

# Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

120

---

Louis B. Rall

Automatic Differentiation:  
Techniques and Applications

---



Springer-Verlag  
Berlin Heidelberg New York 1981

**Editorial Board**

W. Brauer P. Brinch Hansen D. Gries C. Moler G. Seegmüller  
J. Stoer N. Wirth

**Author**

Louis B. Rall  
University of Wisconsin-Madison, Mathematics Research Center  
610 Walnut Street, Madison, Wisconsin 53706, USA

AMS Subject Classifications (1980): 68-02, 68C20, 65D30, 65G10,  
65H10, 65K10

CR Subject Classifications (1981): 1.1, 5.1, 5.11, 5.15, 5.16

ISBN 3-540-10861-0 Springer-Verlag Berlin Heidelberg New York

ISBN 0-387-10861-0 Springer-Verlag New York Heidelberg Berlin

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© by Springer-Verlag Berlin Heidelberg 1981  
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.  
2145/3140-543210

to FRAN

## PREFACE

This book is based on the notes for a series of lectures given at the Computer Science Department (Datalogisk Institut) of the University of Copenhagen in the second semester of the 1979-80 academic year. The invitation of Dr. Ole Caprani of that institution to present these lectures, as well as his assistance with the course, is gratefully acknowledged. One of the students, Mr. J. W. Olesen, is also thanked for doing the necessary work to make software from the University of Wisconsin-Madison operational at the University of Copenhagen.

The automatic differentiation of functions defined by formulas proceeds by fixed rules, and is conceptually no more difficult than the translation of formulas into code for evaluation. In spite of this, the automatic calculation of derivatives and coefficients of power series has seemed somewhat exotic to numerical analysts, and perhaps too mundane to computer scientists interested in the creation of ever better languages and systems for computation. The purpose of these notes is to fill this intellectual gap, and show that a powerful computational tool can be fashioned without excessive effort.

The choice of topics presented is dictated by personal interest and familiarity with software which actually works, programs which have proved to be durable as well as effective. On the basis of ideas suggested by R. E. Moore, work was begun at the Mathematics Research Center by Allen Reiter in 1964-65 on software for differentiation, generation of Taylor coefficients, and interval arithmetic. This led to inter-related developments in programs for the solution of differential equations, nonlinear systems of equations, numerical integration, interval arithmetic, and a precompiler for the addition of new data types to FORTRAN. (The connection with FORTRAN is one of the reasons for the durability of this software.) This period of activity came to an end in 1977-78 with the departure of Julia Gray, F. Crary, G. Kedem, and J. M. Yohe from the Mathematics Research Center. Significant contributions were made along the way by J. A. Braun, D. Kuba, T. Ladner, T. Szymanski, and H. J. Wertz, among others. The support of the U. S. Army Research Office during the entire period of the development of this software is appreciated.

It is not implied that the subject of these lectures is a closed book; rather, it is an open door for future developments. To this end, each topic has been provided with suggestions for projects ranging from simple exercises to the construction of elaborate computational systems.

The production of these notes was assisted by Carol Gubbins, who did a professional job of preparation of the figures. First and foremost, thanks are due to my wife Fran for untiring patience, support, and help with every step of this project from beginning to end.

Madison, Wisconsin: May, 1981

## TABLE OF CONTENTS

### PREFACE

CHAPTER I. INTRODUCTION . . . . .	1
CHAPTER II. FORMULA TRANSLATION . . . . .	3
1. Function Evaluation . . . . .	4
2. The Kantorovich Graph of a Codeable Function . . . . .	6
CHAPTER III. FORMULA DIFFERENTIATION . . . . .	9
1. Rules for Differentiation . . . . .	9
2. Differentiation of Code Lists . . . . .	14
3. Nomenclature for Code Lists . . . . .	27
4. Projects for Chapter 3 . . . . .	34
CHAPTER IV. GENERATION OF TAYLOR COEFFICIENTS . . . . .	35
1. Subroutine Call Lists . . . . .	37
2. Recursion Formulas for Taylor Coefficients . . . . .	39
3. Exponentiation with One Constant Argument . . . . .	46
4. Projects for Chapter 4 . . . . .	52
CHAPTER V. EXAMPLES OF SOFTWARE FOR AUTOMATIC DIFFERENTIATION AND GENERATION OF TAYLOR COEFFICIENTS . . . . .	54
1. CODEX and SUPER-CODEX . . . . .	57
1.1. The Coder . . . . .	59
1.2. The Differentiator . . . . .	61
1.3. Other CODEX Subroutines: ASSIGN, EVAL, and PRINT . . . . .	80
1.4. Features of SUPER-CODEX . . . . .	81
2. TAYLOR and TAYLOR-GRADIENT . . . . .	87
3. Projects for Chapter 5 . . . . .	90
CHAPTER VI. AUTOMATIC COMPUTATION OF GRADIENTS, JACOBIANS, HESSIANS, AND APPLICATIONS TO OPTIMIZATION . . . . .	91
1. Gradient Vectors and Code Lists . . . . .	91
2. Gradients and Optimization Problems . . . . .	94
3. Jacobians and Newton's Method . . . . .	98
4. Second Derivatives: Hessian Matrices and Operators . . . . .	102
5. Projects for Chapter 6 . . . . .	111
CHAPTER VII. AUTOMATIC ERROR ANALYSIS . . . . .	112
1. Errors in Computation . . . . .	112
2. Interval Arithmetic . . . . .	113
3. Automatic Computation of Lipschitz Constants . . . . .	118
4. Use of Differentials in Sensitivity and Error Analysis . . . . .	120

# VIII

5. Projects for Chapter 7 . . . . .	121
CHAPTER VIII. SOLUTION OF NONLINEAR SYSTEMS OF EQUATIONS . . . . .	122
1. Simple Iteration and the Automatic Contraction Mapping Theorem . .	122
2. Newton's Method and the Automatic Kantorovich Theorem . . . . .	125
3. Interval Versions of Newton's Method and the Automatic Theorems of Nickel and Moore . . . . .	128
4. The Program NEWTON . . . . .	132
5. Some Methods for Finding Initial Approximations . . . . .	134
6. Projects for Chapter 8 . . . . .	135
CHAPTER IX. NUMERICAL INTEGRATION WITH RIGOROUS ERROR ESTIMATION . . .	136
1. Notation . . . . .	136
2. Numerical Integration of Systems of Ordinary Differential Equations	137
3. Numerical Integration . . . . .	138
4. The Program INTE . . . . .	142
5. Projects for Chapter 9 . . . . .	150
CHAPTER X. ADDITIONAL NOTES ON APPLICATIONS, SOFTWARE, AND TECHNIQUES . .	152
1. Generation of Taylor Coefficients . . . . .	152
2. Straightforward Differentiators . . . . .	153
3. Symbolic Algebraic Manipulators . . . . .	154
REFERENCES . . . . .	155
NAME INDEX . . . . .	159
SUBJECT INDEX . . . . .	160