438

**Glossary of Notation . . . . . 440**

**Bibliography . . . . . 443**

**Author Index . . . . . 460**

**Subject Index . . . . . 466**