

Jürgen Gerhard

LNC3 3218

# **Modular Algorithms in Symbolic Summation and Symbolic Integration**



Springer

TP301.6  
G368

Jürgen Gerhard

# Modular Algorithms in Symbolic Summation and Symbolic Integration



E200500352



Springer

Author

Jürgen Gerhard

Maplesoft

615 Kumpf Drive, Waterloo, ON, N2V 1K8, Canada

E-mail: Gerhard.Juergen@web.de

This work was accepted as PhD thesis on July 13, 2001, at

Fachbereich Mathematik und Informatik

Universität Paderborn

33095 Paderborn, Germany

Library of Congress Control Number: 2004115730

CR Subject Classification (1998): F.2.1, G.1, I.1

ISSN 0302-9743

ISBN 3-540-24061-6 Springer 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, re-use of illustrations, recitation, broadcasting, reproduction on microfilms 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. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign

Printed on acid-free paper SPIN: 11362159 06/3142 5 4 3 2 1 0

To Herbert Gerhard (1921–1999)

## Foreword

This work brings together two streams in computer algebra: symbolic integration and summation on the one hand, and fast algorithmics on the other hand.

In many algorithmically oriented areas of computer science, the *analysis of algorithms* – placed into the limelight by Don Knuth’s talk at the 1970 ICM – provides a crystal-clear criterion for success. The researcher who designs an algorithm that is faster (asymptotically, in the worst case) than any previous method receives instant gratification: her result will be recognized as valuable. Alas, the downside is that such results come along quite infrequently, despite our best efforts.

An alternative evaluation method is to run a new algorithm on examples; this has its obvious problems, but is sometimes the best we can do. George Collins, one of the fathers of computer algebra and a great experimenter, wrote in 1969: “I think this demonstrates again that a simple analysis is often more revealing than a ream of empirical data (although both are important).”

Within computer algebra, some areas have traditionally followed the former methodology, notably some parts of polynomial algebra and linear algebra. Other areas, such as polynomial system solving, have not yet been amenable to this approach. The usual “input size” parameters of computer science seem inadequate, and although some natural “geometric” parameters have been identified (solution dimension, regularity), not all (potential) major progress can be expressed in this framework.

Symbolic integration and summation have been in a similar state. There are some algorithms with analyzed run time, but basically the mathematically oriented world of integration and summation and the computer science world of algorithm analysis did not have much to say to each other.

Gerhard’s progress, presented in this work, is threefold:

- a clear framework for algorithm analysis with the appropriate parameters,
- the introduction of modular techniques into this area,
- almost optimal algorithms for the basic problems.

One might say that the first two steps are not new. Indeed, the basic algorithms and their parameters – in particular, the one called dispersion in Gerhard’s work – have been around for a while, and modular algorithms are a staple of computer algebra. But their combination is novel and leads to new perspectives, the almost optimal methods among them.

A fundamental requirement in modular algorithms is that the (solution modulo  $p$ ) of the problem equal the solution of the (problem modulo  $p$ ). This is generally not valid for all  $p$ , and a first task is to find a nonzero integer “resultant”  $r$  so that the requirement is satisfied for all primes  $p$  not dividing  $r$ . Furthermore,  $r$  has to be “small”, and one needs a bound on potential solutions, in order to limit the size and number of the primes  $p$  required. These tasks tend to be the major technical obstacles; the development of a modular algorithm is then usually straightforward. However, in order to achieve the truly efficient results of this work, one needs a thorough understanding of the relevant algorithmics, plus a lot of tricks and shortcuts.

The integration task is naturally defined via a limiting process, but the Old Masters like Leibniz, Bernoulli, Hermite, and Liouville already knew when to treat it as a symbolic problem. However, its formalization – mainly by Risch – in a purely algebraic setting successfully opened up perspectives for further progress. Now, modular differential calculus is useful in some contexts, and computer algebra researchers are aware of modular algorithms. But maybe the systematic approach as developed by Gerhard will also result in a paradigm shift in this field. If at all, this effect will not be visible at the “high end”, where new problem areas are being tamed by algorithmic approaches, but rather at the “low end” of reasonably domesticated questions, where new efficient methods will bring larger and larger problems to their knees.

It was a pleasure to supervise Jürgen’s Ph.D. thesis, presented here, and I am looking forward to the influence it may have on our science.

Paderborn, 9th June 2004

Joachim von zur Gathen

# Preface

What fascinated me most about my research in symbolic integration and symbolic summation were not only the strong parallels between the two areas, but also the differences. The most notable non-analogy is the existence of a polynomial-time algorithm for rational integration, but not for rational summation, manifested by such simple examples as  $1/(x^2 + mx)$ , whose indefinite sum with respect to  $x$  has the denominator  $x(x + 1)(x + 2) \cdots (x + m - 1)$  of exponential degree  $m$ , for all positive integers  $m$ . The fact that Moenck's (1977) straightforward adaption of Hermite's integration algorithm to rational summation is flawed, as discussed by Paule (1995), illustrates that the differences are intricate.

The idea for this research was born when Joachim von zur Gathen and I started the work on our textbook *Modern Computer Algebra* in 1997. Our goal was to give rigorous proofs and cost analyses for the fundamental algorithms in computer algebra. When we came to Chaps. 22 and 23, about symbolic integration and symbolic summation, we realized that although there is no shortage of algorithms, only few authors had given cost analyses for their methods or tried to tune them using standard techniques such as modular computation or asymptotically fast arithmetic. The pioneers in this respect are Horowitz (1971), who analyzed a modular Hermite integration algorithm in terms of word operations, and Yun (1977a), who gave an asymptotically fast algorithm in terms of arithmetic operations for the same problem. Chap. 6 in this book unites Horowitz's and Yun's approaches, resulting in two asymptotically fast and optimal modular Hermite integration algorithms. For modular hyperexponential integration and modular hypergeometric summation, this work gives the first complete cost analysis in terms of word operations.

**Acknowledgements.** I would like to thank:

My thesis advisor, Joachim von zur Gathen.

Katja Daubert, Michaela Huhn, Volker Strehl, and Luise Unger for their encouragement, without which this work probably would not have been finished.

My parents Johanna and Herbert, my sister Gisela, my brother Thomas, and their families for their love and their support.

My colleagues at Paderborn: the Research Group Algorithmic Mathematics, in particular Marianne Wehry, the MUPAD group,

in particular Benno Fuchssteiner, and SciFace Software, in particular Oliver Kluge.

My scientific colleagues all over the world for advice and inspiring discussions: Peter Bürgisser, Frédéric Chyzak, Winfried Fakler, Mark Giesbrecht, Karl-Heinz Kiyek, Dirk Kussin, Uwe Nagel, Christian Neliuss, Michael Nüsken, Walter Oevel, Peter Paule, Arne Storjohann, and Eugene Zima.

Waterloo, 23rd July 2004

Jürgen Gerhard

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Gooß, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

# Lecture Notes in Computer Science

For information about Vols. 1–3223

please contact your bookseller or Springer

- Vol. 3337: J.M. Barreiro, F. Martin-Sanchez, V. Maojo, F. Sanz (Eds.), *Biological and Medical Data Analysis. XI, 508 pages. 2004.*
- Vol. 3333: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), *Advances in Multimedia Information Processing - PCM 2004, Part III. XXXV, 785 pages. 2004.*
- Vol. 3332: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), *Advances in Multimedia Information Processing - PCM 2004, Part II. XXXVI, 1051 pages. 2004.*
- Vol. 3331: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), *Advances in Multimedia Information Processing - PCM 2004, Part I. XXXVI, 667 pages. 2004.*
- Vol. 3223: G. Antoniou, H. Boley (Eds.), *Rules and Rule Markup Languages for the Semantic Web. X, 215 pages. 2004.*
- Vol. 3222: R. Klette, J. Žunić (Eds.), *Combinatorial Image Analysis. XII, 760 pages. 2004.*
- Vol. 3316: N.R. Pal, N.K. Kasabov, R.K. Mudi, S. Pal, S.K. Parui (Eds.), *Neural Information Processing. XXX, 1368 pages. 2004.*
- Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), *Advances in Artificial Intelligence - IBERAMIA 2004. XX, 987 pages. 2004. (Subseries LNAI).*
- Vol. 3312: A.J. Hu, A.K. Martin (Eds.), *Formal Methods in Computer-Aided Design. XI, 445 pages. 2004.*
- Vol. 3311: V. Roca, F. Rousseau (Eds.), *Interactive Multimedia and Next Generation Networks. XIII, 287 pages. 2004.*
- Vol. 3309: C.-H. Chi, K.-Y. Lam (Eds.), *Content Computing. XII, 510 pages. 2004.*
- Vol. 3308: J. Davies, W. Schulte, M. Barnett (Eds.), *Formal Methods and Software Engineering. XIII, 500 pages. 2004.*
- Vol. 3307: C. Bussler, S.-k. Hong, W. Jun, R. Kaschek, Kinshuk, S. Krishnaswamy, S.W. Loke, D. Oberle, D. Richards, A. Sharma, Y. Sure, B. Thalheim (Eds.), *Web Information Systems - WISE 2004 Workshops. XV, 277 pages. 2004.*
- Vol. 3306: X. Zhou, S. Su, M.P. Papazoglou, M.E. Orłowska, K.G. Jeffery (Eds.), *Web Information Systems - WISE 2004. XVII, 745 pages. 2004.*
- Vol. 3305: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), *Cellular Automata. XV, 883 pages. 2004.*
- Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), *Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004. (Subseries LNAI).*
- Vol. 3302: W.-N. Chin (Ed.), *Programming Languages and Systems. XIII, 453 pages. 2004.*
- Vol. 3299: F. Wang (Ed.), *Automated Technology for Verification and Analysis. XII, 506 pages. 2004.*
- Vol. 3298: S.A. McIlraith, D. Plexousakis, F. van Harmelen (Eds.), *The Semantic Web - ISWC 2004. XXI, 841 pages. 2004.*
- Vol. 3295: P. Markopoulos, B. Eggen, E. Aarts, J.L. Crowley (Eds.), *Ambient Intelligence. XIII, 388 pages. 2004.*
- Vol. 3294: C.N. Dean, R.T. Boute (Eds.), *Teaching Formal Methods. X, 249 pages. 2004.*
- Vol. 3293: C.-H. Chi, M. van Steen, C. Wills (Eds.), *Web Content Caching and Distribution. IX, 283 pages. 2004.*
- Vol. 3292: R. Meersman, Z. Tari, A. Corsaro (Eds.), *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops. XXIII, 885 pages. 2004.*
- Vol. 3291: R. Meersman, Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part II. XXV, 824 pages. 2004.*
- Vol. 3290: R. Meersman, Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part I. XXV, 823 pages. 2004.*
- Vol. 3289: S. Wang, K. Tanaka, S. Zhou, T.W. Ling, J. Guan, D. Yang, F. Grandi, E. Mangina, I.-Y. Song, H.C. Mayr (Eds.), *Conceptual Modeling for Advanced Application Domains. XXII, 692 pages. 2004.*
- Vol. 3288: P. Atzeni, W. Chu, H. Lu, S. Zhou, T.W. Ling (Eds.), *Conceptual Modeling - ER 2004. XXI, 869 pages. 2004.*
- Vol. 3287: A. Sanfeliu, J.F. Martínez Trinidad, J.A. Carascas Ochoa (Eds.), *Progress in Pattern Recognition, Image Analysis and Applications. XVII, 703 pages. 2004.*
- Vol. 3286: G. Karsai, E. Visser (Eds.), *Generative Programming and Component Engineering. XIII, 491 pages. 2004.*
- Vol. 3285: S. Manandhar, J. Austin, U.B. Desai, Y. Oyanagi, A. Talukder (Eds.), *Applied Computing. XII, 334 pages. 2004.*
- Vol. 3284: A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), *Mobility Aware Technologies and Applications. XII, 382 pages. 2004.*
- Vol. 3283: F.A. Aagesen, C. Anutariya, V. Wuwongse (Eds.), *Intelligence in Communication Systems. XIII, 327 pages. 2004.*
- Vol. 3282: V. Guruswami, *List Decoding of Error-Correcting Codes. XIX, 350 pages. 2004.*
- Vol. 3281: T. Dingsøyr (Ed.), *Software Process Improvement. X, 207 pages. 2004.*
- Vol. 3280: C. Aykanat, T. Dayar, İ. Körpeoğlu (Eds.), *Computer and Information Sciences - ISCIS 2004. XVIII, 1009 pages. 2004.*
- Vol. 3278: A. Sahai, F. Wu (Eds.), *Utility Computing. XI, 272 pages. 2004.*

- Vol. 3274: R. Guerraoui (Ed.), Distributed Computing. XIII, 465 pages. 2004.
- Vol. 3273: T. Baar, A. Strohmeier, A. Moreira, S.J. Mellor (Eds.), <<UML>> 2004 - The Unified Modelling Language. XIII, 454 pages. 2004.
- Vol. 3271: J. Vicente, D. Hutchison (Eds.), Management of Multimedia Networks and Services. XIII, 335 pages. 2004.
- Vol. 3270: M. Jeckle, R. Kowalczyk, P. Braun (Eds.), Grid Services Engineering and Management. X, 165 pages. 2004.
- Vol. 3269: J. Lopez, S. Qing, E. Okamoto (Eds.), Information and Communications Security. XI, 564 pages. 2004.
- Vol. 3266: J. Solé-Pareta, M. Smirnov, P.V. Mieghem, J. Domingo-Pascual, E. Monteiro, P. Reichl, B. Stiller, R.J. Gibbins (Eds.), Quality of Service in the Emerging Networking Panorama. XVI, 390 pages. 2004.
- Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004. (Subseries LNAI).
- Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004. (Subseries LNAI).
- Vol. 3263: M. Weske, P. Liggesmeyer (Eds.), Object-Oriented and Internet-Based Technologies. XII, 239 pages. 2004.
- Vol. 3262: M.M. Freire, P. Chemouil, P. Lorenz, A. Gravey (Eds.), Universal Multiservice Networks. XIII, 556 pages. 2004.
- Vol. 3261: T. Yakhno (Ed.), Advances in Information Systems. XIV, 617 pages. 2004.
- Vol. 3260: I.G.M.M. Niemegeers, S.H. de Groot (Eds.), Personal Wireless Communications. XIV, 478 pages. 2004.
- Vol. 3258: M. Wallace (Ed.), Principles and Practice of Constraint Programming - CP 2004. XVII, 822 pages. 2004.
- Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004. (Subseries LNAI).
- Vol. 3256: H. Ehrig, G. Engels, F. Parisi-Presicce, G. Rozenberg (Eds.), Graph Transformations. XII, 451 pages. 2004.
- Vol. 3255: A. Benczúr, J. Demetrovics, G. Gottlob (Eds.), Advances in Databases and Information Systems. XI, 423 pages. 2004.
- Vol. 3254: E. Macii, V. Paliouras, O. Koufopavlou (Eds.), Integrated Circuit and System Design. XVI, 910 pages. 2004.
- Vol. 3253: Y. Lakhnech, S. Yovine (Eds.), Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. X, 397 pages. 2004.
- Vol. 3252: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004 Workshops. XVIII, 785 pages. 2004.
- Vol. 3251: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004. XXII, 1025 pages. 2004.
- Vol. 3250: L.-J. (LJ) Zhang, M. Jeckle (Eds.), Web Services. X, 301 pages. 2004.
- Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004. (Subseries LNAI).
- Vol. 3246: A. Apostolico, M. Melucci (Eds.), String Processing and Information Retrieval. XIV, 332 pages. 2004.
- Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004. (Subseries LNAI).
- Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004. (Subseries LNAI).
- Vol. 3243: S. Leonardi (Ed.), Algorithms and Models for the Web-Graph. VIII, 189 pages. 2004.
- Vol. 3242: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tiño, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII. XX, 1185 pages. 2004.
- Vol. 3241: D. Kranzlmüller, P. Kacsuk, J.J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XIII, 452 pages. 2004.
- Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004. (Subseries LNBI).
- Vol. 3239: G. Nicosia, V. Cutello, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 444 pages. 2004.
- Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004. (Subseries LNAI).
- Vol. 3237: C. Peters, J. Gonzalo, M. Bräschler, M. Kluck (Eds.), Comparative Evaluation of Multilingual Information Access Systems. XIV, 702 pages. 2004.
- Vol. 3236: M. Núñez, Z. Maamar, F.L. Pelayo, K. Pousttchi, F. Rubio (Eds.), Applying Formal Methods: Testing, Performance, and M/E-Commerce. XI, 381 pages. 2004.
- Vol. 3235: D. de Frutos-Escrig, M. Nunez (Eds.), Formal Techniques for Networked and Distributed Systems - FORTE 2004. X, 377 pages. 2004.
- Vol. 3234: M.J. Egenhofer, C. Freksa, H.J. Miller (Eds.), Geographic Information Science. VIII, 345 pages. 2004.
- Vol. 3233: K. Futatsugi, F. Mizoguchi, N. Yonezaki (Eds.), Software Security - Theories and Systems. X, 345 pages. 2004.
- Vol. 3232: R. Heery, L. Lyon (Eds.), Research and Advanced Technology for Digital Libraries. XV, 528 pages. 2004.
- Vol. 3231: H.-A. Jacobsen (Ed.), Middleware 2004. XV, 514 pages. 2004.
- Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004. (Subseries LNAI).
- Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004. (Subseries LNAI).
- Vol. 3226: M. Bouzeghoub, C. Goble, V. Kashyap, S. Spaccapietra (Eds.), Semantics of a Networked World. XIII, 326 pages. 2004.
- Vol. 3225: K. Zhang, Y. Zheng (Eds.), Information Security. XII, 442 pages. 2004.
- Vol. 3224: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection. XII, 315 pages. 2004.

# Table of Contents

|                                                                   |     |
|-------------------------------------------------------------------|-----|
| <b>1. Introduction</b>                                            | 1   |
| <b>2. Overview</b>                                                | 7   |
| 2.1 Outline                                                       | 12  |
| 2.2 Statement of Main Results                                     | 13  |
| 2.3 References and Related Works                                  | 21  |
| 2.4 Open Problems                                                 | 24  |
| <b>3. Technical Prerequisites</b>                                 | 27  |
| 3.1 Subresultants and the Euclidean Algorithm                     | 28  |
| 3.2 The Cost of Arithmetic                                        | 33  |
| <b>4. Change of Basis</b>                                         | 41  |
| 4.1 Computing Taylor Shifts                                       | 42  |
| 4.2 Conversion to Falling Factorials                              | 49  |
| 4.3 Fast Multiplication in the Falling Factorial Basis            | 57  |
| <b>5. Modular Squarefree and Greatest Factorial Factorization</b> | 61  |
| 5.1 Squarefree Factorization                                      | 61  |
| 5.2 Greatest Factorial Factorization                              | 68  |
| <b>6. Modular Hermite Integration</b>                             | 79  |
| 6.1 Small Primes Modular Algorithm                                | 80  |
| 6.2 Prime Power Modular Algorithm                                 | 85  |
| 6.3 Implementation                                                | 87  |
| <b>7. Computing All Integral Roots of the Resultant</b>           | 97  |
| 7.1 Application to Hypergeometric Summation                       | 103 |
| 7.2 Computing All Integral Roots Via Factoring                    | 109 |
| 7.3 Application to Hyperexponential Integration                   | 112 |
| 7.4 Modular LRT Algorithm                                         | 116 |
| <b>8. Modular Algorithms for the Gosper-Petkovšek Form</b>        | 121 |
| 8.1 Modular $GP'$ -Form Computation                               | 134 |

**9. Polynomial Solutions of Linear First Order Equations** ..... 149

    9.1 The Method of Undetermined Coefficients ..... 155

    9.2 Brent and Kung’s Algorithm for Linear Differential Equations ..... 158

    9.3 Rothstein’s SPDE Algorithm ..... 161

    9.4 The ABP Algorithm ..... 165

    9.5 A Divide-and-Conquer Algorithm: Generic Case ..... 169

    9.6 A Divide-and-Conquer Algorithm: General Case ..... 174

    9.7 Barkatou’s Algorithm for Linear Difference Equations ..... 179

    9.8 Modular Algorithms ..... 180

**10. Modular Gosper and Almkvist & Zeilberger Algorithms** ..... 195

    10.1 High Degree Examples ..... 198

**References** ..... 207

**Index** ..... 217

# List of Figures and Tables

|     |                                                                               |     |
|-----|-------------------------------------------------------------------------------|-----|
| 2.1 | Algorithm dependency graph . . . . .                                          | 14  |
| 3.1 | The $d$ th submatrix of the Sylvester matrix . . . . .                        | 29  |
| 4.1 | Running times for shift by 1 with classical arithmetic . . . . .              | 44  |
| 4.2 | Running times for Taylor shift with classical arithmetic . . . . .            | 45  |
| 4.3 | Running times for Taylor shift with classical arithmetic . . . . .            | 45  |
| 4.4 | Running times for Taylor shift with fast arithmetic . . . . .                 | 48  |
| 4.5 | Running times for Taylor shift with fast arithmetic . . . . .                 | 48  |
| 4.6 | Comparison of running times for Taylor shift . . . . .                        | 49  |
| 4.7 | Cost estimates for polynomial basis conversion . . . . .                      | 57  |
| 6.1 | Timings for the 1st series with small primes Hermite integration . . . . .    | 90  |
| 6.2 | Timings for the 2nd series with small primes Hermite integration . . . . .    | 91  |
| 6.3 | Timings for the 3rd series with small primes Hermite integration . . . . .    | 92  |
| 6.4 | Timings for the 1st series with prime power Hermite integration . . . . .     | 93  |
| 6.5 | Timings for the 2nd series with prime power Hermite integration . . . . .     | 94  |
| 6.6 | Timings for the 3rd series with prime power Hermite integration . . . . .     | 95  |
| 9.1 | The linear system $Lu = c$ with respect to the monomial basis. . . . .        | 166 |
| 9.2 | Cost estimates for polynomial solutions of first order equations . . . . .    | 192 |
| 9.3 | Cost estimates for modular algorithms solving first order equations . . . . . | 192 |

# List of Algorithms

|      |                                                                     |     |
|------|---------------------------------------------------------------------|-----|
| 3.4  | Monic Extended Euclidean Algorithm                                  | 29  |
| 4.7  | Small primes modular Taylor shift                                   | 48  |
| 4.18 | Modular conversion from $\mathcal{M}$ to $\mathcal{M}_{\mathbf{b}}$ | 55  |
| 4.19 | Modular conversion from $\mathcal{M}_{\mathbf{b}}$ to $\mathcal{M}$ | 55  |
| 4.23 | Evaluation in the falling factorial basis                           | 58  |
| 4.25 | Interpolation in the falling factorial basis                        | 58  |
| 5.2  | Yun's squarefree factorization                                      | 61  |
| 5.6  | Small primes modular squarefree factorization                       | 63  |
| 5.12 | Prime power modular squarefree factorization                        | 67  |
| 5.18 | Gff computation                                                     | 70  |
| 5.20 | Gff computation à la Yun                                            | 71  |
| 5.24 | Small primes modular gff computation                                | 74  |
| 6.4  | Small primes modular Hermite integration                            | 83  |
| 6.9  | Prime power modular Hermite integration                             | 85  |
| 7.2  | Modular bivariate EEA                                               | 98  |
| 7.12 | Prime power modular integral root distances                         | 104 |
| 7.20 | Prime power modular Man & Wright algorithm                          | 109 |
| 7.25 | Small primes modular integral residues                              | 113 |
| 7.29 | Small primes modular LRT algorithm                                  | 117 |
| 8.2  | Gosper-Petkovšek form                                               | 121 |
| 8.6  | Small primes modular shift gcd                                      | 123 |
| 8.12 | Small primes modular Gosper-Petkovšek form                          | 126 |
| 8.26 | GP'-refinement computation                                          | 137 |
| 8.28 | GP'-form computation                                                | 139 |
| 8.32 | Small primes modular GP'-refinement                                 | 140 |
| 8.37 | Prime power modular GP'-refinement                                  | 144 |
| 9.12 | Method of undetermined coefficients: part I                         | 155 |
| 9.15 | Method of undetermined coefficients: part II                        | 157 |
| 9.20 | Brent & Kung's algorithm                                            | 159 |
| 9.23 | Rothstein's SPDE algorithm                                          | 162 |
| 9.25 | Rothstein's SPDE algorithm for difference equations                 | 163 |
| 9.27 | ABP algorithm for first order differential equations                | 165 |
| 9.31 | ABP algorithm for first order difference equations                  | 168 |
| 9.35 | Difference divide & conquer: generic case                           | 171 |

|      |                                                   |     |
|------|---------------------------------------------------|-----|
| 9.38 | Differential divide & conquer: generic case ..... | 173 |
| 9.41 | Difference divide & conquer: general case .....   | 175 |
| 9.44 | Differential divide & conquer: general case ..... | 178 |
| 9.46 | Barkatou's algorithm .....                        | 179 |
| 9.56 | Modular Brent & Kung / ABP algorithm .....        | 183 |
| 9.60 | Modular Barkatou algorithm .....                  | 187 |
| 9.64 | Modular difference divide & conquer .....         | 189 |
| 9.67 | Modular differential divide & conquer .....       | 191 |
| 10.1 | Modular Gosper algorithm .....                    | 195 |
| 10.4 | Modular Almkvist & Zeilberger algorithm .....     | 196 |

# 1. Introduction

Modular algorithms are successfully employed in various areas of computer algebra, e.g., for factoring or computing greatest common divisors of polynomials or for solving systems of linear equations. They control the well-known phenomenon of intermediate expression swell, delay rational arithmetic until the very last step of the computation, and are often asymptotically faster than their non-modular counterparts.

At the time of writing, modular algorithms only rarely occur in computational differential and difference algebra. The goal of this work is to bring these worlds together. The main results are four modular algorithms for symbolic integration of rational and hyperexponential functions and symbolic summation of hypergeometric terms, together with a complete cost analysis. To our knowledge, this is the first time that a cost analysis is given at all in the case of hyperexponential integration and hypergeometric summation.

In the remainder of this introduction, we illustrate the main ideas with a rational integration example. The algorithm for integrating rational functions given in most undergraduate calculus textbooks decomposes the denominator into a product of linear – or at most quadratic – factors, performs a partial fraction decomposition, and then integrates the latter term by term. Consider the following rational function:

$$\frac{f}{g} = \frac{x^3 + 4x^2 + x - 1}{x^4 + x^3 - 4x^2 + x + 1} \in \mathbb{Q}(x).$$

The irreducible factorization of the denominator polynomial  $g$  over the real numbers is

$$g = (x - 1)^2 \left( x - \frac{3 + \sqrt{5}}{2} \right) \left( x - \frac{3 - \sqrt{5}}{2} \right),$$

and the partial fraction decomposition is

$$\frac{f}{g} = \frac{1}{(x - 1)^2} + \frac{7}{5} \cdot \frac{1}{x - 1} - \frac{1}{5} \cdot \frac{1}{x - \frac{3 + \sqrt{5}}{2}} - \frac{1}{5} \cdot \frac{1}{x - \frac{3 - \sqrt{5}}{2}}.$$

Thus the integral is

$$\int \frac{f}{g} = \left( -\frac{1}{x - 1} \right) + \frac{7}{5} \ln(x - 1) - \frac{1}{5} \ln \left( x - \frac{3 + \sqrt{5}}{2} \right) - \frac{1}{5} \ln \left( x - \frac{3 - \sqrt{5}}{2} \right).$$