BIBLIOGRAPHY

ON NUMERICAL SOFTWARE

by

Bo Einarsson

Permanent affiliation: FÖRSVARETS FORSKNINGSANSTALT (FOA)
National Defence Research Institute
Department 2/Proving Grounds
Box 98, S-147 00 Tumba, Sweden

Memorandum No. UCB/ERL M77/19

25 March 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720
BIBLIOGRAPHY

ON NUMERICAL SOFTWARE

by

Bo Einarsson

Permanent affiliation: FÖRSVARETS FORSKNINGSANSTALT (FOA)
National Defence Research Institute
Department 2/Proving Grounds
Box 98, S-147 00 Tumba, Sweden

Memorandum No. UCB/ERL M77/19

25 March 1977
BIBLIOGRAPHY

ON NUMERICAL SOFTWARE

by

Bo Einarsson

Permanent affiliation: FÖRSVARETS FORSKNINGSANSTALT (FOA)
National Defence Research Institute
Department 2/Proving Grounds
Box 98, S-147 00 Tumba, Sweden

Memorandum No. UCB/ERL M77/19

25 March 1977
Abstract:

This bibliography has been written at the request of the IFIP Working Group on Numerical Software (IFIP WG 2.5), and is intended to serve both members of the working group and others intent on improving numerical software.

It has been divided into twenty-one different areas. Within each area the references are given in alphabetical order by the first author. Some references occur in two or at most three areas. This is especially true for the individual articles in the books of Section 2. For some entries a summary is included; either the original abstract, or a shortened form of the original, or a summary written by the present author.

The aim of the bibliography is to be useful in the production and evaluation of good software for numerical mathematics. However, it does not include references to algorithms in the numerical analysis literature, nor does it include references to individual software products (routines). Section 7 on bibliographies includes many entries not strictly within the scope of the present work. I have tried to get Sections 4 (Numerical Program Libraries), 11 (Transportability), and 16 (Evaluations) as complete as possible, but the other sections are not so well covered, especially as regards matters outside of numerical software.
CONTENTS

Abstract i

Contents ii

History of this Bibliography iv

Acknowledgements v

1. Programming Methods and Principles 1

2. Books related to Mathematical Software 5

3. Standards 14

4. Numerical Program Libraries 20
   a) General Articles 20
   b) Some Commercial Numerical Libraries 27
   c) Some Non-commercial Numerical Libraries 29

5. Classification 34

6. Documentation 35

7. Bibliographies 37

8. Desirable Language Features for Numerical Programming 43

9. Communication Languages 44

10. FORTRAN Dialects 45

11. Transportability 48

12. Aids for Software Generation and Evaluation 53

13. Machine Constants 65

14. Hardware Features 66

15. Criteria for Evaluating Numerical Software 70
   a) General Articles 70
   b) Guidelines for Transportable Evaluation Tools 73
   c) Criteria for Evaluating Quadrature, Ordinary Differential Equation and Integral Equation Software 73
16. Evaluations
   a) General Articles and Miscellaneous Evaluations 74
   b) Linear Algebra 76
   c) Quadrature 79
   d) Ordinary Differential Equations 83
   e) Functions 88
   f) Optimization 92
   g) Integral Equations 93

17. Formal Proofs of Correctness 94

18. Structured Programming 98

19. Test Problems 103

20. Miscellaneous 105

21. Journals Related to Numerical Software 117
History of this Bibliography

A first version of this bibliography, with the title "An Annotated Bibliography on Mathematical Software and Related Topics," was written in 1974 by Eva Edberg and Jan Johansson and was later published in the SIGNUM Newsletter, Vol. 11, No. 2 (August 1976), pp. 9-16, and No. 3 (October 1976), p. 6. That version was produced at Uppsala University, with financial support from the Swedish Institute of Applied Mathematics and the Stockholm Data Center.

At the request of the IFIP Working Group on Numerical Software (IFIP WG 2.5) I started early in 1975 to collect additional material, and with the support of the Swedish National Defense Research Institute, a preliminary version was distributed in May 1976 and a slightly revised version in October 1976.

This version, which is the first version in the form of a formal report, was produced at the Computer Science Division and Electronics Research Laboratory, University of California, Berkeley, California.

It has been discussed by IFIP WG 2.5 to put the bibliography in machine readable form at a later time. Consequently I have chosen not to have the material retyped in a uniform manner and apologize to the reader for any resulting inconvenience.
Acknowledgements

I would like to thank Eva Edberg and Jan Johansson for their excellent original work. I also thank Ed L. Battiste, Fred N. Fritsch, Thos E. Hull, Tapio Niemelä, John R. Rice, and Larry F. Shampine for significant contributions.

My work on the present version has been supported by grants from the Sweden-America Foundation and the National Science Foundation, while the production of the present report has been financed by the Office of Naval Research Contract N00014-76-C-0013. I wish to express my sincere thanks to these three organizations for their support.

I am also very grateful to Professors Elwyn R. Berlekamp, William Kahan, and Beresford N. Parlett for their kind hospitality during my stay at the Computer Science Division of the University of California at Berkeley during the winter quarter of 1977.

Finally, I would like to thank Maj-Britt Kähre, Agneta Österlund, and Suzanne Briggs for excellent typing of the many entries.
1. PROGRAMMING METHODS AND PRINCIPLES


Available free of charge for institutions from TRW Technical Information Center, Mail Station S/1930, One Space Park, Redondo Beach, California 90278, Phone (213) 535-4321.

I would like to recommend a careful study of this long report.


Göran Fick, Program structures in Fortran when adjustable dimensions are used, including a method to simulate dynamic memory allocation, FOA 2 report C 2556-E5, September 1972 (In English).


HETZEL, William C (Ed) (1973): PROGRAM TEST METHODS

HILL, I.D., SCOWEN, R.S., WICHMANN, B.A., Writing Algorithms in Algol 60, Software-Practice and Experience 5 (1975), pp 229-244.
This report discusses the difficulties of writing procedures and programs in Algol 60 for general use. It also shows how the problems can be alleviated and overcome.


This book is a study of a large number of "real" programs, each of which provides one or more lessons in style.


Describes an investigation where "ordinary programs" were collected at some computer centers and closely analyzed and judged. Discusses the use of "program profiles", statistics on the rate of use of different types of statements in a program.
The authors have realised that as most users are unable to get inside commercially available mathematical programming packages, a large variety of generally inefficient, "home-grown" programs have been developed and also a lot of untested new algorithms have been published. The final chapter deals with problems that may be encountered in either moving the routine onto a new computer or altering the size of problems that may be solved.

McCRAKKE, Daniel D and WEINBERG, Gerald M: How to Write a Readable FORTRAN Program, DATAMATION Vol 18(10), pp. 73-73.
Since you've got to write the program anyway, why not do it in a way that produces documentation as a by-product?


2. BOOKS RELATED TO MATHEMATICAL SOFTWARE


COMPUTER SCIENCE AND STATISTICS: Nth Annual Symposium on the Interface. Annual Conference Proceedings, N = 9 was at Harvard University, 1-2 April 1976 and N = 8 was at University of California, Los Angeles, 13-14 February 1975.


Some Side Effects of Striving for Portability - C. Reinsch

Mathematical Software Transportability Systems - Have the Variations a Theme? - J. Boyle

Aids to Portability within the NAG Project - J. Du Croz, S. J. Hague, and J. L. Siemieniuch

Features for FORTRAN Portability - F. Krogh

The IMSL Fortran Converter: An Approach to Solving Portability Problems - T. Aird

Multiple Program Realizations Using the TAMPR System - K. Dritz
On the Enhancement of Portability within the NAG Project: A Statistical Survey - B. Ford and J. Bentley

The Importance of Standardized Interfaces for Portable Statistical Software - N. Victor and M. Sund

A Study of Portability in Technical and Scientific Computing - I. Dahlstrand

Fortran Poisoning and Antidotes - B. Smith

Semantics of Floating Point Arithmetic and Elementary Functions - T. Hull

Two Numerical Analyst's Views on the Draft Proposed ANS Fortran - C. Lawson and J. Reid

The Production and Testing of Special Function Software in the NAG Library - J. Schonfelder

Portable Special Function Routines - W. Fullerton

Algol 68 as a Language for Numerical Software - M. Delves

Writing the Elementary Function Procedures for the ALGOL 68C Compiler - P. Kemp

Criteria for Transportable Algol Libraries - P. Hemker

Fortran Portability via Models and Tools - S. Brown and A. Hall

PORT - A Portable Mathematical Subroutine Library - P. Fox

Machine Parameters for Numerical Analysis - W. Cody

Machine Requirements for Reliable, Portable Software - T. Dekker

Intermediate Languages: Current Status - W. Waite


Contents:

EVANS, D. J.: Iterative sparse matrix algorithms, pp. 49 - 83.
EINARSSON, Bo: Testing and evaluation of some subroutines for numerical quadrature, pp. 149 - 157.
GENZ, A. C.: Some extrapolation methods for the numerical calculation of multidimensional integrals, pp. 159 - 172.
WALSH, Joan: Initial and boundary value routines for ordinary differential equations, pp. 177 - 189.
THOMAS, C. L.I.: POTENT - a package for the numerical solution of potential problems in general two-dimensional regions, pp. 315 - 336.

FLETCHER, R. and HEDEMAN, M. D.: Setting up a numerical advisory service, pp. 413 - 421.


HETZEL, William C. (Ed.) [1973]: PROGRAM TEST METHODS

Contents:
PART I INTRODUCTION

PART II TESTING CONCEPTS

PART III DESIGNING PROGRAMS FOR TESTING
VYSSOTSKY, V. A.: Common Sense in Designing Testable Software, pp. 41 - 47.
SNOWDON, R. A.: System for the Preparation and Validation of Structured Programs, pp. 57 - 72.

PART IV DESIGNING LANGUAGES FOR TESTING

PART V TESTING MATHEMATICAL SOFTWARE
PART VI TESTING LARGE SOFTWARE SYSTEMS


PART VII MODELS OF PROGRAM BEHAVIOR


PART VIII STANDARDS AND MEASUREMENTS OF PROGRAM QUALITY


PART IX BIBLIOGRAPHY


Contents:
HILL, David R.: Second derivative multistep formulas based on g-splines, pp. 25-38.
SUBJECT INDEX, pp. 287-291.


Contents:

PART ONE: PROLOGUE
RICE, John R.: Historical Notes, pp. 3 - 12.
RICE, John R.: Discussion of Papers, pp. 43 - 47.

PART TWO: PROCEEDINGS OF THE SYMPOSIUM
THATCHER, H. C. jr, Making Special Arithmetics Available, pp. 113 - 119.
de BOOR, Carl, On Writing an Automatic Integration Algorithm, pp. 201 - 209.


PART THREE: SELECTED MATHEMATICAL SOFTWARE

G. H. Golub, "The block Lanczos method for computing Eigenvalues"
G. W. Stewart, "Research, development, and LINPACK"
M. J. D. Powell, "A technique that gains speed and accuracy in the minimax solution of over-determined linear equations"
G. E. Collins, "Infallible calculation of polynomial zeros to specified precision"
R. E. Barnhill, "Representation and approximation of surfaces"
C. W. Gear, "Simulation: conflicts between real time and software"
D. C. Hoaglin, "Mathematical software and exploratory data analysis"
C. L. Lawson, "Software for C1 surface interpolation"
W. R. Cowell, L. D. Fosdick, "Mathematical software production"
W. S. Brown, "Portability"
I. Babuska, "Computational aspects of the finite element method"
L. F. Shampine, "The art of writing a Runge-Kutta code"
A. Brandt, "Multi-level adaptive techniques for partial differential equations: ideas and software"


Contains 24 contributions of which some are of interest for numerical software, especially session 2 on portability and session 5 on numerical applications.


3. STANDARDS

All the ANSI and ISO standards are available through your national standards organization.

American National Standard, Basic FORTRAN, ANSI X3.10-1966, American National Standards Institute

American National Standard, FORTRAN, ANSI X3.9-1966, American National Standards Institute


ANSI X3J3: Draft proposed revised Fortran standard. SIGPLAN Notices, March 1976. (Special issue)
This is the formal document submitted to ANSI X3 for further processing. Several revisions have been made since publication of the draft.


BSR X3.53 programming language PL/I (draft standard).

BSR X3.60, Programming Language Minimal Basic (draft standard).


Center publication S-22, 1971. Adr: Norsk Regnesentral, Forskningsveien 1b,
Oslo, Norway.

DAY, A. C., CLARKE, P. A., HILL, D. and REID, J. K.: The proposed new
standard for FORTRAN: a critical examination. The Computer Journal,
This is a critical review of the draft dated 26 September 1975, which
has been obsolete for a long time.

ECMA (European Computer Manufacturers' Association),
ECMA STANDARD for a Subset of ALGOL 60,

FE LDMA N, S. I.: A Fortranner's lament: comments on the draft proposed
ANS Fortran standard. SIGPLAN Notices, Vol. 11, No. 12 (December 1976),
pp. 25-34.
Discusses the draft in SIGPLAN Notices, March 1976.

FORD, Brian, REID, J. K. and SMITH, B. T.: Three proposed amendments to
the draft proposed ANS Fortran standard. Submitted to SIGNUM Newsletter,
September 1976. See also FORWORD, Fortran Development Newsletter, Vol. 2,
No. 4 (October 1976), pp. 29-31.

P. A. Fox, A. D. Hall and N. L. Schryer,
"The PORT Mathematical Subroutine
Library", Computing Science Technical
Report No. 47, Bell Laboratories, Murray
Hill, New Jersey, September 1976.

GRISWOLD, R. et al.: The Snobol 4 programming language, Second Ed.,

GROOMS, David W.: Computer software standards (a bibliography with
NTIS/PS-76/0411/9WC.

The purpose of this report is to propose a set of standard subroutines (modules) for performing many of the elementary operations of numerical linear algebra. The goal is to make it more feasible to produce efficient portable FORTRAN programs in the area of linear algebra.


The report contains a proposed list of calling sequence parameters for ordinary differential equation solvers.

IBM System 360 operating system, PL/I(F) language reference manual, IBM Order No. GC28-8201.


This version is the Naur(1963) Revised Algol 60 Report with the modifications of De Morgan et al. (1976) included.


IFIP WG 2.5: Amendments to Draft Proposed ANS FORTRAN, Specifying the Double Precision Complex Data Type. Manuscript June 29, 1976.

IFIP WG 2.5: Parameters for Transportable Numerical Software. Manuscript July 1, 1976. Submitted to IFIP for publication.

IFIP WG 2.5: MAP Statement for Fortran to Assist in the Portability of Numerical Software. Manuscript September 10, 1976 (16pp.)
ISO: ISO recommendation R1538, programming language Algol 60, International Organization for Standardization, 1972. (This standard is not recognized by IFIP, at least the first printing was erroneous.)


LARMOUTH, J (1973): Serious FORTRAN, Software Vol 3, pp. 87-107. Discusses writing of "serious" programs in a scientific environment. It is directed to those who intend to write FORTRAN programs which have more than a transient life. This first part deals with the implications of conforming to the ANSI standard when writing FORTRAN.

LARMOUTH, J (1973): Serious FORTRAN - Part 2, Software Vol 3, pp. 197-225. The second part of this paper addressed to those who write FORTRAN programs of more than transitory life deals with facilities outside the Standard, optimization, recursion, the design of user interfaces, debugging and program proving.


LAWSON, C L; HANSON, R J; KINCAID, D and KROGH, F T: Basic Linear Algebra Subprograms for Fortran Usage. Submitted.

MATHEMATICAL CENTER: An extensive Test Set for ALGOL 68 Compilers has been assembled and comprises 160 programs covering the entire language. Please address requests to: Dick Grune, Mathematical Center, Tweede Boerhaavestraat 49, Amsterdam, The Netherlands.


An easy to read description of the proposed new Fortran.


The goal of this publication is to enable programmers already familiar with FORTRAN to write standard (i.e. interchangeable) programs.


NEEDHAM, Tom: Graphic extensions to the Basic language, Digest of Papers, Comp Con Spring 77, IEEE Catalog No. 77CH1165-OC, pp. 314-317.

PALME, Jacob: Experience from the Standardization of the SIMULA Programming Language. SOFTWARE - Practice & Experience, Vol 6, No. 3, July-September 1976, pp 405-409.


4. **NUMERICAL PROGRAM LIBRARIES**

a) **GENERAL ARTICLES**


Contains information on various source types, technical, financial and organizational problems.


Probably the best overview of principles and ideas behind the National Activity to Test Software.

E. CATE, A. ERISMAN, P. LU and R. SOUTHALL - Boeing
A User Oriented Multi-Level Math Library

In: Rice, John R.
Purdue University, May 29-31, 1974.


COWELL, Wayne R and FOSDICK, Lloyd D: A Program for Development of High Quality Mathematical Software.


CHARLES DUNHAM - University of Western Ontario
Development and Publication of Numerical Algorithms

In: Rice, John R.
Purdue University, May 29-31, 1974.
Contains a description of the most usual program libraries with comments on availability, economical conditions and reliability. An addendum from March 1974 is also available from ITM.

EKBLOM, liakan (1973): Matematisk programvara; problem och möjlig-neter (in Swedish) (Mathematical software; problems and possibilities), Ref: DATA No. 6
Concentrates on the role of the computer centre in distributing mathematical software. Contains a review of the Stanford study of FORTRAN programs by Knuth.

This book is intended to serve as a unified program library catalog for the unit members within the Joint User Group of the ACM. The book can be important in efforts to develop industry-wide standards for program documentation, because there are already standards in force within the individual units.
The directory contains two main sections: a program description section and a subject index. About 1000 programs from 11 units are listed, the most recent additions having been made in late 1970.
The 1974 edition is available from Macmillan Information, 866 Third Avenue, New York.

FORD, Brian: The Nottingham Algorithms Group (NAG) Project.

FORD, B and HAGUE, S J: The organisation of numerical algorithms libraries, pp. 357-372.


More detailed results are available as "User Ratings of Proprietary Software" for $12 from Datapro Research Corporation, 1805 Underwood Boulevard, Delran, New Jersey 08075.

M. A. HENNELL - University of Liverpool
The Design and Implementation of an Algol 68 Numerical Algorithms Library for NAG
In: Rice, John R.
Purdue University, May 29-31, 1974.

INTERNATIONAL COMPUTER PROGRAMS Inc: Software Directory.
1119 Keystone Way, Carmel, Indiana 46032.
More than 3,000 proprietary software products produced by more than 800 vendors are described in the latest semiannual, two-volume ICP Software Directory. The first volume (338 pp.) is on systems software, and the second (614 pp.) covers applications software. There are over 2,000 product updates in this version, plus 600 new product listings. Price: $100/yr. by subscription, or $60 for two volumes on a one-time basis.

JOHNSON, O G: IMSL's Ideas on Subroutine Library Problems.

In RICE, John R (Ed): Mathematical Software. Based on the proceedings of the Mathematical Software Symposium held at Purdue University, Lafayette, Indiana, April 1970.


SCHICHER, O.; SCHMID, F. and ZELLE, K: Algorithmen Programme
Programmpakete, Institut für Stadtforschung, Währinger-
A Catalog of more than 3000 subroutines from journals, research
centers, and computer manufacturers.

SMITH, B. T., BOYLE, J. M. and CODY, W. J.: The NATS approach
of quality software, pp. 393 - 405.
Proceedings of the Loughborough University Conference of
the Institute of Mathematics and Its Applications held

SMITH, B.T. et al: Matrix Eigensystem Routines, EISPACK

IFIP WG 2.5 Position Paper, January 13-16, 1975, Oxford
University, 5 pp.

TAYLOR, D. B., FORD, B. and HAGUE, S. J.: Management practices
in the development and distribution of mathematical software
with emphasis on computational aids in multi-machine environment,
pp. 373 - 382.
Proceedings of the Loughborough University Conference of
the Institute of Mathematics and Its Applications held

TRAUB, J. F.: High Quality Portable Numerical Mathematics
Software, pp. 131 - 139.
In RICE, John R (Ed): Mathematical Software. Based on the
proceedings of the Mathematical Software Symposium held
at Purdue University, Lafayette, Indiana, April 1970.

-25-

   The present work cites almost all the significant papers on advances in the mathematical theory; reports on applications; covers such topics as classical analysis, functional analysis, approximation theory, fluids and diffusion; lists finite element packages.

   A catalog of more than 400 published Algol procedures for matrix problems, with comments on accuracy and usefulness.
4. NUMERICAL PROGRAM LIBRARIES

b) SOME COMMERCIAL NUMERICAL LIBRARIES (available at cost)

1. IMSL is a mathematical and statistical subroutine library in Fortran, available for IBM 360/370, Xerox, UNIVAC 1100, Honeywell-Bull 600/6000, DEC System-10, CDC 6600/7600/CYBER 70/170, Burroughs 6700/7700

   It is used at five hundred computer centers. New release each year as announced. Further information from International Mathematical & Statistical Libraries, Inc., 7500 Bellaire Boulevard, Sixth Floor, GNB Building, Houston, Texas 77036. Tel. (713) 772-1927.

2. NAG is a British subroutine library, available in both Fortran and Algol 60. An Algol 68 version is being developed. Versions exist for IBM 360/370, CDC 6600/7600, CYBER 70/170, Burroughs 5700/6700/7700, UNIVAC 1100, PRIME, DEC System 10 and various ICL computers. It is used extensively in British Universities and is now being made available internationally. Further information from the NAG Central Office, 7 Banbury Road, Oxford OX2 6NN, England. Tel (0865) 511245.

   New release each year as announced. Work is in progress on a version for GEC computers.

3. The PORT Mathematical Subroutine Library is a rather recent software product from Bell Laboratories, Murray Hill, New Jersey 07974. It has been written with particular emphasis on portability, which is achieved by careful language specification and specifying computer classes by means of predefined machine constants. The library has been described by Phyllis Fox, see references in section 2, COWELL, Wayne R (Editor) with entry FOX and section 4a. The library has been tested extensively on IBM and Honeywell.

Information on availability is given in the SIGNUM Newsletter, Vol. 11, No. 2 (August 1976), p. 8.


A version in PL/1 is called PL-Math, but does not contain the sparse routines.


Further information from Control Data. This library is now considered almost obsolete.
4. **NUMERICAL PROGRAM LIBRARIES**

c) **SOME NON-COMMERCIAL NUMERICAL LIBRARIES**

1. **SSP.** This old and well-known Fortran library from IBM is available on most computers. It is considered to be out-of-date but still used. There also exists a PL/1 version (The only known non-commercial PL/1 library?) Ref. System/360 Scientific Subroutine Package, Version III, Programmer's Manual GH 20-0205-3 with Technical Newsletter N 20-1944.

2. **MATH-PACK.** Univac has a Fortran library similar to SSP. Most other computer manufacturers also have similar libraries.

3. **CNRS.** A collection "Procedures Algol en Analyse Numérique" has been issued by Centre National de la Recherche Scientifique, Service des Publications, Ventes 15, Quai Anatole-France, Paris 7.

4. **HARWELL.** The United Kingdom Atomic Energy Authority Research Group at Harwell has a large Fortran library, that is being used also at many other centers. Further information is available from Mr S. Marlow, Building 8.9, AERE, Harwell, Oxfordshire OX11 ORA. See also the report AERE-R 7477, A catalogue of Subroutines.

   The original version is for IBM 360/370. The library is specially strong in optimization and sparse matrices.

5. **NPL.** The National Physical Laboratory has a large library of routines which aims to cover most numerical processes widely required in the scientific disciplines. The Algorithms are coded in Algol 60 and Standard Fortran although some may be available in one language only. The highest degree of machine and compiler independence has been aimed at. There is one document for each algorithm for each available language. Routines may be supplied individually.
The library is based upon expertise accumulated by numerical analysts at NPL through years of research and problem solving. These analysts may be consulted upon most numerical problems and usually provide advice and assistance in the early stages of use of an algorithm. Material is continually added to the library as a result of research effort at NPL.

A guide to the library will be supplied free of charge upon application to: Dr. J.R.A. Cooper, Division of Numerical Analysis and Computing, National Physical Laboratory, Teddington, Middlesex TW11 OLW.

6. NATS. National Activity to Test Software is a United States project managed by the Applied Mathematics Division of the Argonne National Laboratory in cooperation with universities in North America and Europe. So far the eigensystem package EISPACK and the special functions package FUNPACK are available for all main U.S. computers.

Further information from Burton S. Garbow, Applied Mathematics Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.

EISPACK is available for IBM 360/370, CDC 6600/7600, UNIVAC 1100, DEC System-10, Honeywell 600/6000 and Burroughs 6700.

FUNPACK is available for IBM 360/370, CDC 6600/7600 and UNIVAC 1100.


7. JPL. The Jet Propulsion Laboratory in Pasadena, California has a private library, described in Report 1846-23; Rev. A, February 1, 1975. This report is an excellent example of how a private library ought to be documented and presented to its local users.

8. NUMAL.

On request of the Academic Computing Centre of Amsterdam (SARA) the Mathematical Centre adapted and extended its library of numerical procedures for use with the CD CYBER 70 System, the resulting library called "NUMAL" (" NUM" ERICAL PROCEDURES IN "AL"GOL 60).

The aim of NUMAL is to provide a high level numerical library for Algol 60 Programmers. The library contains a set of validated numerical procedures together with supporting documentation. Except for a small number of double precision arithmetic routines all the source texts are written in Algol 60 and they are to a high degree independent of the computer/compiler used.
The library is now in use by several scientific computer centers in the Netherlands. Full reference documentation (ca. 800 pages) is distributed to subscribers. Once a year additions and improvements are released.

Ref. Mathematical Centre report NW8/76: NUMAL, a library of numerical procedures in ALGOL 60. Index and KWIC-index (3rd edition).

Further information from Stichting Mathematisch Centrum, 2e Boerhaavestraat 49, Amsterdam-1005, The Netherlands.

9. CERN. The "Centre Européenne pour la Recherche Nucléaire" in Geneva has one of the largest subroutine libraries in Europe. It is intended for the CDC 6600/7600 series, but mappings exist to other systems, including UNIVAC 1100. Some routines are very machine dependent. The library is classified according to SHARE and contains three categories:

a) numerical routines
b) data manipulation routines
c) nuclear physics applications

Further information from Program Librarian, CERN, Div. 25, CH-1211 Geneva 23, Switzerland.
10. CPC (Computer Physics Communications Program Library). Computer programs in Physics are being collected at the Queen’s University of Belfast and are announced in the Computer Physics Communications, see Volume 1 (1970), pp. 473-476 and Volume 10 (1975), p. 203. The current indexes of the contents of the program library as well as the programs are available from the C.P.C. Program Library, School of Physics and Applied Mathematics, Queen’s University, Belfast BT7 1NN, Northern Ireland.

11. ANL. The Argonne Code Center is responsible for operating a computer software and data exchange and information center under U.S. Energy Research and Development Administration. Registered Installations receive a copy of the Program Abstracts, ANL 7411. A nominal fee is required for non-ERDA installations. Information is available from:

Argonne Code Center
Argonne National Laboratory
9700 South Cass Avenue
ARGONNE, Illinois 60439.

5. **CLASSIFICATION**

A keyword index, as well as a revised classification scheme, is being developed for the Collected Algorithms from ACM.


CODY, William J: Letter to Professor Lloyd D Fosdick Dec 3, 1974, regarding the proposed Bolstad Classification scheme (3 pp.).

FORD, Brian: Classification of Numerical Algorithm Libraries. Manuscript 9 January 1975, Oxford University (2 pp.)


IBM Systems Reference Library, Catalog of Programs for IBM System/360, GC 20-1619.
Contains three different classification systems, of which the one for Type III and IV Programs is being used widely.


6. DOCUMENTATION


Documentation conventions for FORTRAN are urgently needed to provide for better understanding and communication of packages and routines. The required properties of these conventions are defined, and a descriptive list of documentation conventions entitled "Stirling FORDOC 01", in use at the University of Stirling is presented in the appendix.


FRITSCH, F. N. and HAUSMAN, R. F.: On the documentation of computer programs, UCID-30043, March 1972, Lawrence Livermore Laboratory, 16 pp. Contains suggested format for computer program documentation (7 pp.), programming suggestions (5 pp.), and testing principles (1 p.).


This paper concerns what kind of documentation that a program should have for the purpose of certification and validation by an independent tester, for example a reviewer for an algorithm journal.


K. A. REDISH - McMaster University
Tree Structures for a Program Library Index
In: Rice, John R.
Purdue University, May 29-31, 1974.

ROBERTS, K. V.: The Publication of Scientific FORTRAN Programs,
This article outlines some general principles which appear to be necessary if an international literature of published scientific programs is to be successfully established. Programming conventions are suggested for FORTRAN, together with several automatic documentation tools which have already been tried out and found useful.


7. **BIBLIOGRAPHIES**

ACR: Quarterly bibliography of computers and data processing, a subject/author index to computer literature. Published by Applied Computer Research, P. O. Box 9280, Phoenix, Arizona 85068.

ANNUAL BIBLIOGRAPHY OF COMPUTER ORIENTED BOOKS. Computing Newsletter, Box 7345, Colorado Springs, Colorado 80933.

This bibliography contains more than 1000 entries.
Price: $4 ($5 if invoice required).

BARNARD, David and THOMPSON, David: An annotated bibliography on computer program engineering, University of Toronto, Toronto, Canada, 100 pp.
Available as R76-264 from IEEE Repository.

Fritz Bierbaum,

Intervall-Mathematik, Eine Literaturübersicht, Interner Bericht Nr. 74/2 und 75/3, UNIVERSITAT KARLSRUHE, Institut für Praktische Mathematik, D-75 Karlsruhe 1, Englerstrasse 2, Postfach 6380, Germany.

B. W. Boehm et al,

Characteristics of Software Quality, TRW SOFTWARE SERIES, TRW-SS-73-09, December 1973. (Section 6 is a nine page annotated bibliography).
Available free of charge for institutions from TRW Technical Information Center, Mail Station S/1930, One Space Park, Redondo Beach, California 90278, Phone (213) 535-4321.

Survey with extensive bibliography.

Supplement 1 to ORNL-4778 revision, see Householder below.

Some 900 basic information sources have been given a critical evaluation. The sources are listed in ten categories. The author has written a critical review for each entry.
Ref: Computing Reviews 16 No. 28333 and 28475.
Elise de Doncker and Robert Piessens,
A bibliography on automatic integration.


FORSYTHE, George E: Recent References on Solving Elliptic Partial Differential Equations by Finite Differences or Finite Elements. SIGNUM Newsletter Vol 6, No. 1, January 1971, pp. 32-56.


Myron Ginsberg, Bibliography 36.

GIRARD, E and RAULT, J-C: L'étude des méthodes de test et d'évaluation de la fiabilité du logiciel (Study of test methods and evaluation of software reliability). Report SCAS 74.589, May 1974, in French, 89 pp. and a bibliography with 1556 items. THOMSON-CSF, DIB, 33 Rue de Vouillé, F-75724 Paris.
GROOMS, David W:


GROOMS, David W.: Symbolic programming (a bibliography with abstracts). National Technical Information Service (May 1976), NTIS/PS-76/0121/4WC.

GROOMS, David W.: Computer software maintenance (a bibliography with abstracts). National Technical Information Service (May 1976), NTIS/PS-76/0322/8WC.

GROOMS, David W.: Computer software transferability and portability (a bibliography with abstracts). National Technical Information Service (May 1976), 46 pp. NTIS/PS-76/0388/9WC.

GROOMS, David W.: Computer software standards (a bibliography with abstracts). National Technical Information Service (June 1976), 92 pp. NTIS/PS-76/0411/9WC.


GROOMS, David W.: Integration of partial differential equations (a bibliography with abstracts). National Technical Information Service (October 1976), 152 pp. NTIS/PS-76/0791/4WC.

William C. Hetzel (Editor),


Part IX, Bibliography, is an extensive annotated bibliography, pp. 313-348.

This listing combines, updates, and otherwise augments all previously issued volumes. The intention has been to list recent literature and selected older classics that have a possible interest for those working in this branch of numerical analysis.

See also the supplement by Carpenter. Both reports are available from the National Technical Information Service at the prices $13.60 and $6, respectively.


Loren P. Meissner,
Structured Fortran Bibliography (I),
(Loren P. Meissner (50-B 3239), Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720).

Meissner, Loren P.
Structured Fortran Preprocessor Survey. (Preprocessor update)

Loren P. Meissner,
Fortran Bibliography.
FORWORD, Fortran Development Newsletter, No. 4, August 1975, p. 7.
Edward F. Miller, Jr.,
Document List, Program Validation Project, GENERAL RESEARCH CORPORATION,
5383 Hollister Avenue, P.O. Box 3587, Santa Barbara, California 93105.

MISURI, Giorgio: Survey of Existing Programming Aids.
SIGPLAN Notices, Vol 11, No. 8, August 1976, pp 38-41,
(33 refs).

NTIS: Weekly government abstracts: Computers, Control and Information Theory.


William G. Poole, Jr. and Robert G. Voigt, Bibliography 35.


P.L.J. van Rooij and F. Schurer,
A bibliography on spline functions. II.


TRW: Index to Publications in Print, TRW SOFTWARE SERIES, TRW-SS-Index (cf. Boehm above for address)


WHITEMAN, J R: A Bibliography for Finite Elements. Academic Press, London 1975. 202 pp., £3.50/89.25. The present work cites almost all the significant papers on advances in the mathematical theory; reports on applications; covers such topics as classical analysis, functional analysis, approximation theory, fluids and diffusion; lists finite element packages.

8. DESIRABLE LANGUAGE FEATURES FOR NUMERICAL PROGRAMMING


HULL, T. E. and HOFBAUER, J. J.: Language facilities for numerical computation, 1974, 18 pp. See also in Rice, John R. (ed.): Mathematical software II, Informal Proceedings of a conference, Purdue University, 29-31 May 1974. (pp. 1 - 18)


As the title implies, this article contains undesirable language features for numerical programming.
9. COMMUNICATION LANGUAGES

W. MORVEN GENTLEMAN - University of Waterloo
Interface between Numerical Analysis and Symbolic Computation
In: Rice, John R.
Purdue University, May 29-31, 1974.


JOE THAMES - PROSE, Inc.
PROSE, A Very High Level General Purpose Language
In: Rice, John R.
Purdue University, May 29-31, 1974.

See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.
10. FORTRAN DIALECTS

ALCOA, Aluminium Company of America: Tests of Standard Fortran. Tests programs which facilitate the checking of standard Fortran compilers, developed as ALCOA Engineering Standard 32.6.7.1 - May, 1972, are available from

Dr T J Williams
Purdue Laboratory for Applied Industrial Control
Purdue University
West Lafayette
Indiana 47907

The charge for the card deck is US $20 plus postage (for 15 pounds).

Note: This package can be considered as the opposite of PFORT, which checks the program, and not the compiler.


Available from Försvarsmàlets Forskningsanstalt, National Defence Research Institute, S-10450 Stockholm, Sweden.


A Fortran-based expanded language.


11. TRANSPORTABILITY

T. AIRD, E. BATTISTE and W. GREGORY - IMSL, Inc.
Portability of Mathematical Software Coded in an ANSI Based Fortran

In: Rice, John R.
Purdue University, May 29-31, 1974.


ATKINS, M S: Problems of program portability, paper presented at IUCC Computer Science Colloquium, University College of Swansea, September 1974.


Concentrates on technical and legal problems surrounding programs and structuring data.


DAHLSTRAND, Ingemar: Portabilitet inom teknisk-vetenskaplig ADB (Portability in scientific computations) (In Swedish).


FJORD, Brian and SMITH, Brian T: Transportable Mathematical Software - A Substitute for Portable Mathematical Software. IFIP WG 2.5 Position Paper, January 13-16, Oxford University (16 pp.). (1975)


GROOMS, David W.: Computer software transferability and portability (a bibliography with abstracts). National Technical Information Service (May 1976), 46 pp. NTIS/PS-76/0388/9WC.


Summary
We consider the attainment of portability through the successive refinements of an initial coding attempt. This may be seen as prediction followed by correction. Recent attention has been given to the mechanical selection of software variants stored in a single composite file. We conclude that, under some circumstances, it is also necessary to consider the updating of the file as well as extraction from it.


IFIP WG 2.5: MAP Statement in Fortran to Assist in the Portability of Numerical Software. Manuscript September 10, 1976 (16pp.)

Krogh, F.T., "A language to simplify maintenance of software which has many versions", Computing Memorandum No. 360, Jet Propulsion Laboratory, April 18, 1974.


SCHNEIDER, Ben Ross: Travels in Computerland, or, Incompatibilities and Interfaces. Addison-Wesley 1974.

A book on an outsider's perspective on the world of computing.


GIPZ is a program package in FORTRAN for simple graphics e.g. simple line drawings and data display. It provides very high level subroutines and has been designed and written with portability in mind. This manual describes the design of the system and a user's detailed view of it. Program listings are not included.

Available from Försvarets Forskningsanstalt, National Defence Research Institute, S-104 50 Stockholm 80, Sweden.


WARREN, J.: Software portability, Stanford University Digital Systems Laboratory, Technical Note No. 48 (September 1974).

12. AIDS FOR SOFTWARE GENERATION AND EVALUATION


Stefan Arnborg, Strukturerad induktion och strukturerad programmering - praktiska och opraktiska metoder för programvalidering (Structured induction and structured programming - Practical and impractical methods for program validation) FOA P report C 8402-M3 (E5), May 1974 (In Swedish)

The report gives a summary of formal and informal program validation techniques and structured programming methods. A number of applications are described and a discussion of the relevance of described ideas is included.

Available from Försvarets Forskningsanstalt, National Defence Research Institute, S-104 50 Stockholm, Sweden.


BOEHM, B W: Some steps toward formal and automated aids to software requirements analysis and design.

This paper describes a formalism for requirements analysis and design which includes a properties-oriented formalism for requirements traceability, consistency, and completeness checking, the Requirements/Properties Matrix, and a process-oriented formalism similar to others in being hierarchial and working with black-box descriptions of modules. An initial manual version of the technique has worked successfully in two small-to-medium software projects. It is now being extended to handle large software projects in a semi-automated fashion.


BOEHM, B W; McCLEAN, R K; and URFRIG, D B: Some Experience with Automated Aids to the Design of Large-Scale Reliable Software. Proceedings of the International Conference on Reliable Software, 21-23 April 1975, Los Angeles, California, pp. 105-113.


An account of some experiences with FLOW for debugging and testing large FORTRAN programs for the IBM System/360. FLOW gives information about number of executions of selected statements, number of calls to subroutines, tracing of execution and execution time for each subroutine.

Also available as Report TRW-SS-72-08. (December 1972)


Automated Input/Output Variable Classification as an Aid to Validation of Fortran Programs

In: Rice, John R.
Purdue University, May 29-31, 1974.


This editing program gives a systematic layout of the program and renumbers the statement numbers. It is written in ANS Fortran.


GINSBERG, Myron and FRAILEY, Dennis J: The design and use of a portable system for testing the arithmetic behavior of mathematical software. The 1975 International Conference on Reliable Software.
FDS: A FORTRAN Debugging System Overview and Installer's Guide

Andrew D. Hall

FDS is a comprehensive debugging system designed for use with Honeywell Series 600/6000 FORTRAN IV. The main features include a symbolic post-mortem dump, an interactive debugging system, and an automatic subprogram measurement system. All facilities may be used in either the batch or the timesharing environments of GCOS. This paper gives a brief overview of the capabilities and implementation of FDS and provides instructions for installing the system in GCOS release SR-F.


HITACHI Ltd, Tokyo: Fortran Analyzer. FORTRAN IV (93 percent), ASSEMBLER (7 percent) 10,887 source statements. IBM 370. COS-02510 Price: Program $970.00/Documentation $9.50.


THOMAS E. HULL - University of Toronto
Language Facilities for Numerical Computation

In: Rice, John R.
Mathematical Software II, Informal Proceedings of a
Conference.
Purdue University, May 29-31, 1974.

IEEE: Proceedings of the Workshop "Currently Available
Program Testing Tools: Technology and Experience".
Los Angeles, April 24-25, 1975. Arranged by the IEEE Computer
Society Technical Committee on Software Engineering.

INGALLS, D. H. H.: FETE—a Fortran execution time estimator,

IRVINE, C. A. and BRACKETT, John W.: Automated software engineering
through structured data management, IEEE Transactions on Software
SEF is a system designed to support the development of well-engineered
software. It provides support for an integrated collection of subsystems.

JESSOP, W. H. et al.: ATLAS, an automated software testing system,
Proceedings of the 2nd International Conference on Software Engineering
(13-15 October 1976), San Francisco, California. Available from ACM or
IEEE, Catalog No. 76 CH1125-4C, pp. 629-635.

JET PROPULSION LAB, California Institute of Technology,
Pasadena: Structured Programming-to-Fortran Translator:
SPTRAN. SPTRAN 680 cards, UNIVAC 1100, NPO-13602. Price:
Program $380.00/Documentation $9.00. Ref: Computer Program

KENNEDY, K. W.: Node listing applied to data flow analysis, Conference
record of the 2nd ACM Symposium on Principles of Programming Languages,

Summary
Although Fortran is not a pleasant language to use, it does have the advantages of universality and (usually) relative efficiency. The RATFOR language attempts to conceal the main deficiencies of Fortran while retaining its desirable qualities, by providing decent control flow statements and some 'syntactic sugar'. RATFOR is implemented as a preprocessor which translates this language into Fortran.


FRED T. KROGH - Jet Propulsion Laboratory
A Language to Simplify Maintenance of Software which has Many Versions

In: Rice, John R.
Purdue University, May 29-31, 1974.
Also available as Computing Memorandum No. 360, JPL (Section 914).


This paper describes our experience in using a common set of code generating subroutines as the basis for compilers of ALGOL W, ALGOL 60 and subsets of PL/I and COBOL. The advantages of using a common set of primitives for implementing more than one compiler include: (1) greatly reduced effort to implement additional languages; (2) simplified maintenance of the generated compilers; (3) standardized diagnostic messages for all compilers implemented using this system; and (4) simplified documentation of the compilers.

A method for estimating the number of errors remaining in a software package is proposed and analyzed. It is based upon a scheme proposed by H D Mills of IBM, in which a set of known errors is "seeded" into the software. A specified number of tests is conducted, each test capable of finding one of the indigenous or unknown errors, or one of the seeded errors with the same probability, or of finding no error. Report TRW-SS-72-09.

Summary
A recent revival of interest in measuring program execution behaviour has led to a number of distinct approaches. Arguments are given for a fairly simple method of modifying FORTRAN source code to collect frequency counts. No symbol table is necessary and only a single reserved name is introduced into the source.


Certain types of errors in the coding of FORTRAN programs can be detected by careful analysis of the input/output usage of the variables in the program. Algorithms are presented which employ depth-first searching techniques to verify whether input uses and output uses are improperly interspersed. These algorithms can also be used to determine the input and output parameters for entire subprograms. This capability extends the usefulness of these verification techniques, and can also be used in the attempts to automate documentation production.


RAMAMOORTHY, C. V. and HO, S. F.: FORTRAN automated code evaluation system, Electronics Research Laboratory, University of California, Berkeley, Memo ERL-M466 (July 1974).


RAMAMOORTHY, C. V. et al.: The status and structure of software testing procedures, Digest of Papers, Comp Con Spring 77, IEEE Catalog No. 77CH1165-OC, pp. 367-369.


ROCKWELL International Corp., Canoga Park, California, Space Div.: Indices and cross references from computer readable text. PLI 97 source statements, IBM 360, MSC-19423. Price: Program/Documentation $ 50.00.

This program was developed to provide indices and cross reference tables from computer readable texts. The program will produce an index on selected words or phrases. The program will search the text and reference all occurrences of the specific words or phrases used as search keys. This program was configured to process administrative terminal service (ATS) generated texts. The program will operate with time share option (TSO) or on batch processing computers. Ref: Computer Program Abstracts Vol 7(3) 1975.


The PFORT Verifier: User's Guide  
B. G. Ryder and A. D. Hall  
The PFORT Verifier is a program which checks a FORTRAN program for adherence to PFORT, a portable subset of ANSI FORTRAN. It diagnoses errors in inter-program-unit communication and COMMON usage which compilers often miss. The Verifier itself is written in PFORT and can easily be installed on a variety of computers. This paper describes the use of the Verifier and presents the portable subset in considerable detail. This is a revised version of Computing Science Technical Report #12, May 1973.

Computing Science Reports, Room 2C-576, Bell Laboratories, Murray Hill, New Jersey 07974.


This paper describes a system which checks correctness of array accesses automatically without any inductive assertions or human interaction.


Summary
It is shown that it is possible to gather run-time statistics from a program without either slowing it down too much, or changing things such as the hardware or the operating-system kernel.


-64-
13. **MACHINE CONSTANTS**


IFIP WG 2.5: Parameters for Transportable Numerical Software. Manuscript July 1, 1976. Submitted to IFIP for publication.


ÖSTLIRBY, Ole: Forelæsningsnoter til Matematik B, Regneringskunst (Lecture Notes in Mathematics, Numerical Accuracy) (In Danish). Department of Mathematics, Aarhus University, DK-8000 Aarhus C, Denmark (1975).
14. HARDWARE FEATURES


For scientific computations on a digital computer the set of real numbers is usually approximated by a finite set F of "floating-point" numbers. We compare the numerical accuracy possible with different choices of F having approximately the same range and requiring the same word length. In particular we compare different choices of base (or radix) in the usual floating point systems. The emphasis is on the choice of F, not on the details of the number representation or the arithmetic, but both rounded and truncated arithmetic are considered. Theoretical results are given, and some simulations of typical floating-point computations (forming sums, solving systems of linear equations, finding eigenvalues) are described. If the leading fraction bit of a normalized base-2 number is not stored explicitely (saving a bit), and the criterion is to minimize the mean square roundoff error, then base 2 is best. If unnormalized numbers are allowed, so the first bit must be stored explicitely, then base 4 (or sometimes base 8) is the best of the usual systems.


The appearance of hexadecimal floating-point arithmetic systems has prompted a continuing discourse of the relative numerical merits of various choices of base. Until lately this discourse has centered about the static properties of floating-point representation of numbers, and has primarily concerned only binary and hexadecimal representations. Recent events may change this discourse considerably. A third numerically attractive alternative for the choice of base has been proposed, and a comparison of the dynamic numerical properties of floating-point arithmetic systems has been completed. This paper surveys these recent events and summarizes our current knowledge of the numerical characteristics of floating-point number systems.
Discuss the choice of a suitable work length, basis and rounding principles. A simple example shows that the base 2 together with rounding is to be preferred.


DEKKER, T: Machine Requirements for Reliable, Portable Software. See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.


Comments on the paper by Malcolm [1972] : "Algorithms to Reveal Properties of Floating Point Arithmetic". The algorithms in that paper has to be changed to some extent to work well on some machines. Results from using Malcolm's routines are cited.


KAHAN, W: The Floating Point Processor on a chip or two. See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.


KAHAN, W. and PARLETT, B. N.: Can you count on your calculator. Submitted to Electronics.


Two algorithms are presented in the form of FORTRAN subroutines. Each subroutine computes the radix and number of digits in the floating numbers and whether rounding or chopping is done on the machine on which it is run. The methods are shown to work on any "reasonable" floating point computer.

See also Gentleman and Marovich (1974).


Impact of Future Computer Architecture on Mathematical Software

In: Rice, John R.
    Mathematical Software II, Informal Proceedings of a Conference,
    Purdue University, May 29-31, 1974.


15. CRITERIA FOR EVALUATING NUMERICAL SOFTWARE

a) GENERAL ARTICLES


BIELSKI, John P. and BLANKERTZ, William H.: The general acceptance test system (GATS), Digest of Papers, Comp Con Spring 77, IEEE Catalog No. 77CH1165-0C, pp. 207-210.


Summary

This paper describes an experimental evaluation of the execution speed of object programs produced by six PL/I compilers. An analysis is made of the relative speed of the object code produced by each compiler and also of the relative execution speed of each statement.

15. CRITERIA FOR EVALUATING NUMERICAL SOFTWARE

b) GUIDELINES FOR TRANSPORTABLE EVALUATION TOOLS

REINSCH, Christian H: Building a library of numerical algorithms: A case study from the handbook "Linear Algebra". IFIP WG 2.5 Position Paper, January 13-16, 1975, Oxford University, 6 pp.

c) CRITERIA FOR EVALUATING QUADRATURE, ORDINARY DIFFERENTIAL EQUATION AND INTEGRAL EQUATION SOFTWARE


Factors involved in the development of good software are discussed, with practical reference to programs for solving ODE's. These factors include the basic structuring of the programs themselves, along with the appropriateness of various language facilities, comparisons of efficiency, proofs of correctness, certification and distribution etc.

16. EVALUATIONS

a) GENERAL ARTICLES AND MISCELLANEOUS EVALUATIONS


More detailed results are available as "User Ratings of Proprietary Software" for $12 from Datapro Research Corporation, 1805 Underwood Boulevard, Delran, New Jersey 08075.


LOESER, R. [1974]: Some Performance Test of "quicksort" and Descendants CACM Vol 17(3), pp. 143 - 152.

Detailed performance evaluations are presented for six ACM algorithms: quicksort (No 64), Shellsort (No 201), stringsort (No 207), "TREESORTS" (No 245), quickersort (No 271) and qsort (No 402). Algorithms 271 and 402 are refinements of algorithm 64, and all three are discussed in some detail. The evidence given here demonstrates that qsort (No 242) requires many more comparisons than its author claims. Of all these algorithms, quickersort requires the fewest comparisons to sort random arrays.


Several hundred college and university computer installations now offer various types of statistical packages for general use. Among those most widely available are OSIRIS, SPSS, EMD, DATA-TEXT and TSAR. In order to provide users with a basis for selection and use, tests were made for each of the systems, and the results are summarized as to cost and performance.


16. EVALUATIONS

b) LINEAR ALGEBRA


A. BLASER and U. SCHAUER - IBM Deutschland GMBH
Performance Measurements on the Sparse Matrix Subroutines of SL-MATH

In: Rice, John R.
Purdue University, May 29-31, 1974.


C. L. LAWSON - Jet Propulsion Laboratory
Standardization of Fortran Callable Subprograms for Basic Linear Algebra

In: Rice, John R.
Purdue University, May 29-31, 1974.


An algorithm is presented for computing exactly general solutions for systems of linear equations with integer or polynomial coefficients.

A computing time study of several algorithms for the exact solution of systems of linear equations with integer or polynomial coefficients is presented. The analytical computing times for rational Gauss elimination, exact division elimination (one-step and two-step), and the modular algorithm are summarized and supplemented. Extensive empirical studies illustrate the superiority of the modular algorithm in agreement with the analytical results. All algorithms were programmed in FORTRAN IV for the SAC-1 System and all cases were run on a UNIVAC 1108.

CLEVE B. MOLER - University of New Mexico
Software for Matrix Computations

In: Rice, John R.
Purdue University, May 29-31, 1974.

JAMES PANTTAJA - Univer. of California - Berkeley
A Comparison of the PLU and QR Methods for Determining Eigenvalues of Real Hessenberg Matrices

In: Rice, John R.
Purdue University, May 29-31, 1974.


An algorithm is developed for obtaining eigenvalues of real, symmetric, tridiagonal matrices. It combines dynamically Given's method of bisection and the use of Sturm sequences with various acceleration devices. A FORTRAN IV computer implementation of the algorithm was used on ten test matrices found in the literature. The new method is as precise and reliable as the best published program (Kahan and Varah, 1966); it is never slower, and in at least one case is two and a half times faster than the Kahan and Varah program.

16. EVALUATIONS

c) QUADRATURE


de BOOR, Carl, On Writing an Automatic Integration Algorithm, pp. 201 - 209.


EINARSSON, Bo: Testing and evaluation of some subroutines for numerical quadrature, pp. 149 - 157.


Jones, R., Results of some quadrature tests, unpublished (1973).


Several methods for general purpose numerical quadrature are considered. Selection criteria are total time, including bookkeeping time and function evaluation time, and accuracy of the estimate. The effectiveness of several new methods has been compared with the adaptive Simpson's rule and under certain circumstances at least two of these new methods perform better.

This report is a preliminary version of the article with the same title in Rice [1971]. The report contains more formulas and more text than the article, which mostly consists of computer print-outs.


KROGH, F T and SNYDER, W V: Preliminary Results with a New Quadrature Subroutine. Jet Propulsion Laboratory, Section 914, Computational Memorandum No. 363 (Revised April 1975), Pasadena, California.


Results of numerical experiments involving different automatic quadrature routines and problem families. The method used as described in two articles by Lyness and Kaganove.


Patterson, T. N. L., Algorithm 468: Algorithm for automatic numerical integration over a finite interval. 

Piessens, R., An algorithm for automatic integration, 

Piessens, R., and DeDonker, E., A Bibliography on 
Automatic Integration, Report TW 26, Applied Mathematics 
and Programming Division, Katholieke University, Leuven 
(August 1975). Celestynenlaan 200A, B-3030, Heverlee, 
Belgium. (For published version see DeDoncker in section 7)

Treats only sine and cosine oscillations. Sixteen test integrands are used in the calculation of the quality Q. Extensive listings are included.


Section 3, The Selection of Quadrature Algorithms, pp. 77-82, discusses ten different evaluations.


16. EVALUATIONS

d) ORDINARY DIFFERENTIAL EQUATIONS


Contains an extensive bibliography and special information on report and code availability.


W. ENRIGHT, R. BEDET, G. HALL, T. HULL and B. LINDBERG - University of Toronto

On the Evaluation of Numerical Methods for Initial Value Problems in Ordinary Differential Equations

In: Rice, John R.
Purdue University, May 29–31, 1974.


A study comparing the performance of several computer programs for integrating systems of ordinary differential equations (initial value problems) is reported. The integration methods represented include multistep methods (predictor-correctors), single-step methods (Runge-Kutta) and extrapolation methods (both polynomial and rational). The testing procedure is described together with the evaluation criteria applied. A set of 7 test problems on which the programs were tested is included in an appendix. For the particular problems and criteria used in the investigation it was found that a program based on rational extrapolation showed best performance. 25 references are given.


Numerical methods for systems of first order ordinary differential equations are tested on a variety of initial value problems. The problems, methods and comparison criteria are specified very carefully.


P. KEMP - Cambridge University
Mathematical Software for Ordinary Differential Equations: Chebyshev Series Methods

In: Rice, John R.
Purdue University, May 29-31, 1974.


This paper discusses how to numerically test a subroutine for the solution of ordinary differential equations. Results obtained with a variable order Adams methods are given for eleven simple test cases.


-85-

A review of popular numerical methods for solution of initial value problems for systems of ordinary differential equations, existing programs and recommendations.


A. SILVER and E. SULLIVAN - Goddard Space Flight Ctr.
The Numerical Solution of Ordinary Differential Equations by the Taylor Series Method

In: Rice, John R.
Purdue University, May 29-31, 1974.

WALSH, Joan: Initial and boundary value routines for ordinary differential equations, pp. 177 - 189.


-86-

16. EVALUATIONS

e) FUNCTIONS

D. AMOS, S. DANIEL and M. WESTON - Sandia Laboratories
CDC 6600 Subroutines IBESS and JBESS for Bessel
Functions I_\nu(x) and J_\nu(x), x \geq 0, \nu \geq 0

In: Rice, John R.
Mathematical Software II, Informal Proceedings of a
Conference.
Purdue University, May 29-31, 1974.
To be published in ACM Transactions on Mathematical Software.

BRENT, R P: Fast Multiple-Precision Evaluation of Elementary
Functions. J. Association for Computing Machinery, 23 (April,

CLARK, N A; CODY, W J; HILLSTROM, K E and THIELEKER, E A:
Performance Statistics of the FORTRAN IV (H) LIBRARY for
the IBM SYSTEM/360. SICNUM Newsletter Vol 2, No. 3, December

CLENSHAW, C. W.: Rational approximations for special functions,
pp. 275 - 284.
Proceedings of the Loughborough University Conference of
the Institute of Mathematics and Its Applications held

CODY, W. J. [1969]: Performance Testing of Function Subroutines,
Discussion about possibilities for testing of routines for computa-
tion of functions. The method recommended, which also is well
described, is based on the computation of both the function to be
tested and a routine in extended precision for random arguments.

CODY, W. J.: Software for the Elementary Functions,
pp. 171 - 186.
In RICE, John R (Ed): Mathematical Software. Based on the
proceedings of the Mathematical Software Symposium held
at Purdue University, Lafayette, Indiana, April 1970.


FULLEERTON, W: Absolutely Portable Special Function Routines. See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.


KEMP, P: Writing Elementary Function Procedures for the ALGOL 68C Compiler. See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.


Describes a very careful way of testing function routines at the implementation on different computers.


SCHONFELDER, J: The Production and Testing of Special Function Software in the NAG Library. See COWELL, Wayne R (Editor) in section 2, Books related to Numerical Software.


16. EVALUATIONS

f) OPTIMIZATION


Contains a review of some modern methods for non-linear optimization and directions where to find programs for these methods.


Comparison between PRAXIS (Powell's conjugate axis method), STREAM (Rosenbrock method) and MIGRAD from MINUIT (CERN). For the class of functions on which the minimizers in this study were tested there is no doubt - PRAXIS is, by far, superior.
16. EVALUATIONS

g) INTEGRAL EQUATIONS


L. M. DELVES - University of Liverpool
An Automatic Ritz-Galerkin Procedure for the Numerical Solution of Linear Fredholm Integral Equations of the Second Kind


17. FORMAL PROOFS OF CORRECTNESS


COUSOT, P. and COUSOT, R.: Static verification of dynamic type properties of variables, Research Report No. 25, Laboratoire d'Informatique, Grenoble, France.


This report describes the Euclid language, intended for the expression of system programs which are to be verified. Euclid draws heavily on Pascal.


LUCKHAM, David and SUZUKI, Norihisa: Automatic program verification V: verification-oriented proof rules for arrays, records, and pointers. Dept. of Computer Science, Stanford University (March 1976), 52 pp., Report No. STAN-CS-76-549. Also available from NTIS as AD-A027 455/5WC.

MILLER, Edward F: RXVP, Fortran Automated Verification System, Level 1, System Summary October 1974, Program Validation Project, GENERAL RESEARCH Corporation, P.O.Box 3587, Santa Barbara, California 93105.


18. STRUCTURED PROGRAMMING


Discusses structured programming in connection with FORTRAN programming.

HULL, T E: Correspondence on Structured Fortran. SIGNUM Newsletter Vol 9, No. 2, April 1974, p. 22.


Loren P. Meissner,
Structured Fortran Bibliography (I).
(Loren P. Meissner (50-B 3239), Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720).

Meissner, Loren P.
Structured Fortran Preprocessor Survey. (Preprocessor update)


The December 1976 issue (No. 12) contains on pp. 1-9 several letters commenting on this article.


Discusses structured programming with special emphasis on Fortran.


Contributions:
Bridge, R F and Thomson, E W. BRIDGES - A Tool for Increasing the Reliability of References to FORTRAN Variables.
Cook, A James. Experience with Extensible, Portable FORTRAN Extensions.
Wells, Mark B. Preprocessing of Typed Two-Dimensional Mathematical Expressions.

Describes a suggested programming language, upward compatible with Fortran IV, and which contains newly defined data and control structures.

Contains a list of 55 preprocessors.


19. TEST PROBLEMS

Most references in section 16 (Evaluations) also contain test problems.


BRENNER, J. L.: A set of test matrices for testing computer programs, CACM Vol 5, pp. 443 - 444.


Contains a chapter on the construction of test matrices and a large number of matrices with inverses, determinants, eigenvalues and eigenvectors given. 81 references.


RICE, John R: A set of 74 test functions for nonlinear equation solvers, Report CSD-TR 34, Computer Science Department, Purdue University, Lafayette, Indiana, April 1969.

Contains in an appendix 41 test matrices together with their inverses and/or eigenvalues.


ZIELKE, G: Testmatrizen mit freien Parametern. Computing 15 (2), 1975, pp. 87-103. (In German)
Test Matrices with Free Parameters.
A method to construct test matrices with free parameters is given. As an example of application a new test matrix of even order which is also appropriate to the eigenproblem is derived. Moreover some known test matrices with constant elements are generalized in such a way that they can be supplied with condition numbers of arbitrary order.
20. MISCELLANEOUS

ABERTH, O [1974]: A Precise Numerical Analysis Program. CACM 17(9), 509 - 513.

A description is given of a program for computing the solution to a small number of standard numerical analysis problems to any specified accuracy, up to a limit of 2000 correct decimal places. Each computed number is bounded in an interval with a multiple precision midpoint. Arithmetic operations involving these numbers are executed according to interval arithmetic concepts, with non-significant digits automatically discarded. Details are supplied of problem specification and computation.


Shows how the different FORTRAN functions and some mathematical software were used at the Purdue University Computer Center.

C. BAILEY and R. JONES - Sandia Laboratories
Usage and Argument Monitoring of Mathematical Library Routines

In: Rice, John R.
Purdue University, May 29-31, 1974.


Survey with extensive bibliography.


ELSPAS, Bernard; LEVITT, Karl N; WALDINGER, Richard J
and WAKSMAN, Abraham: An Assessment of Techniques for
Proving Program Correctness. Computing Surveys, Vol 4,
No. 2, June 1972, pp. 97-147.

IEEE Transactions on Software Engineering, Vol SE-1, No. 4, December
1975, pp. 350-357.

FELDSTEIN, A and GOODMAN, R: Convergence Estimates for the
Distribution of Trailing Digits. J. Association for Computing

FORSYTHE, G. E. [1970]: Pitfalls in Computations, or Why a Math
Book Isn't Enough, Stanford University Technical Report CS-147
A splendid introduction for the layman to the problems of numerical
computations. Originally written for a mathematical congress, it
is well worth reading also for non-professional mathematicians.

FOSDICK, Lloyd D.: Proceedings of the Conference on the Validation and
Distribution of Computer Software, 30-31 March 1972, Boulder, Colorado,
Technical Report No. CU-CS-004-72, Department of Computer Science,
University of Colorado, Boulder, Colorado 80309.

FOSDICK, Lloyd D. [1972]: The Production of Better Mathematical
A well written summary of problems and possibilities for document-
tation, standardization and evaluation of mathematical software.
Contains extensive bibliography.

FRANCIS, Ivor; HEIBERGER, Richard M and VELLEMAN, Paul r:
Report and Proposal of the Committee on Evaluation of Program
Packages to the Section on Statistical Computing, August 1974,

FRITSCH, Fred N.: Profile on the Numerical Mathematics Section of
the Lawrence Livermore Laboratory. SIGNUM Newsletter, Vol 8(4),
pp. 9 - 10.
Contains a description of the work at the laboratory and a list
of reports.


GAULDING, Scott N.: A software design methodology and tools, Digest of Papers, Comp Con Spring 77, IEEE Catalog No. 77CH1165-OC, pp. 198-201.

C. WILLIAM GEAR - University of Illinois
What Do We Need in Programming Languages?

In: Rice, John R.
Purdue University, May 29-31, 1974.


JANSON, R J; kROGhi, F T and LAWSON, C L: Improving the Efficiency of Portable Software for Linear Algebra. SIGNUM Newsletter Vol 8, No. 4, October 1973, p. 16.


Discusses 7 different possibilities for error exits from routines for computations of functions.

ELLIS HOROWITZ - University of Southern California
A Sorting Algorithm for Polynomial Multiplication
Page 45
-83

In: Rice, John R.
Purdue University, May 29-31, 1974.


HUANG, J C: Program Testing. (May 1974) Department of Computer Science, University of Houston, Texas.


HULL, T. E. et al [1972]: The Correctness of Numerical Algorithms, Proc. of the Conf. on Proving Assertions About Programs, Jan. 1972. Among the numerous articles on proving the correctness of a given algorithm, this is the first to cover also the numerical aspects (e.g. rounding errors). Examples are given from linear algebra and ordinary differential equations.

IGNIZIO, James P. [1973]: Validating Claims for Algorithms Proposed for Publication. Oper. Res. Vol 21, pp. 852 - 854. Calls attention to the problems of validating the claims made for algorithms in published papers, and proposes a system for review that will validate such claims thoroughly before publication.


W. M. KAHAN - University of California, Berkeley
The Table-Makers' Dilemma and other Quandaries
In: Rice, John R.
Purdue University, May 29-31, 1974.


MALCOLM, Michael A. [1973]: A Machine-independent ALGOL Procedure for Accurate Floating-Point Summation, Stanford University, Calif., STAN-CS-73-374. Describes an ALGOL 60 procedure which is an implementation of the floating-point summation technique described in Malcolm (1971). This implementation is machine-independent in a certain sense.


MAURER, W. D.: Software systems design and correct software, Digest of Papers, Comp Con Spring 77, IEEE Catalog No. 77CH1165-OC, pp. 194-197.


JOEL MOSES - Massachusetts Institute of Technology
The Evolution of Algebraic Manipulation Algorithms
In: Rice, John R.
Purdue University, May 29-31, 1974.

MUSA, John D: A theory of software reliability and its application.


NAG Project Note Number 5: Draft Specification for TEST PROGRAM CONVENTIONS, 7 pp.


RICE, John R: Algorithmic Progress in Solving Partial Differential Equations. Computer Science Department, Report CSD-TR 173, January 1976, Purdue University, Lafayette, Indiana 47907, USA.


THAYER, T. A. et al.: Software reliability study. TRW Defense and Space Systems Group, Redondo Beach, California (August 1976), 343 pp., TRW-76-2266.1.9-5, Available from NTIS as AD-A030 798/3WC.


Numerical mathematics is viewed as the analysis of continuous algorithms. Four of the components of numerical mathematics are discussed. These are: foundations (finite precision number system), synthesis and analysis of algorithms, analysis of error, programs and program libraries.
Generalized logarithmic law is derived for the distribution of the first t significant digits of a random digital integer. This result is then used to determine the distribution of the roundoff errors in floating point operations, which is a mixture of uniform and reciprocal distributions.


VICTOR A. VYSSOTSKY - Bell Telephone Laboratories
Some Problems in Large Software Systems

In: Rice, John R.
Purdue University, May 29-31, 1974.

WANG, Arne: An Axiomatic Basis for Proving Total Correctness of Goto-Programs. BIT 16 (1976), pp 88-102.


-115-
WICHMANN, B A and JONES, B: Testing ALGOL 60 Compilers.
SOFTWARE - Practice & Experience, Vol 6, (2), 1976,
pp 261-270.

WORTMAN, David B.: Notes from a workshop on the attainment of reliable
software, University of Toronto, Toronto, Canada, 51 pp.
Report from June 1974 workshop in Toronto, arranged by ACM and
IEEE. This report is available as R76-276 from IEEE Repository.

RICHARD ZIPPEL - Massachussetts Institute of Technology Page 191
Power Series Expansions in MACSYMA

In: Rice, John R.
Mathematical Software II, Informal Proceedings of a
Conference.
Purdue University, May 29-31, 1974.
21. JOURNALS RELATED TO NUMERICAL SOFTWARE

The following journals publish algorithms and/or computer codes within the numerical software area:

ACM Transactions on Mathematical Software (Association for Computing Machinery, 1133 Avenue of the Americas, New York, New York 10036)

Angewandte Informatik (Friedr. Vieweg + Sohn GmbH, Burgplatz 1, D-3300 Braunschweig, German Federal Republic)

BIT (DATA A/S, Kronprinsensgade 14, DK-1114 Copenhagen K, Denmark)

Communications of the ACM (Association for Computing Machinery, 1133 Avenue of the Americas, New York, New York 10036)


Computer Physics Communications (North-Holland Publishing Company, P.O.Box 103, Amsterdam W, The Netherlands)

Computing (Springer-Verlag, P.O.Box 367, A-1011 Vienna, Austria)

Journal of Computational and Applied Mathematics (Koninklijke Vlaamse Ingenieursvereniging, Jan van Rijswijcklaan 58, B-2000 Antwerp, Belgium)

Numerische Mathematik (Springer-Verlag, Heidelberger Platz 3, D-1 Berlin 33, German Federal Republic)
The following are important review journals:

Computing Reviews (Association for Computing Machinery, 1133 Avenue of the Americas, New York, New York 10036). This journal also has the annual "Bibliography and Subject Index of Current Computing Literature".


Quarterly Bibliography of Computers and Data Processing, Applied Computer Research, P.O. Box 9280, Phoenix, Arizona 85068.