James H. Davenport, Fritz Krückeberg,
Ramon E. Moore, Siegfried M. Rump (editors):

Symbolic, Algebraic and Validated Numerical Computation

Dagstuhl-Seminar-Report; 43
03.08.-07.08.92 (9232)

ISSN 0940-1121

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK

Schloß Dagstuhl

Seminar Report 9232

Symbolic, Algebraic and Validated Numerical Computation

August 3 - 7, 1992

OVERVIEW

The first Dagstuhl Seminar on *Symbolic, Algebraic and Validated Numerical Computation* was organized by J. H. Davenport, Bath, F. Krückeberg, Bonn, R. E. Moore, Columbus, and S. M. Rump, Hamburg. It brought together 32 participants from 5 countries, 7 participants came from overseas.

The 26 talks covered a wide range of topics of the three areas Computer Algebra, Validated Computation, and Numerical Computation. The aim of the seminar was to bring together experts of those three areas to discuss common interests.

Both Computer Algebra and Validated Computation aim on computing correct results on the computer. Here correct is to be understand in a mathematical sense including all model, discretization and rounding errors. Both approaches can benefit from Numerical Computation by validating an error bound for an approximation.

In the talks we saw some algorithms with result verification for finite dimensional as well as infinite dimensional problems, a promising global optimization algorithm, an interesting approach to analyze the sensitivity of algebraic problems, and hybrid algorithms combining two or even three of the main areas of the conference. Moreover, we saw a number of practical applications.

Everybody enjoyed the very pleasant atmosphere, the excellent food and the surroundings inviting to intensive discussions and recreational hiking.

We would like to express our thanks to all who contributed to the conference and to the administration of the Dagstuhl center for their excellent job.

Contents

RAMON MOORE

Parameter Identification with Bounded Error Data

MARKUS MÜLLER

A Hybrid Reduce-Fortran Finite Element Test Environment
(together with R. Lorentz)

M.T. NAKAO

Numerical Verifications of Solutions of Partial Differential Equations

ARNOLD NEUMAIER

Confidence regions, ellipsoid arithmetic, and the wrapping effect

M. POHST

On Computing in Algebraic Number Fields

SIEGFRIED RUMP

Symbolic, Algebraic and Validated Computation

MARK SCHAEFER

Precise Optimization using Range Arithmetic

H.J. STETTER

Sensitivity Analysis of Algebraic Algorithms

BERND TIBKEN

Simulation of Uncertain Discrete Systems

PETER WALERIUS

High Precision and Verified Computations of Pole Assignment

MICHAEL A. WOLFE

On Certain Computable Tests and Componentwise Error Bounds

Abstracts

Finding where n functions of n variables are simultaneously zero

by Oliver Aberth

First, a description was given of the goals and problems attempted of precise numerical analysis. Problems must first be stated so that they are effectively solvable, that is, have no intrinsic difficulties. The problem of solving $AX = B$, where A is an $n \times n$ matrix of constants, B is a column vector of constants, and X is a column vector of unknowns, was used as an example and converted into several suitable problems for precise numerical analysis.

Next the specific problem of finding where n functions of n variables are simultaneously zero was discussed. When $n = 1$, the sign change of $f(a)$ and $f(b)$ is helpful in detecting when a zero in $[a, b]$ is present. This can be generalized to the topological degree modul for general n. There is an interval arithmetic method of evaluating this for general n, and the method can be used to find the zeros of the n functions.

The Newton-Kantorowič Theorem and the Verification of Solutions

by G. Alefeld

Using error bounds which follow from the Newton-Kantorowič Theorem we device a method for bounding approximate solutions of nonlinear systems. From the last two iterates of the floating point Newton's-method we construct an interval-vector which is used to test for a solution. The idea has proved to be very successful.

Bracketing Frequencies of Vibrations of Turbine Blades in Large Turbo-Machines

by Henning Behnke

The natural bending vibrations of a free standing blade are concidered. The mathematical model describing this problem is an eigenvalue problem with a system of ordinary differential equations of fourth order. The eigenvalues depend on a real parameter Ω, the angular velocity. It is shown, how bounds for the eigenvalue-curves can be computed by means of the Rayhigh-Ritz and N.J. Lehmann procedures. Rounding errors are controlled rigorously by the use of interval arithmetic.

Generating Derivative Codes from Fortran Programs

by George F. Corliss

The numerical methods employed in the solution of many scientific computing problems require the computation of derivatives of a function $f : \mathbb{R}^n \to \mathbb{R}^m$. ADIFOR (Automatic Differentiation in FORtran) is a source transformation tool that accepts Fortran 77 code for the computation of f(x) and writes portable Fortran 77 code for the computation of $J(x)S$, where J is the matrix of first derivatives $J_{kl} = \frac{\partial f_k}{\partial x_l}$, and S is a matrix with n rows. The ADIFOR interface is very flexible. It allows the computation of the Jacobian itself ($S = I_{n \times n}$), the product of the Jacobian with a vector y ($S = y$), or the exploitation of known structure of the Jacobian. ADIFOR is the result of close cooperation between compiler writers and numerical analysisists to target real-life optimization and ODE problems. We introduce the principles behind ADIFOR, outline its functionality, and give examples of its use.

Some Symbolic-Numeric Case Studies

by J. H. Davenport

We present two examples of hybrid symbolic and numeric computation.

Stability Analysis

We take the problem of Dr. Tibken: the dynamical system

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} p_1 & -p_2 \\ p_2 & p_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

where $p_1, p_2 \in [0.4,\ 0.5]$. We can compute many iterations of this process symbolically, then use IRENA (the **I**nterface between **RE**duce and **NA**g), to call a numeric minimisation routine to find the (local) minimum of the x_i across the range of the p_j. This confirms the fact that the process actually converges.

Enzyme Kinetics

This was a presentation of a lengthy calculation, using the methods of Gröbner bases, numerical minimisation and least-squares fitting to estimate parameter values for a reasonably complex biochemical reaction from experimental data. This can be seen as one example of the problem considered by Prof. Moore, of finding parameter regions in the presence of uncertain data, though we did not approach the problem in precisely that way.

Introduction to Computer Algebra

by J.H. Davenport

In this talk, we survey computer algebra and its part in the wider task of "computation", noting that it is only since the advent of digital computers that "computation" has become identified in the minds of most with "numeric computation". Computer algebra has the following possibilities of interacting with numeric computation.

- It may replace numeric computation. A classic example of this is some extremely complex quadruple integrals related to on-chip capacitance, which were quite difficult to evaluate numerically, but which needed to be evaluated very often. A symbolic solution, albeit 400 lines long, was found to be much preferable, since evaluation was faster and more accurate.

- It may augment numerical computation, by various techniques which can always be used by hand, but may become too tedious in practice. These include symbolic differentiation; series expansion (generally combined with symbolic cancellation of dominant terms); expression rearrangement; and, particularly relevant to the theme of this conference, symbolic tracking of the relationship between different interval-valued variables.

- It may be used to analyse numerical computation. One particular area that is drawing a lot of attention at the moment is the stability analysis of difference schemes. Computer algebra can also be used to produce closed-form solutions to compare with numerical ones, or to produce test data or high-precision comparison data for conventional or interval methods.

Is Exploiting Partially Separable Structure Worthwhile?

by David M. Gay

As Griewank and Toint have pointed out, many optimization problems have a partially separable objective function, i.e., one of the form $f(x) = \sum_{i \in p} f_i(U_i x)$, where $x \in \mathbb{R}^n$ and U_i is a real $n_i \times n$ matrix with $n_i \ll n$. Backward Automatic Differentiation (AD) applied either to $d^T \nabla f$ or to $\nabla_x \varphi'(0)$, where $\varphi(\alpha) = f(x + \alpha d)$ [see Christianson's recent paper in SMA J. Num. Anal.] can deliver $(\nabla^2 f)d$ (i.e., Hessian of f times vector d) in a number of operations proportial to those needed to compute f. How does computing $(\nabla^2 f)d$ compare with the conventional technique of computing a quasi-Newton approximation H_i to $\nabla^2 f_i$ and computing $\sum_i (U_i^T H_i U_i)d$, both in a truncated Newton scheme?

Another way to exploit partially separable structure with AD is to compute the n_i columns of $\nabla^2 f_i$, then use either the explicit $\nabla^2 f = \sum_{i \in p} U_i^T \nabla^2 f_i U_i$ or $(\nabla^2 f)d = \sum_i U_i^T \nabla^2 f_i U_i d$. How do these alternatives compare with those above?

Initial computational results comparing just the finite difference $[\nabla f(x + hd) - \nabla f(x)]/h$ with a backward AD computation of $(\nabla^2 f)d$ show the finite difference to be somewhat faster. I had hoped by now to have more computational experience in this area, but have been waylaid by work on a forthcoming book on AMPL (a modeling language for mathematical programming, joint work with Bob Fourer and Brian Kernighan). This itself is a project relevant to this meeting, as the AMPL translator converts an algebraic description of an optimization problem into a symbolic representation — expression DAGs — that other programs, e.g., solvers, can manipulate.

Validated Computation with Sets of Hyperrectangles

by Karl-Udo Jahn

Due to the special shape of axis-parallel intervals during enclosure processes for solution sets overestimations may occur. That's why more general enclosure sets are of interest. These sets should have most of the useful properties of intervals (reprensentable by a few parameters; Minkowski operations, outwardly directed roundings, intersection as well as interval hull are effective performable; Hansdorff and maximum distances are computable, extensions of functions can be defined), and they should be useful to avoid the wrapping effect, to allow further operations and to approximate arbitrary closed bounded sets as close as one wishes. It can be shown that finite sets of intervals resp. their unions fulfill the above-mentioned requirements. Furthermore, Banach-like fixed point theorems are available.

A Validation Method for Global Optimization Problems

by Christian Jansson

A method for the global optimization problem with simple bound constraints is presented. The method is based on the tools of interval arithmetic and uses a special branch-and-bound technique in connection with a descent algorithm. Besides the calculation of approximations of the global minimum value and global minimum points in addition guaranteed lower and upper bounds of the global minimum value are computed. Derivatives of the function are not required. Numerical results on a large set of test functions are presented.

High Precision Calculation of Polynomial Complex Roots

by Werner Krandick

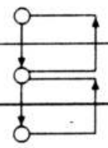
An algorithm is presented which calculates polynomial complex roots to specified precision. The method is infallible, i.e. unlike numerical algorithms. It will always terminate with correct results. Its average computing time seems to be cubic in the degree of the polynomial, and linear in the length of the polynomial coefficients. Preliminary experiments indicate a quadratic rate of convergence.

Applying the "Principle of the argument" from complex analysis a winding number computation is used to determine the number of polynomial roots in a rectangle. This method is applied under a bisection scheme in order to compute isolating rectangles for the complex roots of a polynomial, i.e. rectangles which contain exactly one root. Once a sufficiently small isolating rectangle has been found, a geometrical construction using tangents to certain algebraic curves guesses a (usually very small) subrectangle which is likely to contain the root. The (infallible) winding number computation is used to validate the guess; if the guess proves incorrect, bisection is used to obtain a subrectangle. This process continues until isolating rectangles for all roots have been found and refined as desired.

The general Structure of integrated Symbolic, Algebraic and Validated Numerical Computation

by Fritz Krückeberg

By an integration of Symbolic Computation, Computer Algebra and Numerical Computation we have the possibility of producing a new generation of powerful methods. These new methods can help to fulfill the needs of users, engineers and also the needs for verification and correctness. The general strategy: To get a new generation of methods by synthesis of Numerical Mathematics and Pure Mathematics by the computer. The paradigm of future oriented computing is to realize: Verifications, Flexibility and Efficiency. The general structure consists of three (or more) levels with a process-control backward and forward:

(3)	Symbolic Computation	Computer Algebra Systems	
(2)	Numerical Methods	Numerical Methods and Procedures	
(1)	Arithmetic Operations	(Interval)-Arithmetic with variable precision	
No.	⇑ The Levels	⇑ The Procedures	⇑ The Control

What we need is a deeper understanding (a theory) of
(3) how to find and select the best formal representation,
(2) how to find the best numerical method and how to control their parameters,
(1) how to control the word-length and how to select the arithmetic representation

Three Applications of Interval Methods for Computing Feasible Regions of Nonlinear Systems

by Weldon A. Lodwick

Interval methods for computing feasible regions of nonlinear systems associated with three applications were discussed. Work in progress and some results were presented. The applications were in the areas of processing models, such as refineries, radiation therapy of cancer tumors and constraint logic programming.

Validated Computations for Ordinary Differential Equations

by Rudolf Lohner

For different kinds of problems with ordinary differential equations methods are presented which can prove the existence of solutions and verify bounds by numerical computation with interval arithmetic. The basic type of problems are initial value problems (ivp) where Taylor expansions are used and their remainder terms are bounded by intervals. Boundary value problems (bvp) are reduced to ivp and finite dimensional equations by means of a multiple shooting method which is also executed in interval arithmetic. Also eigenvalue problems free boundary value problems and others can be handled by transforming them to standard bvp on a fixed interval. The methods have been programmed in PASCAL-XSC and examples are presented which show the applicability of the method (including chaotic solutions of the Lorenz equations and mildly stiff bvp).

A Hybrid Reduce-Fortran Finite Element Test Environment

by R.A. Lorentz & M. Müller

The basic idea of the above mentioned test environment is to allow one to determine whether or not a given multivariate interpolation (by multivariate polynomials) can be used for the numerical solution of self-adjoint partial differential equations via the finite element method. In particular, the user should only be required to formulate the interpolation after which the program carries out all the computations by itself.

The class of interpolating functionals which are supported include not only the usual function values, derivatives at points and normal derivatives at points, but also non-standard functionals. For example, mean values of functions or their derivatives over edges or regions, which have been used by Kergin and Hakopicen, are also included. The user of this test environment is, however, not restricted to using the implemented functionals since the environment has been concieved as an open system. It is extremely easy not only to include user-defined functionals but, e.g., to change the norms used to evaluate the error.

The computer algebra program REDUCE is used to determine whether the set of functionals specified together with the space of polynomials to be interpolated from define a regular interpolation. It also determines the form functions of interpolation and calculates the element stiffness matrix of the finite element discretization. It also generates Fortran code for these quantities and passes it to a numerical finite element solver. Finally, the quality of the discretization, which includes the size of the errors and the convergence rate, is determined and if desired, graphics are generated.

The test environment runs on all platforms which support REDUCE and Fortran. It can be obtained from either of the authors.

Standards and Algorithm Design for Floating Point Arithmetic

by David W. Matula

We describe how the IEEE standard provided closure and uniqueness to floating point arithmetic and established a more meaningful hierarchical precision environment. The standard adressed the basic single operations $\pm, \times, /, \sqrt{}$, but did not addres more complicated operations, such as the inner product. Following the commercially appropriate model established by the standard that error monitoring and well definedness will be added to available hardware provided the time and hardware resources required are only marginally greater than for unvalidated computation, we attempt to provide suitable validation for the inner product.

We characterize and describe a threshold for catastrophic concellation in an inner product. We describe variations in infinitely precise rounding including the existance of a back up mode, i.e. "round to nearest or otherwise round down" (RND). We introduce the notions of sticky accumulation in higher precision. For the particular example of quad (128 bit) sticky accumulation of double precision (64 bit) floating point numbers, we show a cost effective design for inner product that can detect catastrophic cancellation, otherwise in its absence will yield a validated precise RND rounding. We finally discuss a software model of the process which is fairly efficient given only the extra hardware instruments for low order parts of multiply and add.

Two-stage Interval Iterative Methods*)

by Günter Mayer

Two-stage methods are iterative methods to approximate solutions of linear systems $Ax = b$ with $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$. They are based on two splittings $A = M - N$ and $M = F - G$ which are used to construct the iteration

$\widehat{x}^0 := x^k$

for $i := 1$ to $s(k)$ solve $F\widehat{x}^i := G\widehat{x}^{i-1} + Nx^k + b$

$x^{k+1} := \widehat{x}^{s(k)}$

where $s(k) \in \mathbb{N}$. Two-stage *interval* methods are defined analogously for $n \times n$ interval matrices $[A]$ and interval splittings $[A] = [M] - [N]$ and $[M] = [F] - [G]$, where the loop above is replaced by

$$[\widehat{x}]^i := IGA([F], [G][\widehat{x}]^{i-1} + [N][x]^k + [b]).$$

Here, $IGA(.,.)$ denotes the vector which is generated by the interval Gaussian algorithm. Results are given for $s(k) \equiv s = \text{const}$ (stationary method) and for $s(k)$ being allowed to vary with the iteration index k (instationary method). These results refer to the global convergence of the method, to the speed of convergence, to the quality of enclosure of the solution set $L = \{ x \mid Ax = b, A \in [A], b \in [b] \}$ $([b] \in \mathbb{IR}^n)$ and to the nonsingularity of $A \in [A]$.

---

*) joint work with Andreas Frommer

Parameter Identification with Bounded Error Data

by Ramon Moore

The problem is to find the set of all parameter vectors for which a model agrees with data to within given bounds on errors in the data.

Using interval methods, the problem is now solved.

Numerical Verifications of Solutions of Partial Differential Equations

by Mitsuhiro T. Nakao

We consider a numerical technique to verify the solutions for nonlinear elliptic boundary value problems. Using a finite element solution and explicit error estimates for certain simple linear problems, we construct, on the computer, a set of functions which satisfies the condition of Schauder's or other fixed point theorem in the infinite dimensional space. In order to obtain such a set in numerical settings, the method of computable error estimation plays an essential role. Some numerical examples are presented.

The verification principle provided can also be applied for evolutional problems.

Confidence regions, ellipsoid arithmetic, and the Wrapping effect

by Arnold Neumaier

The wrapping effect is one of the main reasons that applications of interval arithmetic to the enclosure of dynamical systems is difficult. In this talk, the various sources of wrapping are analyzed. A new method for reducing the wrapping effect is proposed, based on ellipsoid arithmetic.

As an application, it is shown how one can propagate ellipsoidal confidence regions for stochastic parameters through arithmetical expressions without reducing the confidence level.

On computing in algebraic number fields

by Michael E. Pohst

We discuss some problems related to doing arithmetic in an algebraic number field F. The (integral) elements of F under consideration are supposed to belong to an order R of F. Then addition, subtraction and multiplication are immediate via the presentation of the elements of R by some fixed $\mathbf{Z}$-basis. However, division in R is already rather complicated and we discuss several distinct methods.

A major topic of computational algebraic number theory is the computation of the unit group $U(R)$ of R. We sketch a new method which essentially makes use of two algorithms from the geometry of numbers, namely

(i) LLL-reduction of lattice bases,

(ii) computation of short(est) vectors in a lattice.

Instead with varying lattices given by bases we use only the corresponding Gram matrices and add weights if appropriate. While accuracy is not required for some applications (where the result can be easily checked for correctness) we need to use validated numerical computations for proving that certain regions do not contain lattice points.

Symbolic, Algebraic and Validated Computation

by Siegfried M. Rump

Both Computer Algebra and Validated Computations aim on computing correct results on the computer, correctness to be understand in a mathematical sense including all model, discretization and rounding errors. Correct results may be given in terms of exact results (rational numbers, algebraic numbers etc.) or enclosing intervals with verified existence and possibly uniqueness of the solution within the computed bounds. There are many applications where the latter way of giving results is sufficient. This is a joint area both computer Algebra and Validation Algorithms may focus on.

Precise Optimization Using Range Arithmetic

by Mark J. Schaefer

We recently developed an implementation of range arithmetic in C++ to serve as the basis for writing numerical programs which compute reliable answers. This arithmetic is more complex than floating-point arithmetic and is a form of interval arithmetic. My talk will provide a brief introduction to range arithmetic and will focus on the adaptation of a Levenberg-Marquardt search algorithm for local minima to this arithmetic in order to guarantee convergence and generate correct results. A few numerical examples will be discussed as well.

Sensitivity Analysis of Algebraic Algorithms

by Hans J. Stetter

Nearly all mathematical problems permit degenerations, often of various degrees (rank deficiency, special positions, etc.). Solution algorithms for such problems should recognize near-degenerate situations and tell us about it: In applications with not fully accurate data we have to assume that the degeneracy exists, and in any case we should treat the problem as a perturbation of the degenerate one for better condition.

In good numerical floating-point algorithms, near degeneracies will show up automatically because a potential loss of accuracy must be checked. If rational arithmetic (without wordlength restriction) is used in algebraic algorithms the problem appears either as precisely degenerate or not at all, at least in present-day implementations is Computer Algebra systems.

In our presentation, we demonstrate that all algebraic algorithms should check for nearby degeneracies. In many cases this can be achieved (with appropriate scaling) by dropping sufficiently small quantities and doing an a posteriori analysis of the effect of these perturbations. After the precise algorithm has been thus modified, it can also be turned into floating-point versions by inclusion of the round-off effects into the perturbation analysis.

The following problem areas have been considered:

— Multivariate interpolation

— Symbolic integration of rational functions

— G.c.d. of two univariate polynomials

— Joint zeros of multivariate polynomials

It is claimed that algebraic algorithms which have been thus enhanced are more suitable for Scientific Computing.

Simulation of Uncertain Discrete Systems

by Bernd Tibken and Eberhard P. Hofer

The simulation problem for discrete time systems $x(k+1) = f(x(k), u(k), p, t)$; $y(k) = h(x(k), u(k), p, k)$ is discussed in the uncertain parameter p and initial condition $x(0)$ setting. The naive use of interval arithmetic is compared to a sophisticated approach which makes use of the global optimization algorithm by Hansen. Componentwise inclusions for the state x and the output y of the system are computed. The results will be published in the proceedings of an NASA conference held in June '92. A copy is available from the authors.

High precision and verified Computations of Pole Assignment

by Peter Walerius

The pole assignment, carried out by state variable feedback is a method for the synthesis of control systems. The presented method introduced here is based on an explicit formula for state variable feedback design in case of multivariable system. This formula yields from the theory of the Complete Modal Synthesis, which is transformed into a system of nonlinear equations. Using an inclusion procudure for the solution of nonlinear systems, the computation of the feedback matrix the pole placement problem can be carried out with high precision and verification.

On Certain Computable Tests and Componentwise Error Bounds

by Shen Zuhe & M. A. Wolfe

The use of interval slope arithmetic in Pandian's existence test and componentwise error bounds for solutions of nonlinear systems $f : D \subseteq \mathbb{R}^N \to \mathbb{R}^N$ with $f \in C^1(D)$ (SIAM J.N.A. 22 (1985)) is investigated. The results which are obtained are related to the results of Moore & Kioustelidis (SIAM J.N.A. 17 (1980)) and of Shen & Neumaier (Computing 40(1988)).

Dagstuhl-Seminar 9232: **List of Participants**

Oliver **Aberth**
Texas A&M University
Dept. of Mathematics
College Station TX 77843-3368
USA
tel.: +1-409-693-55 25

Götz **Alefeld**
Universität Karlsruhe
Institut für Angewandte Mathematik
Kaiserstr 12
W-7500 Karlsruhe 1
Germany
na.alefeld@na-net.standford.edu
tel.: +721-608-20 60

Henning **Behnke**
TU Clausthal
Institut für Mathematik
Erzstraße 1
W-3392 Clausthal-Zellerfeld
Germany
mahb@IBM.RZ.TU-CLAUSTHAL.DE
tel.: +5321-72-31 83

George **Corliss**
Department of Mathematics
Milwaukee WI 53233
USA
georgec@bris.mscs.mu.edu
tel.: +1-414-288-6599

James H. **Davenport**
University of Bath
School of Mathematics
Claverton Down
Bath BA2 7AY
Great Britain
jfd@maths.bath.ac.uk
tel.: +44-225-826181

David M. **Gay**
AT&T Bell Laboratories
Room 2C-463
600 Mountain Avenue
Murray Hill NJ 07974-0636
USA
dmg@research.att.com
tel.: +1-908-582-56 23

Karl-Udo **Jahn**
Technische Hochschule Leipzig
Sektion Mathematik/Informatik
Karl-Liebknecht-Straße 132
O-7030 Leipzig
Germany
tel.: +49-341-3928-482

Christian **Jansson**
TU Hamburg-Harburg
Technische Informatik III
Eißendorfer Str. 38
W-2000 Hamburg 90
Germany
jansson@tu-harburg.dbp.de
tel.: +40-7718-29 23

Fritz **Krückeberg**
Gesellschaft für Mathematik und
Datenverarbeitung mbH
Schloß Birlinghoven
Postfach 1316
W-5205 St. Augustin 1
Germany
krue@gmdzi.de
tel.: +2241-14-23 35 und 23 34

Werner **Krandick**
Johannes Kepler Universität
Institut für Mathematik
RISC
A-4040 Linz
Austria
krandick@risc.uni-linz.ac.at
tel.: +43-7236-3231-44

Weldon A. **Lodwick**
University of Colorado at Denver
Dept. of Mathematics
Campus Box 170
1100 Fourteenth Street
Denver CO 80217-3364
USA
wlodwick@copper.denver.colorado.edu
tel.: 1-303-556-8442

Rudolf **Lohner**
Universität Karlsruhe
Inst. f. Angewandte Mathematik
Kaiserstraße 12
W-7500 Karlsruhe 1
Germany
ae34@iamk4508.mathematik.uni-karlsruhe.de
tel.: +721-608-42 73

Rudolph **Lorentz**
Gesellschaft für Mathematik und
Datenverarbeitung mbH
Schloß Birlinghoven
Postfach 12 40
W-5205 St. Augustin 1
Germany
lorentz@.gmd.de
tel.: +2241-33 21 18

Günter **Ludyk**
Universität Bremen
Institut für Automatisierungstechnik
Postfach 33 04 40
W-2800 Bremen 33
Germany
tel.: +421-27 23 84

Markus **Müller**
GMD Schloß Birnghoven F1T
Institut für Methodische Grundlagen
Postfach 12 40
W-5205 Sankt Augustin 1
Germany
gmap51@f1ibmsv.gmd.de
tel.: +2241-14 27 54

David W. **Matula**
Southern Methodist Universiy
Depart. of Computer Science & Enigneering
Dallas TX 75275
USA
matula@seas.smu.edu
tel.: +1-214-692 3089

Günter **Mayer**
Universität Karlsruhe
Institut für Angewandte Mathematik
Kaiserstr 12
W-7500 Karlsruhe 1
Germany
ae09@dkauni2.bitnet
tel.: +721/608-26 83

Ramon **Moore**
Ohio State University
Depart. of Computer & Information Science
2036 Neil Avenue
Columbus OH 43210-1277
USA
moore-r@cis.ohio-state.edu
tel.: +1-614-846-0903

Mitsuhiro **Nakao**
Kyushu University 33
Dept. of Mathematics
Postal No. 812
Fukuoka
Japan
tel.: +81-092-641-1101-43 74

Arnold **Neumaier**
Universität Freiburg
Institut für Angewandte Mathematik
Hermann Herder Str. 10
W-7800 Freiburg i. Breisgau
Germany
neum@sun1.ruf.uni-freiburg.de
tel.: +761-2 03 30 65

Michael E. **Pohst**
Universität Düsseldorf
Mathematisches Institut
Universitätsstr. 1
W-4000 Düsseldorf 1
Germany
pohst@ze8.rz.uni-duesseldorf.de
tel.: +211-311-21 88

Georg **Rex**
Universität Leipzig
Sektion Mathematik
Augustusplatz 10
O-7010 Leipzig
Germany
rex@mathematik.uni-leipzig.dbp.de
tel.: +719-23 17/18

Siegfried M. **Rump**
TU Hamburg-Harburg
Technische Informatik III
Eisendorfer Str. 38
W-2100 Hamburg 90
Germany

Mark J. **Schaefer**
Universität Tübingen
Wilhelm-Schickhard-Institut
Sand 13
W-7400 Tübingen
Germany

Hans J. **Stetter**
TU Wien
Institut für Angewandte und
Numerische Mathematik
Wiedner Hauptstr. 8/10
A-1040 Wien
Austria
stetter@uranus.tuwien.ac.at
tel.: +43-1-58801-54 08

Bernd **Tibken**
Universität Ulm
Abteilung Meß-
Regel- u. Mikrotechnik
Postfach 40 66 W-7900 Ulm
Germany
+731-502-41 92

Peter **Walerius**
Universität Bremen
Institut für Automatisierungstechnik
Kufsteiner Str.
W-2800 Bremen 33
Germany
tel.: 0421-218-3490

Michael A. **Wolfe**
University of St. Andrews
North Haugh
St. Andrews KY16 9SS
Great Britain
maw@st-andrews.ac.uk
tel.: +44-334-76161-81 75

P. Klint, T. Reps, G. Snelting (editors):
Programming Environments; Dagstuhl-Seminar-Report; 34; 9.3.-13.3.92 (9211)

H.-D. Ehrich, J.A. Goguen, A. Sernadas (editors):
Foundations of Information Systems Specification and Design; Dagstuhl-Seminar-Report; 35; 16.3.-19.3.9 (9212)

W. Damm, Ch. Hankin, J. Hughes (editors):
Functional Languages:
Compiler Technology and Parallelism; Dagstuhl-Seminar-Report; 36; 23.3.-27.3.92 (9213)

Th. Beth, W. Diffie, G.J. Simmons (editors):
System Security; Dagstuhl-Seminar-Report; 37; 30.3.-3.4.92 (9214)

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 5.4.-9.4.92 (9215)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 39; 22.06.-26.06.92 (9226)

E. Börger, Y. Gurevich, H. Kleine-Büning, M.M. Richter (editors):
Computer Science Logic, Dagstuhl-Seminar-Report; 40; 13.07.-17.07.92 (9229)

J. von zur Gathen, M. Karpinski, D. Kozen (editors):
Algebraic Complexity and Parallelism, Dagstuhl-Seminar-Report; 41; 20.07.-24.07.92 (9230)

F. Baader, J. Siekmann, W. Snyder (editors):
6th International Workshop on Unification, Dagstuhl-Seminar-Report; 42; 29.07.-31.07.92 (9231)

J.W. Davenport, F. Krückeberg, R.E. Moore, S. Rump (editors):
Symbolic, algebraic and validated numerical Computation, Dagstuhl-Seminar-Report; 43; 03.08.-07.08.92 (9232)

R. Cohen, R. Kass, C. Paris, W. Wahlster (editors):
Third International Workshop on User Modeling (UM'92), Dagstuhl-Seminar-Report; 44; 10.-13.8.92 (9233)

R. Reischuk, D. Uhlig (editors):
Complexity and Realization of Boolean Functions, Dagstuhl-Seminar-Report; 45; 24.08.-28.08.92 (9235)

Th. Lengauer, D. Schomburg, M.S. Waterman (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 46; 07.09.-11.09.92 (9237)

V.R. Basili, H.D. Rombach, R.W. Selby (editors):
Experimental Software Engineering Issues, Dagstuhl-Seminar-Report; 47; 14.-18.09.92 (9238)

Y. Dittrich, H. Hastedt, P. Schefe (editors):
Computer Science and Philosophy, Dagstuhl-Seminar-Report; 48; 21.09.-25.09.92 (9239)

R.P. Daley, U. Furbach, K.P. Jantke (editors):
Analogical and Inductive Inference 1992 , Dagstuhl-Seminar-Report; 49; 05.10.-09.10.92 (9241)

E. Novak, St. Smale, J.F. Traub (editors):
Algorithms and Complexity of Continuous Problems, Dagstuhl-Seminar-Report; 50; 12.10.-16.10.92 (9242)

J. Encarnação, J. Foley (editors):
Multimedia - System Architectures and Applications, Dagstuhl-Seminar-Report; 51; 02.11.-06.11.92 (9245)

F.J. Rammig, J. Staunstrup, G. Zimmermann (editors):
Self-Timed Design, Dagstuhl-Seminar-Report; 52; 30.11.-04.12.92 (9249)

B. Courcelle, H. Ehrig, G. Rozenberg, H.J. Schneider (editors):
Graph-Transformations in Computer Science, Dagstuhl-Seminar-Report; 53; 04.01.-08.01.93 (9301)

A. Arnold, L. Priese, R. Vollmar (editors):
Automata Theory: Distributed Models, Dagstuhl-Seminar-Report; 54; 11.01.-15.01.93 (9302)

W.S. Cellary, K. Vidyasankar, G. Vossen (editors):
Versioning in Data Base Management Systems, Dagstuhl-Seminar-Report; 55; 01.02.-05.02.93 (9305)

B. Becker, R. Bryant, Ch. Meinel (editors):
Computer Aided Design and Test, Dagstuhl-Seminar-Report; 56; 15.02.-19.02.93 (9307)

M. Pinkal, R. Scha, L. Schubert (editors):
Semantic Formalisms in Natural Language Processing, Dagstuhl-Seminar-Report; 57; 23.02.-26.02.93 (9308)

H. Bibel, K. Furukawa, M. Stickel (editors):
Deduction, Dagstuhl-Seminar-Report; 58; 08.03.-12.03.93 (9310)

H. Alt, B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 59; 22.03.-26.03.93 (9312)

J. Pustejovsky, H. Kamp (editors):
Universals in the Lexicon: At the Intersection of Lexical Semantic Theories, Dagstuhl-Seminar-Report; 60; 29.03.-02.04.93 (9313)

W. Straßer, F. Wahl (editors):
Graphics & Robotics, Dagstuhl-Seminar-Report; 61; 19.04.-22.04.93 (9316)

C. Beeri, A. Heuer, G. Saake, S.D. Urban (editors):
Formal Aspects of Object Base Dynamics, Dagstuhl-Seminar-Report; 62; 26.04.-30.04.93 (9317)

R. Book, E.P.D. Pednault, D. Wotschke (editors):
Descriptional Complexity: A Multidisciplinary Perspective, Dagstuhl-Seminar-Report; 63; 03.05.-07.05.93 (9318)

M. Wirsing, H.-D. Ehrich (editors):
Specification and Semantics, Dagstuhl-Seminar-Report; 64; 24.05.-28.05.93 (9321)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Algebra, Dagstuhl-Seminar-Report; 65; 07.06.-11.06.93 (9323)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 66; 28.06.-02.07.93 (9326)

Ph. Flajolet, R. Kemp, H. Prodinger (editors):
"Average-Case"-Analyse von Algorithmen, Dagstuhl-Seminar-Report; 67; 12.07.-16.07.93 (9328)

J.W. Gray, A.M. Pitts, K. Sieber (editors):
Interactions between Category Theory and Computer Science, Dagstuhl-Seminar-Report; 68; 19.07.-23.07.93 (9329)

V. Marek, A. Nerode, P.H. Schmitt (editors):
Non-Classical Logics in Computer Science, Dagstuhl-Seminar-Report; 69; 20.09.-24.09.93 (9338)

A. Odlyzko, C.P. Schnorr, A. Shamir (editors):
Cryptography, Dagstuhl-Seminar-Report; 70; 27.09.-01.10.93 (9339)