A Brief History of Numerical Libraries

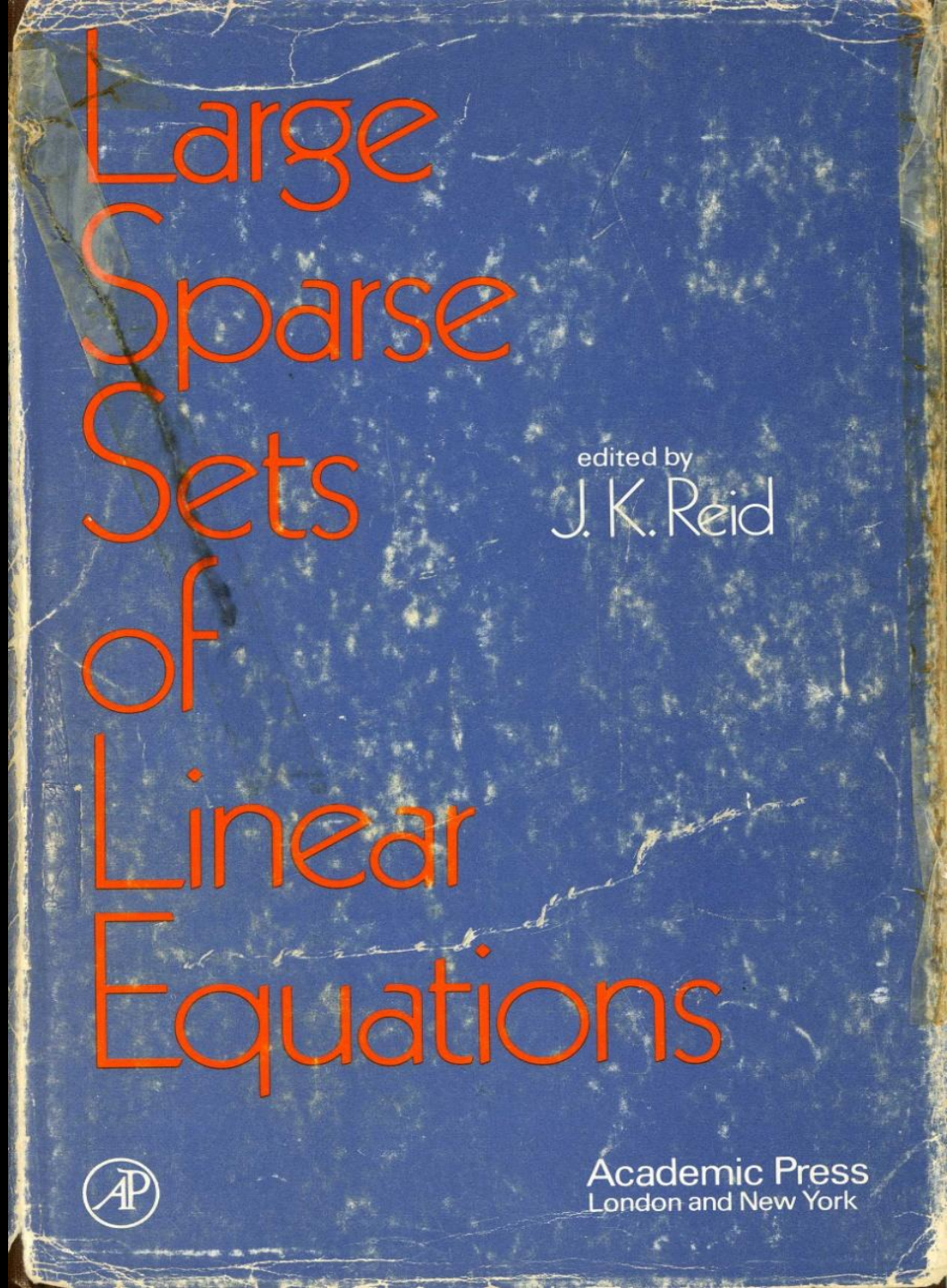
Sven Hammarling
NAG Ltd, Oxford
&
University of Manchester

John Reid at Gatlinburg VII (1977)

"Solution of large finite element systems of linear equations out of main storage"

Proceedings of IMA Conference, Oxford, 1970

Published 1971



John's Article on CG

On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations

J. K. Reid

Mathematics Branch, Atomic Energy Research Establishment Harwell, England.

1. Introduction

The method of conjugate gradients has been known for some time, having been developed independently by E. Stiefel and by M. R. Hestenes with the co-operation of J. B. Rosser, G. Forsythe and L. Paige, but it has received little attention recently. It is difficult to see why this has been so since the method has several very pleasant features when regarded not as a direct method for the solution of full systems of equations but as an iterative method for the solution of large and sparse systems. It is our purpose here to explain these features and to report on some numerical experiments which compare the various versions of the algorithm that are available.

2. The Algorithm and its Variants

We will follow the notation of Hestenes and Stiefel [1] because their original algorithm has advantages over other versions of the method that have been proposed subsequently. Given a system

$$Ax = b \tag{2.1}$$

of n linear equations whose matrix A is symmetric and positive-definite we take a starting vector x_0, form the corresponding residual

$$r_0 = b - Ax_0, \tag{2.2}$$

set $p_0 = r_0$ and then for $i = 0, 1, 2, \ldots$ find the vectors x_{i+1}, r_{i+1} and p_{i+1} and the scalars a_i and b_i by using the recursions

$$a_i = \begin{cases} \text{either } (p_i, r_i)/(p_i, Ap_i) & \text{(2.3a)} \\ \text{or } \quad (r_i, r_i)/(p_i, Ap_i) & \text{(2.3b)} \end{cases}$$

John's CG Article

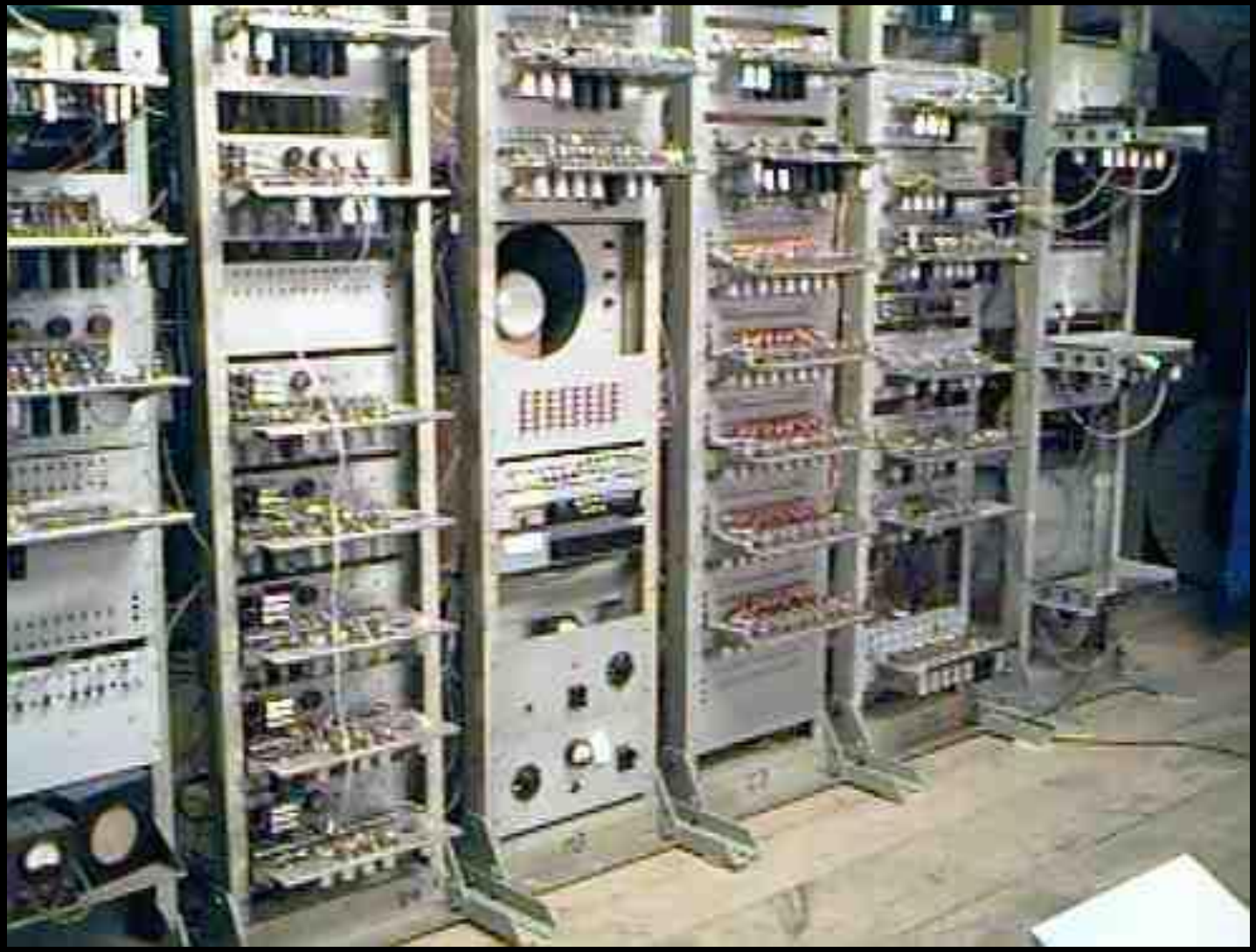
"The paper of Reid (1971) drew the attention of many researchers to the potential of the algorithm as an iterative method for sparse linear systems. It was a catalyst for much of the subsequent work in conjugate gradients."

Golub and O'Leary, SIAM Review, 1989

Rough Outline

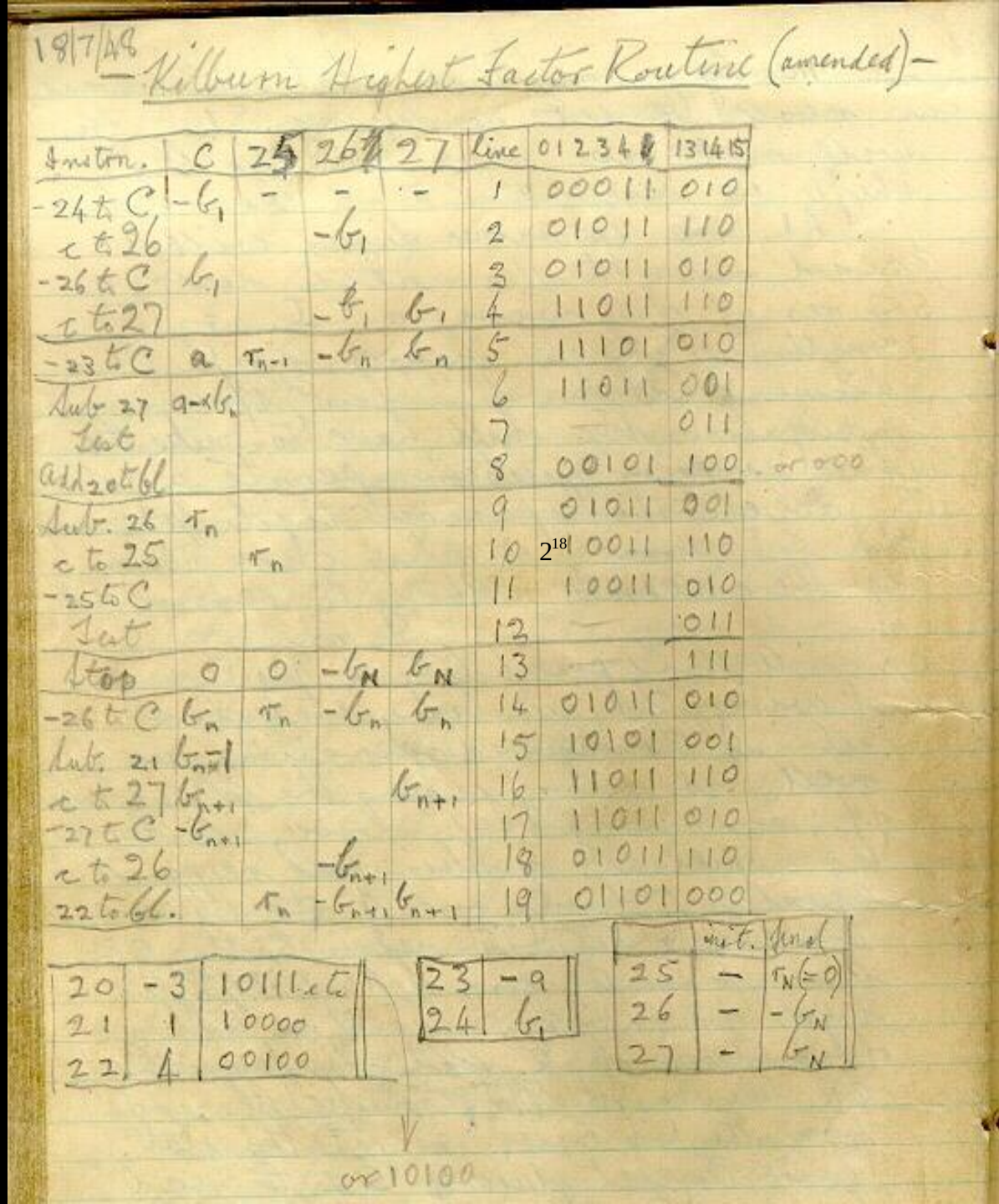
- History
- Libraries, packages and their facets
- Fortran
- Floating Point Arithmetic
- Proceedings and Books
- Summary

Manchester Baby, 21 June 1948 (Replica)



Kilburn/Tootill Program to compute the highest proper factor

2^{18} took 52 minutes
1.5 million instructions
3.5 million store accesses



First Numerical Library

Wilkes, Wheeler and Gill (1951) “Preparation of Programmes for an Electronic Digital Computer” Addison-Wesley

With special reference to the EDSAC and the use of a library of subroutines

“Short, ready-made programs for performing the more common computing operations … are usually called subroutines” (page 1)

Wilkes, Wheeler and Gill

For V:

$z \leftarrow x \pm y$

$y \leftarrow Ax, \quad A = A^T$

PART II

SPECIFICATIONS OF LIBRARY SUBROUTINES

Each subroutine is distinguished by a letter denoting its category and a serial number within that category. The categories are as follows.

Category	Subject
A	Floating point arithmetic.
B	Arithmetical operations on complex numbers.
C	Checking.
D	Division.
E	Exponentials.
F	General routines relating to functions.
G	Differential equations.
J	Special functions.
K	Power series.
L	Logarithms.
M	Miscellaneous.
P	Print and layout.
Q	Quadrature.
R	Read (i.e., Input).
S	nth root.
T	Trigonometrical functions.
U	Counting operations.
V	Vectors and matrices.

In the specifications on succeeding pages the following information is given in abbreviated form immediately beneath the title of each subroutine:

1. Type of subroutine, i.e., whether open, closed, interpretive, or special.
2. Restriction on address of first order. If the word "even" appears it denotes that the first order must have an even address; if no note appears it indicates that the address may be either odd or even.
3. Total number of storage locations occupied by the subroutine.
4. Addresses of any storage locations needed as working space by the subroutine.
5. Approximate operating time (not possible to state in all cases).

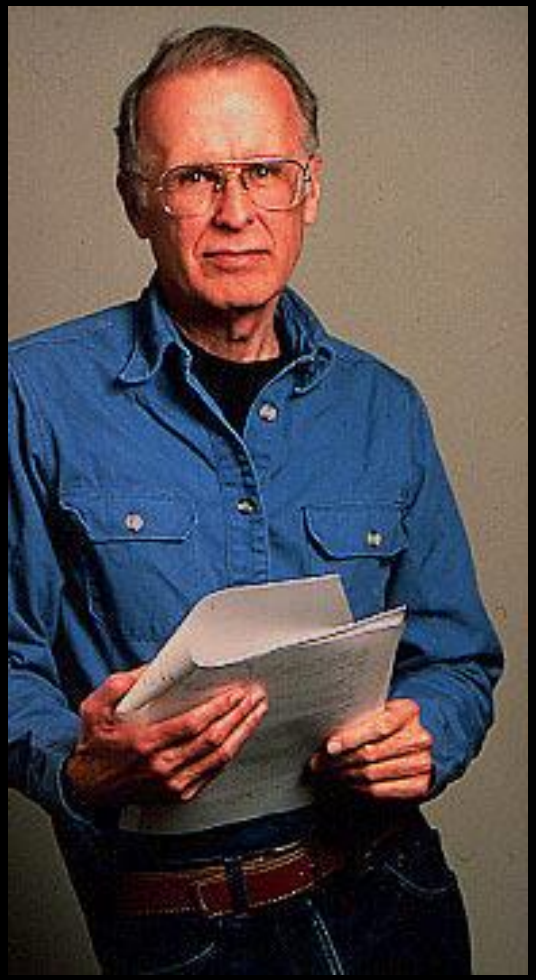
The gaps in the numbering within each category correspond to subroutines which have become obsolete.

Quality Numerical Software

- Should be:
  - Numerically stable, with measures of quality of solution
  - Reliable and robust
  - Accompanied by test software
  - Useful and user friendly with example programs
  - Fully documented
  - Portable
  - Efficient

John Backus

1924 - 2007



Best regards, Walter
John Backus

Programmer's
Reference Manual
October 15, 1956

THE FORTRAN AUTOMATIC CODING SYSTEM FOR THE IBM 704 EDPM®

This manual supersedes all earlier information about the FORTRAN system. It describes the system which will be made available during late 1956, and is intended to permit planning and FORTRAN coding in advance of that time. An Introductory Programmer's Manual and an Operator's Manual will also be issued.

APPLIED SCIENCE DIVISION
AND PROGRAMMING RESEARCH DEPT.
International Business Machines Corporation
590 Madison Ave., New York 22, N. Y.

WORKING COMMITTEE

J. W. BACKUS
R. J. BEEBER
S. BEST
R. GOLDBERG
H. L. HERRICK
R. A. HUGHES
University of California Radiation Laboratory, Livermore, Calif.
L. B. MITCHELL
R. A. NELSON
R. NUTT
United Aircraft Corp., East Hartford, Conn.
D. SAYRE
P. B. SHERIDAN
H. STERN
I. ZILLER

C ← FOR COMMENT / STATEMENT NUMBER (1–5)	CONTINUATION (6)	FORTRAN STATEMENT (7–72)	IDENTIFICATION (73–80)
C		PROGRAM FOR FINDING THE LARGEST VALUE	
C	X	ATTAINED BY A SET OF NUMBERS	
		DIMENSION A(999)	
		FREQUENCY 30(2,1,10), 5(100)	
		READ 1, N,.(A(I), I = 1,N)	
1		FORMAT (I3/(12F6.2))	
		BIGA = A(1)	
5		DO 20 I = 2,N	
30		IF (BIGA-A(I)) 10,20,20	
10		BIGA = A(I)	
20		CONTINUE	
		PRINT 2, N, BIGA	
2		FORMAT (22H1THE LARGEST OF THESE I3, 12H NUMBERS IS F7.2)	
		STOP 77777	

Fortran

- Fortran lives on – now Fortran 2003
- Fortran 2008 under discussion

Thank you John for all your work on the Fortran standard.

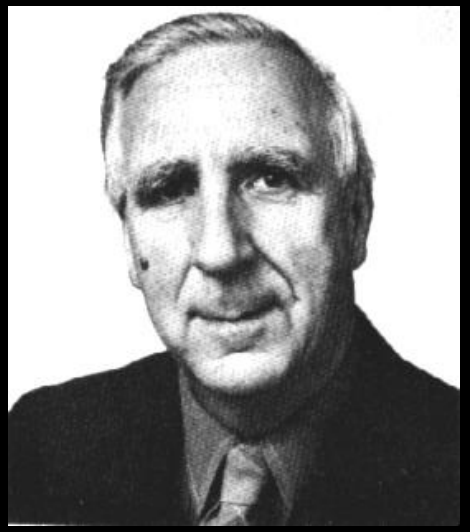
18 years on ISO committee, 10 as convenor

Language Support

- Need more language support of the sort provided in Fortran 2003 and C99
  - Machine (or environment) parameters
  - IEEE arithmetic, such as directed rounding
  - Exception handling
  - Interval arithmetic
- See W Kahan, *How Futile are Mindless Assessments of Roundoff in Floating-Point Computation?* http://www.cs.berkeley.edu/~wkahan/

“I have little doubt that about 80 per cent. of all the results printed from the computer are in error to a much greater extent than the user would believe, ..."

Leslie Fox, IMA Bulletin, 1971



“Giving business people spreadsheets is like giving children circular saws.

The average spreadsheet programmer does little or no advance planning, has no idea whether his or her algorithms are correct … builds in few or no cross-checks, and does little or no testing.

It is a cast iron certainty that the vast majority of spreadsheets contain errors”

Jack Schofield, The Guardian, 2003

NATIONAL PHYSICAL LABORATORY

Notes on Applied Science
No. 32

Rounding Errors in Algebraic Processes

by
J. H. WILKINSON, M.A., Sc.D.

LONDON
HER MAJESTY'S STATIONERY OFFICE
1963

N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, USA, second edition, 2002.

“Since the use of the punched-card equipment required the use of an operator, it encouraged user participation generally, and this was a distinctive feature of Pilot ACE operation

…

Speaking for myself I gained a great deal of experience from user participation, and it was this that led to my own conversion to backward error analysis.”

Wilkinson (1980) in *A History of Computing in the Twentieth Century*, Academic Press

Portability

- For portability, needed a model of floating point arithmetic
- IFIP/WG 2.5. B Ford et al. See: http://www.nsc.liu.se/~boein/ifip/projects/p1.txt
- W S Brown *A simple but realistic model of floating-point computation* ACM Trans. Math. Software,7, 445–480, 1981
- NAG: Chapter X02
- LAPACK: xLAMCH

John was a member of IFIP WG 2.5 for 25 years!

IEEE Arithmetic Formats

Format	Precision	Exponent	Approx Range	Approx precision (u)
Single	24 bits	8 bits	$10^{\pm 38}$	10^{-8}
Double	53 bits	11 bits	$10^{\pm 308}$	10^{-16}
Extended	≥ 64	≥ 15	$10^{\pm 4932}$	10^{-20}

W. Kahan's self-portrait

Handbook for Automatic Computation

Edited by

F. L. Bauer · A. S. Householder · F. W. J. Olver

H. Rutishauser † · K. Samelson · E. Stiefel

Volume II

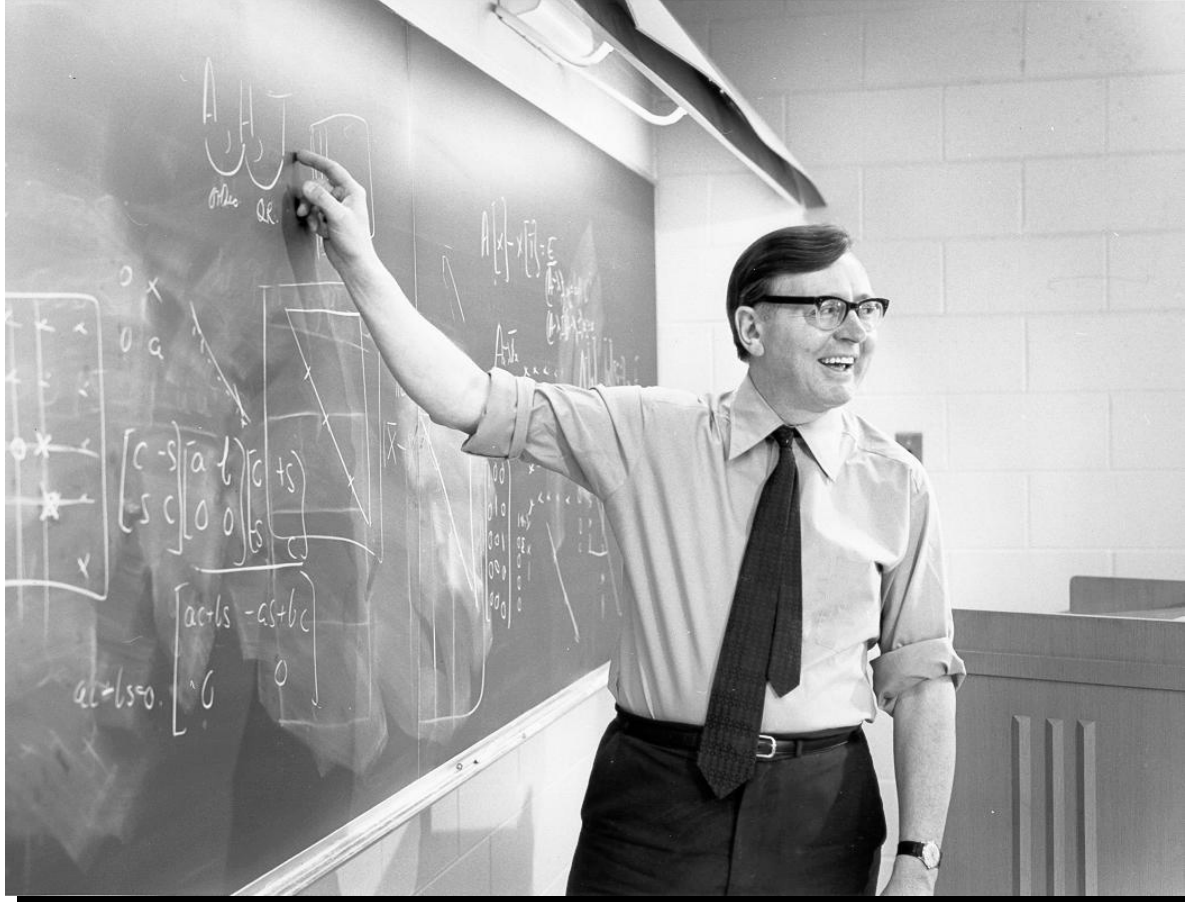
J. H. Wilkinson · C. Reinsch

Linear Algebra

Chief editor

F. L. Bauer

Springer-Verlag Berlin Heidelberg New York 1971



T. J. Dekker, W. Hoffmann; *Algol 60 procedures in numerical algebra part 2;* MC Tracts 23, Mathematisch Centrum, Amsterdam (1968)

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

6

B. T. Smith · J. M. Boyle · J. J. Dongarra
B. S. Garbow · Y. Ikebe · V. C. Klema
C. B. Moler

Matrix Eigensystem Routines –
EISPACK Guide
Second Edition

Springer-Verlag
Berlin · Heidelberg · New York

LINPACK USERS' GUIDE

J.J. Dongarra C.B. Moler
J.R. Bunch G.W. Stewart

siam

Basic Linear Algebra Subprograms

- Level 1 BLAS, 1979
- Level 2 BLAS, 1988
  - Vector machines
- Level 3 BLAS, 1990
  - Hierarchical memory, shared memory parallel
- Dates are for TOMS publication

Blas Festival 2009

English | Gàidhlig

- welcome
- newsletter
- news
- about blas
- gallery
- contact us
- links

mailing list

Subscribe to our mailing list for information on this and future festivals...

tickets

Purchase tickets for all events at thebooth.co.uk or by contacting the venue...

News Feed

Celebrating the Highlands 4th - 12th September 2009

Welcome to the Blas Festival which celebrates the culture of the Scottish Highlands through music and language. Devised by The Highland Council in partnership with Fèisean nan Gàidheal and the Promoters Arts Network, Blas is now in its fourth year.

We are proud to announce that this year's festival which ran from the 5th-13th September 2008 was our most successful yet. In just our fourth year we broke our records for attendances and sell out shows. Congratulations to the promoters, communities, artists and all those who work for Blas in making it such a success.

We have a number of interesting developments in the pipeline which will ensure the Blas Festival remains a great 'homecoming party' for people who love Scottish culture.

This week we will be announcing our successful 2008 figures and we will soon be announcing our programme for 2009!

Check back with us for more news and updates soon!

Latest News

Ceòlraidh is a perfect case for Faculty of Advocates
12 Sep 2008

Leading International Music Artistes Give Blas Exclusive Taste of New Album
07 Sep 2008

Blas Offers Role Model for Festival Delivery
05 Sep 2008

Budget Car Hire

Blas Preferred Car Hire Partner

Homecoming Scotland 2009

homecomingscotland2009.com

John Francis (Mr *QR*)


ARGYLE STREET
blas
Lunch from £4.95
Soup & a Roll £4.95
2 course dinner £11.95
early evening dinner menu from 5 til 7
Dinner for 2 from £20
Tue to Thursday Main course & wine
Sunday Brunch Menu from £5.95
Teas, Coffee & Home Baking
Kids Menu from £2.95
blas
cafe restaurant bar
Teas
Coffees
Home Baking
Lunch &
Dinner
Malt Whisky
Scottish Ales
blas
The taste & flavour of Scotland
www.blasrestaurant.com
Scottish
Tapas from
£2.50 a bowl
Lunch
From £ 3.50
Dinner
£ 8.95 ~ £18.00
blas

Discussion of linear equation solvers on the Pilot ACE

"An interesting feature of the codes is that they made a very intensive use of subroutines; the addition of two vectors, multiplication of a vector by a scalar, inner products, etc., were all coded this way."

Wilkinson, 1980

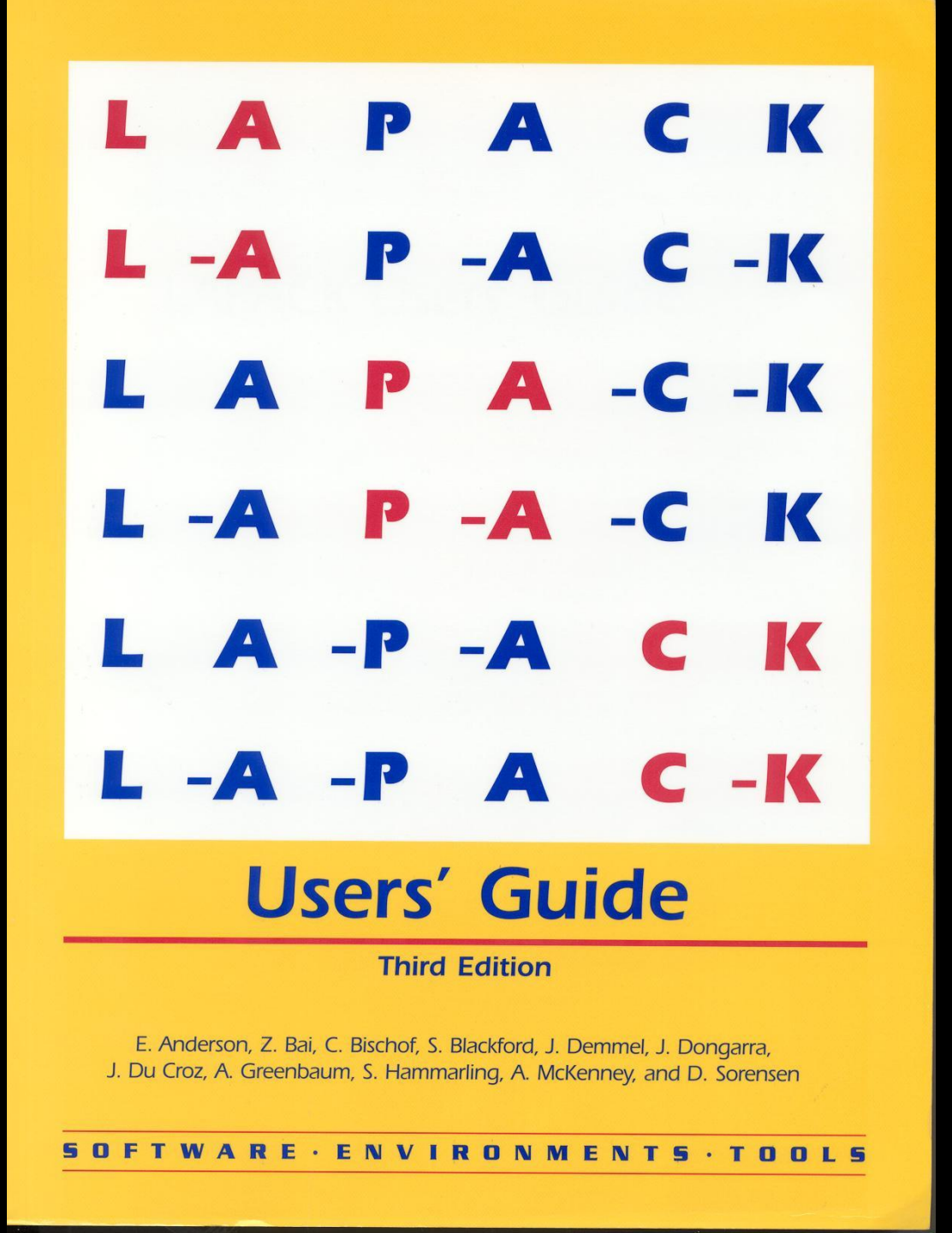
The History of Computing in the 20th Century.

Efficient Use of Data

"Since all machines have stores of finite size often divided up into high speed and auxiliary sections, storage considerations often have a vitally important part to play."

Wilkinson, MTAC, 1955

Emphasis on error and condition estimates, as well as efficiency




LAPACK

LAPACK95 Users' Guide

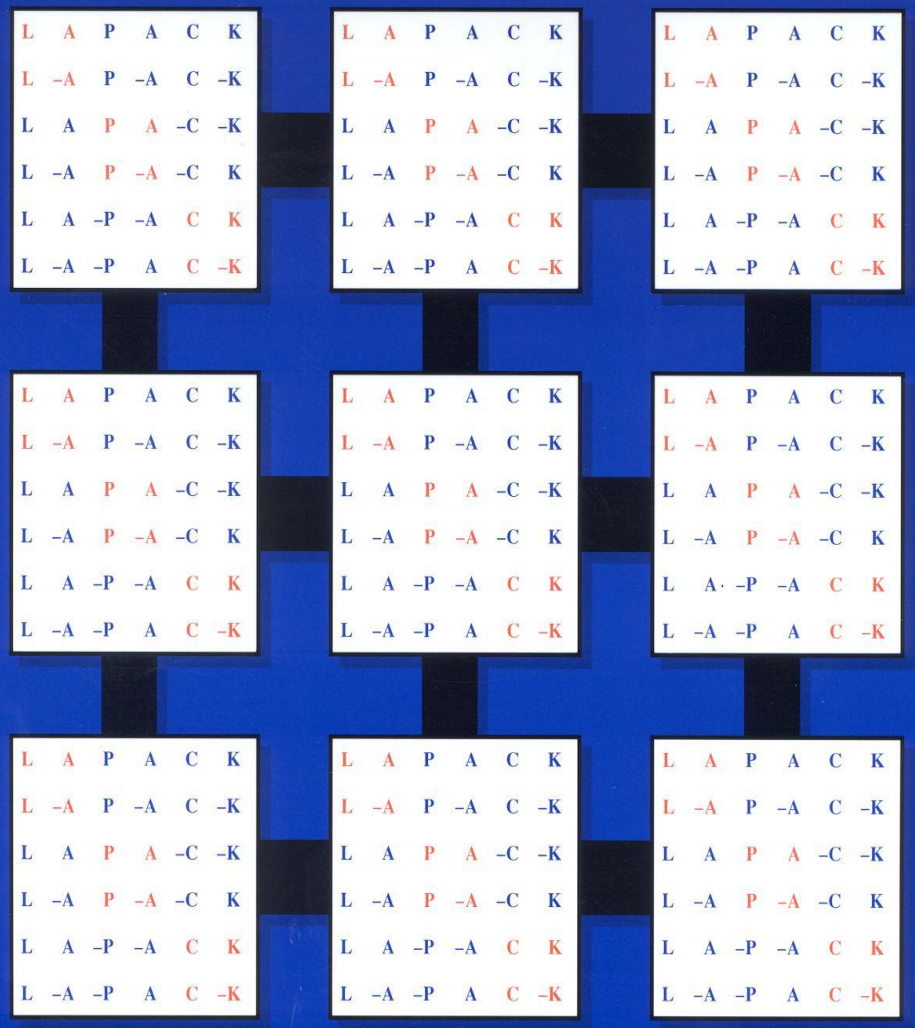
V. A. Barker, L. S. Blackford, J. Dongarra,
J. Du Croz, S. Hammarling, M. Marinova,
J. Waśniewski, and P. Yalamov

siam

SOFTWARE · ENVIRONMENTS · TOOLS

ScaLAPACK Users' Guide

L. S. Blackford · J. Choi · A. Cleary · E. D'Azevedo
J. Demmel · I. Dhillon · J. Dongarra · S. Hammarling
G. Henry · A. Petitet · K. Stanley · D. Walker · R. C. Whaley



Software Code Sizes

- NAG Fortran Library, Mark 21
  - Source: **28.3 Mb**
  - Stringent test programs: **42.1 Mb**
  - Example programs: **4.4 Mb**
  - XML documentation: **136 Mb**
- LAPACK 3.0
  - Source: **12.1 Mb**
  - Testing: **10.9 Mb**
  - Timing: **6.5 Mb**
  - Users' Guide: **407 pages**

Measures of Solution Quality

DGESVX is an 'expert' driver for solving $AX = B$

`DGESVX(...,RCOND,FERR,BERR,WORK,...,INFO)`

`RCOND` : Estimate of $1/\kappa(A)$

`FERR(j)`: Estimated forward error for X_j

`BERR(j)`: Componentwise relative backward error for X_j

(smallest relative change in any element of A and B

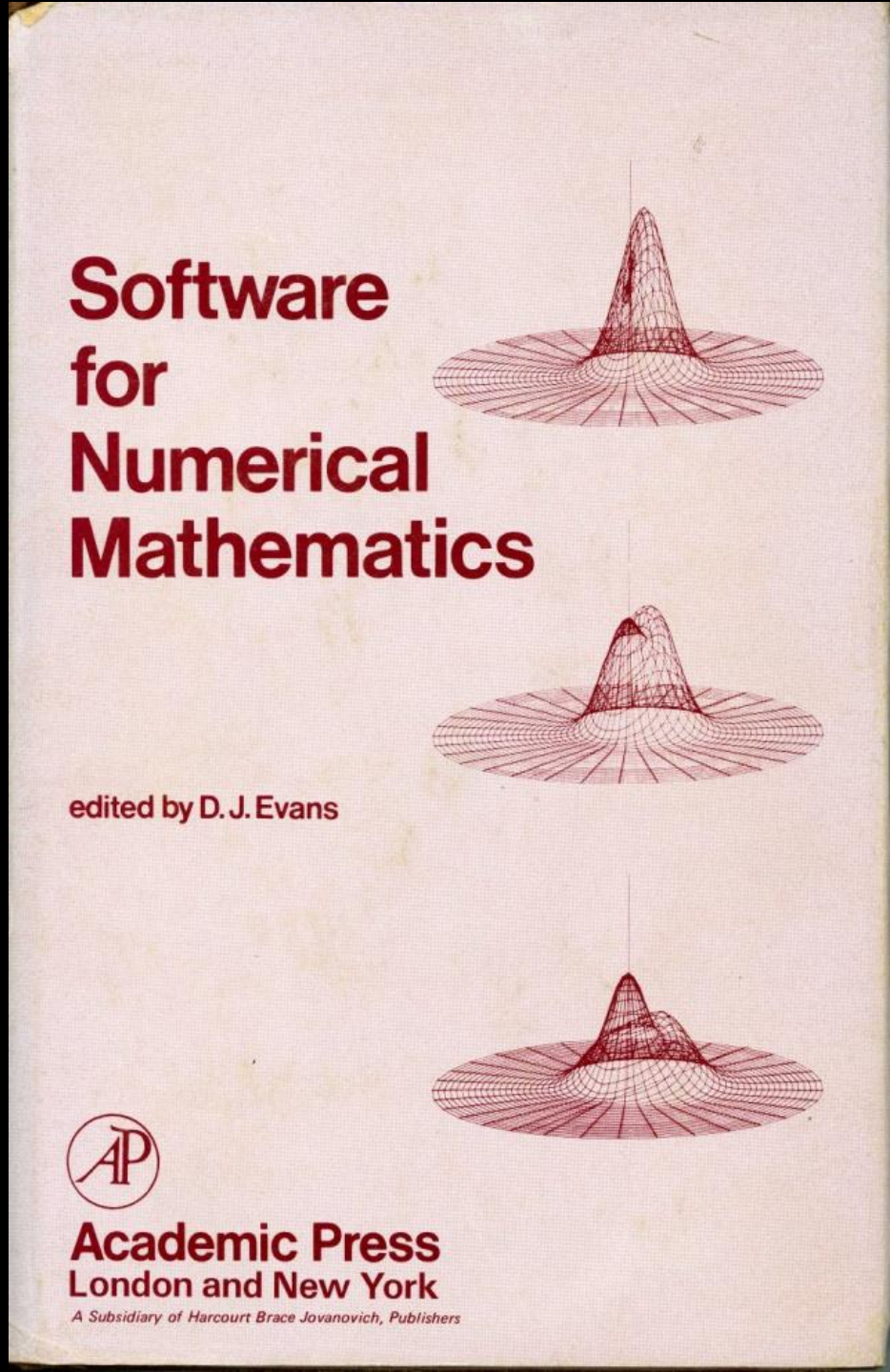
that makes X_j an exact solution)

`WORK(1)`: Reciprocal of pivot growth factor $(1/g)$

`INFO` : > 0 if A is singular or nearly singular

Proceedings of IMA Conference, Loughborough, 1973

Published 1974



Contents

CONTRIBUTORS v

PREFACE vii

1. THEORY OF OPTIMAL ALGORITHMS. By J. F. Traub 1

DISCUSSION: OPTIMAL ALGORITHMS 15

2. LINEAR ALGEBRA ALGORITHMS. By J. H. Wilkinson 17

DISCUSSION: LINEAR ALGEBRA 27

3. DIRECT METHODS FOR SPARSE MATRICES. By J. K. Reid 29

4. ITERATIVE SPARSE MATRIX ALGORITHMS. By D. J. Evans 49

5. SPARSE INVERSE IN THE FACTORED FORM AND MAINTAINING SPARSITY DURING SIMPLEX ITERATIONS. By G. Mitra 85

DISCUSSION: SPARSE MATRICES. Panel. J. K. Reid, D. J. Evans, G. Mitra 99

6. NUMERICAL QUADRATURE: A SURVEY OF THE AVAILABLE ALGORITHMS. By Valerie A. Dixon 105

7. ALGORITHMS FOR INTEGRAL EQUATIONS. By G. F. Miller 139

8. TESTING AND EVALUATION OF SOME SUBROUTINES FOR NUMERICAL QUADRATURE. By Bo Einarsson 149

9. SOME EXTRAPOLATION METHODS FOR THE NUMERICAL CALCULATION OF MULTIDIMENSIONAL INTEGRALS. By A. C. Genz .. 159

DISCUSSION: NUMERICAL QUADRATURE. Panel. Valerie A. Dixon, G. F. Miller, B. Einarsson, A. C. Genz 173

10. INITIAL AND BOUNDARY VALUE ROUTINES FOR ORDINARY DIFFERENTIAL EQUATIONS. By Joan Walsh 177

DISCUSSION: ORDINARY DIFFERENTIAL EQUATIONS 191

11. NONLINEAR OPTIMISATION: A SURVEY OF THE STATE OF THE ART. By L. C. W. Dixon. 193

DISCUSSION: NON-LINEAR OPTIMISATION 217

12. ALGORITHMS FOR CURVE AND SURFACE FITTING. By J. G. Hayes .. 219

13. A DATA FITTING PACKAGE FOR THE NON-SPECIALIST USER. By M. G. Cox 235

14. PIECEWISE QUADRATIC SURFACE FITTING FOR CONTOUR PLOTTING. By M. J. D. Powell 253

DISCUSSION: CURVE AND SURFACE FITTING. Panel. J. G. Hayes, M. G. Cox, M. J. D. Powell. 273

15. RATIONAL APPROXIMATIONS FOR SPECIAL FUNCTIONS. By C. W. Clenshaw 275

16. SPECIAL FUNCTIONS IN THE NAG LIBRARY. By J. L. Schonfelder .. 285

DISCUSSION: SPECIAL FUNCTIONS. Panel. C. W. Clenshaw, J. L. Schonfelder. 301

17. APPLICATION OF ON-LINE TECHNIQUES TO THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS. By A. Sykes 303

18. POTENT–A PACKAGE FOR THE NUMERICAL SOLUTION OF POTENTIAL PROBLEMS IN GENERAL TWO-DIMENSIONAL REGIONS. By C. Ll. Thomas 315

19. A NEW SCHEME FOR INTERACTIVE NUMERICAL COMPUTATION. By D. Hutchinson and P. Jesty 337

DISCUSSION: ON-LINE TECHNIQUES. Panel. A. Sykes, C. Ll. Thomas D. Hutchinson, P. Jesty 353

20. THE ORGANISATION OF NUMERICAL ALGORITHMS LIBRARIES. By B. Ford and S. J. Hague 357

21. MANAGEMENT PRACTICES IN THE DEVELOPMENT AND DISTRIBUTION OF MATHEMATICAL SOFTWARE WITH EMPHASIS ON COMPUTATIONAL AIDS IN A MULTI-MACHINE ENVIRONMENT. By D. B. Taylor, B. Ford and S. J. Hague 373

22. THE DEVELOPMENT AND MAINTENANCE OF MULTI-MACHINE SOFTWARE IN THE NAG PROJECT. By J. A. Prentice 383

23. THE NATS APPROACH TO QUALITY SOFTWARE. By B. T. Smith, J. M. Boyle and W. J. Cody 393

DISCUSSION: ORGANISATION AND PORTABILITY OF PROGRAM LIBRARIES 407

24. SETTING UP A NUMERICAL ADVISORY SERVICE. By R. Fletcher and M. D. Hebden 413

25. USER DOCUMENTATION FOR A GENERAL NUMERICAL LIBRARY: THE NAG APPROACH. By Shirley A. Lill 423

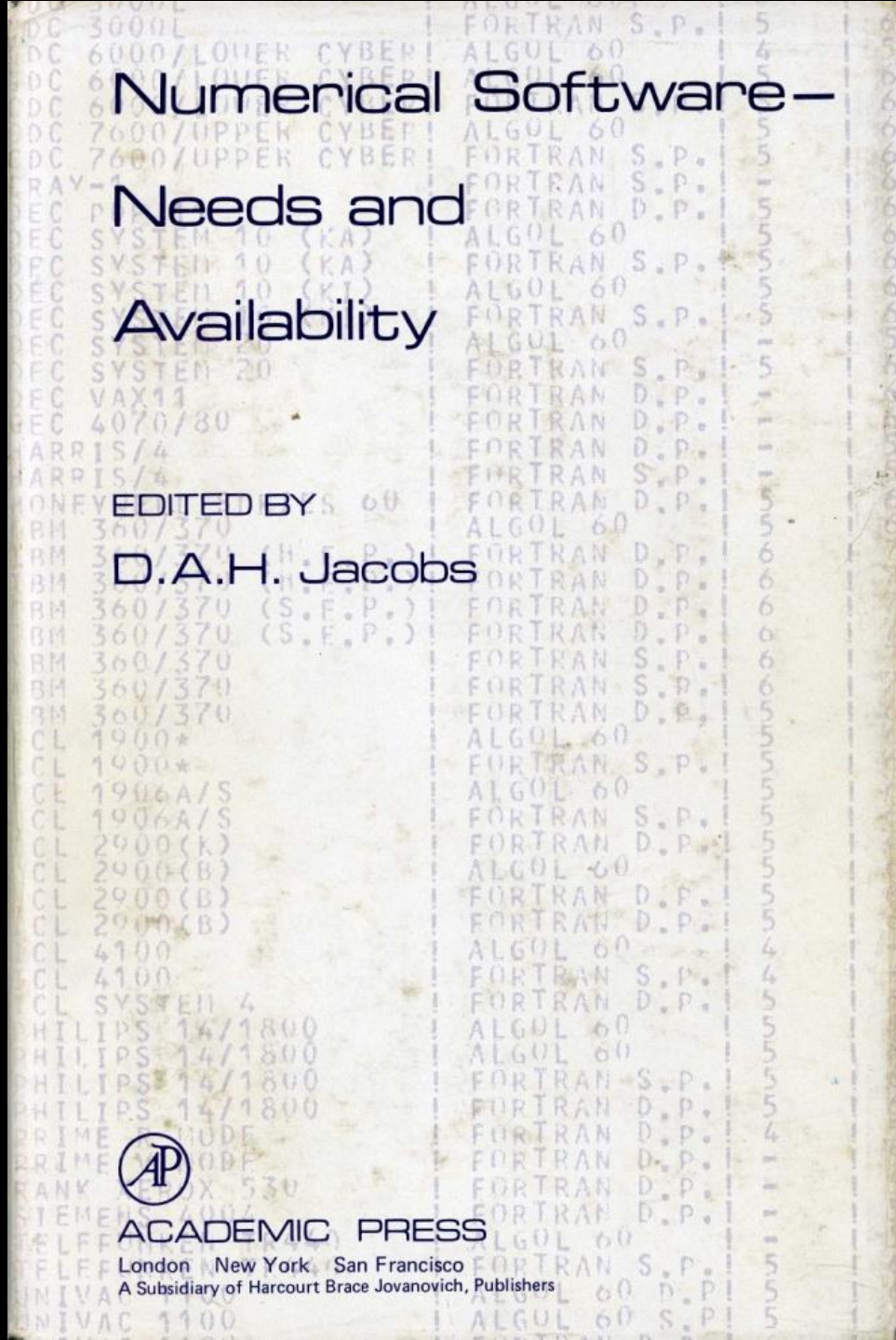
DISCUSSION: DOCUMENTATION. Panel. R. Fletcher, M. D. Hebden, Shirley A. Lill. 433

AUTHOR INDEX 437

SUBJECT INDEX 443

Proceedings of IMA Conference, Sussex 1977

Published 1978



CONTENTS

PART I. LIBRARIES OF NUMERICAL SOFTWARE

I.1 A LIBRARY DESIGN FOR ALL PARTIES by B. Ford and J. Bentley 3
1. Introduction to Numerical Software 3
2. The Concept of a Library 4
3. User Requirements 6
4. Overall Design of Library 7
5. Specific Design Aspects 13
6. Implications of Design Principles for Library Development 17
7. Concluding Remarks 18
8. Acknowledgements 19
9. References 19

I.2 PORTABILITY AND ADAPTABILITY -- WHAT ARE THE ISSUES? by B.T. Smith 21
1. Introduction 22
2. The Problem 22
3. The Source Text 22
4. Algorithms 27
5. The Compromise 29
6. From Adaptable Algorithmic Representations to Transportable Software -- A Practical Example 32
7. References 36

I.3 LIBRARIES: THE USER INTERFACE by P. Kemp 39
1. Introduction 39
2. The Requirements of the End User 40
3. The Requirements of the Computing Centre 41
4. Aspects of the User Interface 42
5. Documentation 43
6. Library Structure 46
7. Advice on Library Use 51
8. Library Contents 52
9. A Look at Some Libraries 53
10. Conclusions 55
11. References 56

I.4 SOFTWARE TOOLS by S.J. Hague 57
1. Introduction 57
2. Manipulation and Mechanisation 59

3. Various Tools Used in Mathematical Software Development 62
4. TAMPR - A Transformational System 67
5. Present State and Future Developments 75
6. Acknowledgements 78
7. References 78

I.5 THE VALUE OF A NUMERICAL LIBRARY TO A LARGE INDUSTRIAL ORGANIZATION by R.W. McIntyre and T. Tate 81
1. Introduction 81
2. Engineering Computing 82
3. Engineering Software 83
4. Numerical Methods in Engineering 85
5. Acquiring a General Purpose Numerical Library 87
6. IMSL as Part of the Company's Computing Service 89
7. Conclusions 93
8. Acknowledgements 94
9. References 94

I.6 USING THE NAG LIBRARY IN INDUSTRIAL RESEARCH by J.P. Whelan 97
1. Introduction 97
2. Changing Computers 98
3. The User Documentation Subset 99
4. A User Advisory Service 101
5. Difficult Problems 102
6. Monitoring Library Usage 104
7. Summary 105

PART II LINEAR AND NON-LINEAR ALGEBRA

II.1 SINGULAR-VALUE DECOMPOSITION - BASIC ASPECTS by J.H. Wilkinson 109
1. Introduction 109
2. Practical Difficulties 112
3. The Singular Value Decomposition 113
4. Advantages of the SVD 118
5. The Golub Algorithm 119
6. Linear Systems and Least Squares 125
7. The Pseudo-Inverse 127
8. Analysis in Terms of the SVD 129
9. Inverse Iteration 131
10. References 135

II.2 SINGULAR VALUE DECOMPOSITION IN MULTIVARIATE ANALYSIS by C.F. Banfield 137
1. Introduction 137

2. Multivariate Techniques Using Singular Value Decomposition 138
3. Techniques on Small Machines Using Singular Value Decomposition 144
4. Statistical Software for Singular Value Decomposition 146
5. Acknowledgement 146
6. References 147

II.3 SOFTWARE FOR SPARSE MATRICES by J.K. Reid 151
1. Introduction 151
2. Direct Solution of Linear Sets of Equations 152
3. Iterative Solution of Linear Sets of Equations 155
4. Least Squares Problems 157
5. Non-Linear Sets of Equations 158
6. Eigenvalue Problems 160
7. The User Interface 161
8. Acknowledgement 162
9. References 163

II.4 NON-LINEAR ALGEBRAIC EQUATIONS IN PROCESS ENGINEERING CALCULATIONS by A.K. Mallin-Jones 167
1. Introduction 167
2. Design of Chemical Plant 167
3. Representation of Flowsheets 169
4. Program Structure 171
5. Effect of Program Structure on Equation Solvers 172
6. Experience of Using "Inside Out" Equation Solvers 173
7. Algorithms 177
8. Alternative Ways of Simulating Flowsheets 178
9. Conclusion 178
10. References 179

PART III DATA-FITTING

III.1 DATA-FITTING ALGORITHMS AVAILABLE, IN PREPARATION, AND IN PROSPECT, FOR THE NAG LIBRARY by J.G. Hayes 183
1. Introduction 183
2. Polynomial Curves 185
3. Cubic Spline Curves 189
4. Spline Surfaces 195
5. Polynomial Surfaces 197
6. ℓ_1 and ℓ_∞ Norms 199
7. References 201

III.2 CURVE FITTING USING THE ROTHAMSTED MAXIMUM LIKELIHOOD PROGRAM by G.J.S. Ross 203
1. Introduction 203

2. Methods for Fitting Non-Linear Curves 214
3. Program Availability 222
4. References 222

PART IV COMPUTER AIDED DESIGN AND SIMULATION

IV.1 INTERACTIVE COMPUTING: A NEW OPPORTUNITY by H.H. Rosenbrock 227
1. Introduction 227
2. Interactive Computing 228
3. An Example 230
4. Software Problems 232
5. Cultural Difficulties 233
6. References 235

IV.2 THE USE OF NUMERICAL SOFTWARE IN THE DIGITAL SIMULATION LANGUAGE PMSP by T. Chambers 237
1. Introduction 237
2. The Use of Digital Simulation 237
3. PMSP - The Plant Modelling System Program 238
4. Properties of PMSP Models 239
5. User Requirements from PMSP 240
6. Description of Numerical Algorithms Used 242
7. Acknowledgement 252
8. References 252

PART V DIFFERENTIAL AND INTEGRAL EQUATIONS

V.1 SOLUTION OF LARGE, STIFF INITIAL VALUE PROBLEMS - THE STATE OF THE ART by A.R. Curtis 257
1. Introduction 257
2. Problem Classification and Sources 258
3. Software Design Considerations 261
4. FACSIMILE - A Problem-Oriented Backward-Difference Program 274
5. Experience with FACSIMILE 275
6. Acknowledgements 277
7. References 277

V.2 SOME FACTORS AFFECTING THE EFFICIENCY OF STIFF INTEGRATION ROUTINES by H.H. Robertson 279
1. Introduction 279
2. Numerical Stability 280
3. Prediction and Error Estimates 283
4. Structure 285
5. Local Convergence of the Parallel Chord Iteration 290
6. Applications 292
7. Numerical Comparison of Partitioned Updating 295

8. Concluding Remarks 298
9. References 299

V.3 NUMERICAL SOFTWARE FOR INTEGRAL EQUATIONS by L.M. Delves 303
1. Introduction 303
2. Volterra Equations of the Second Kind 304
3. Fredholm Equations of the Second Kind 306
4. Equations of the First Kind 308
5. Eigenvalue Equations 310
6. Comparing Integral Equation Routines 311
7. Dealing with Practical Problems 318
8. References 321

V.4 PROBLEMS OF ALGORITHM DESIGN FOR PARTIAL DIFFERENTIAL EQUATIONS by J. Walsh 325
1. Introduction 325
2. Parabolic Equations 330
3. Hyperbolic Equations and Characteristics 332
4. The Problem of Singularities 334
5. Design of Packages 336
6. References 337

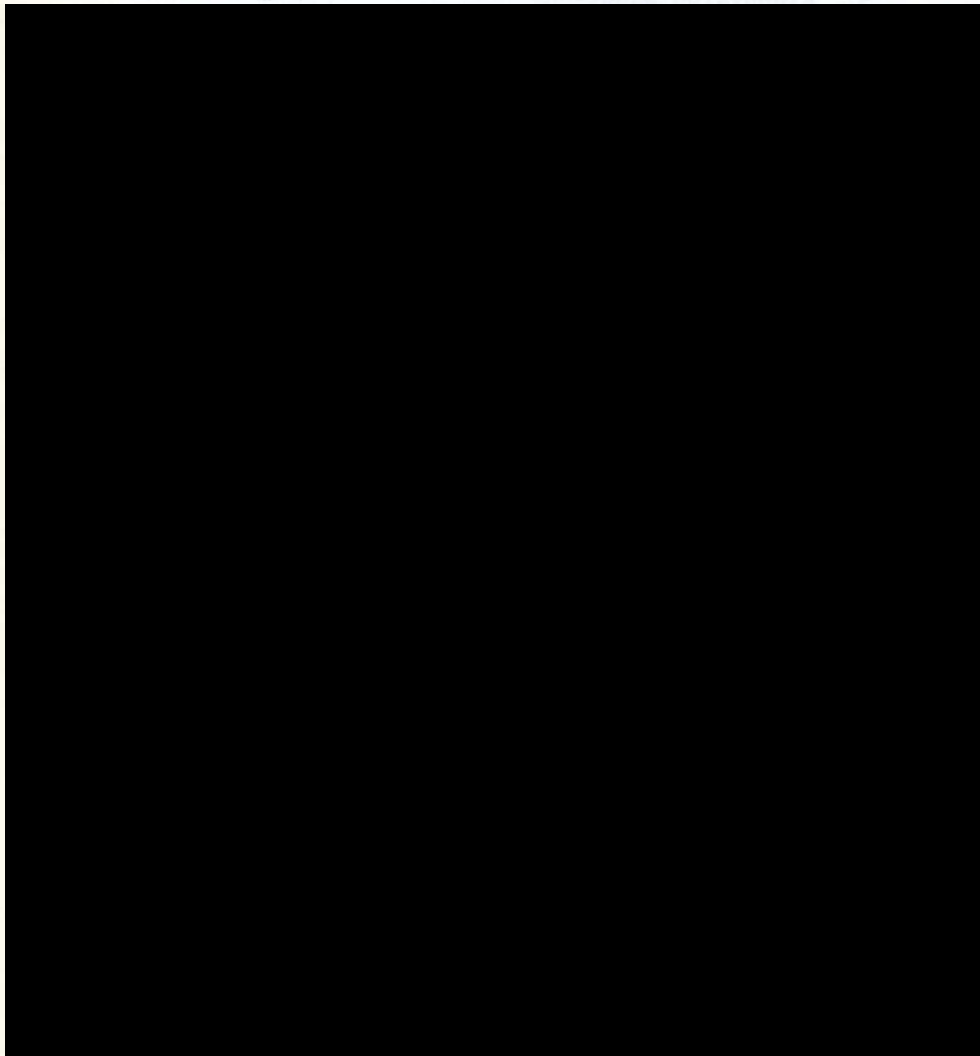
V.5 SOME ELLIPTIC BOUNDARY-VALUE PROBLEMS IN THE CEGB by D.A.H. Jacobs 339
1. Introduction 339
2. The BERSAFE Suite 341
3. Iterative Black-Box Routines 347
4. Numerical Software - Outstanding Needs 353
5. Conclusions 357
6. Acknowledgements 357
7. References 358

PART VI OPTIMIZATION

VI.1 MATHEMATICAL PROGRAMMING SYSTEMS by E.M.L. Beale 363
1. Introduction 363
2. The Structure of a Mathematical Programming System 364
3. Algorithmic Aspects 368
4. Matrix Generators and Report Writers 371
5. References 374

VI.2 OPTIMIZATION BY NON-LINEAR SCALING by W.C. Davidon 377
1. Introduction 377
2. An Algorithm Schema 378
3. Properties of the Algorithm 380
4. Concluding Remarks 382
5. References 382

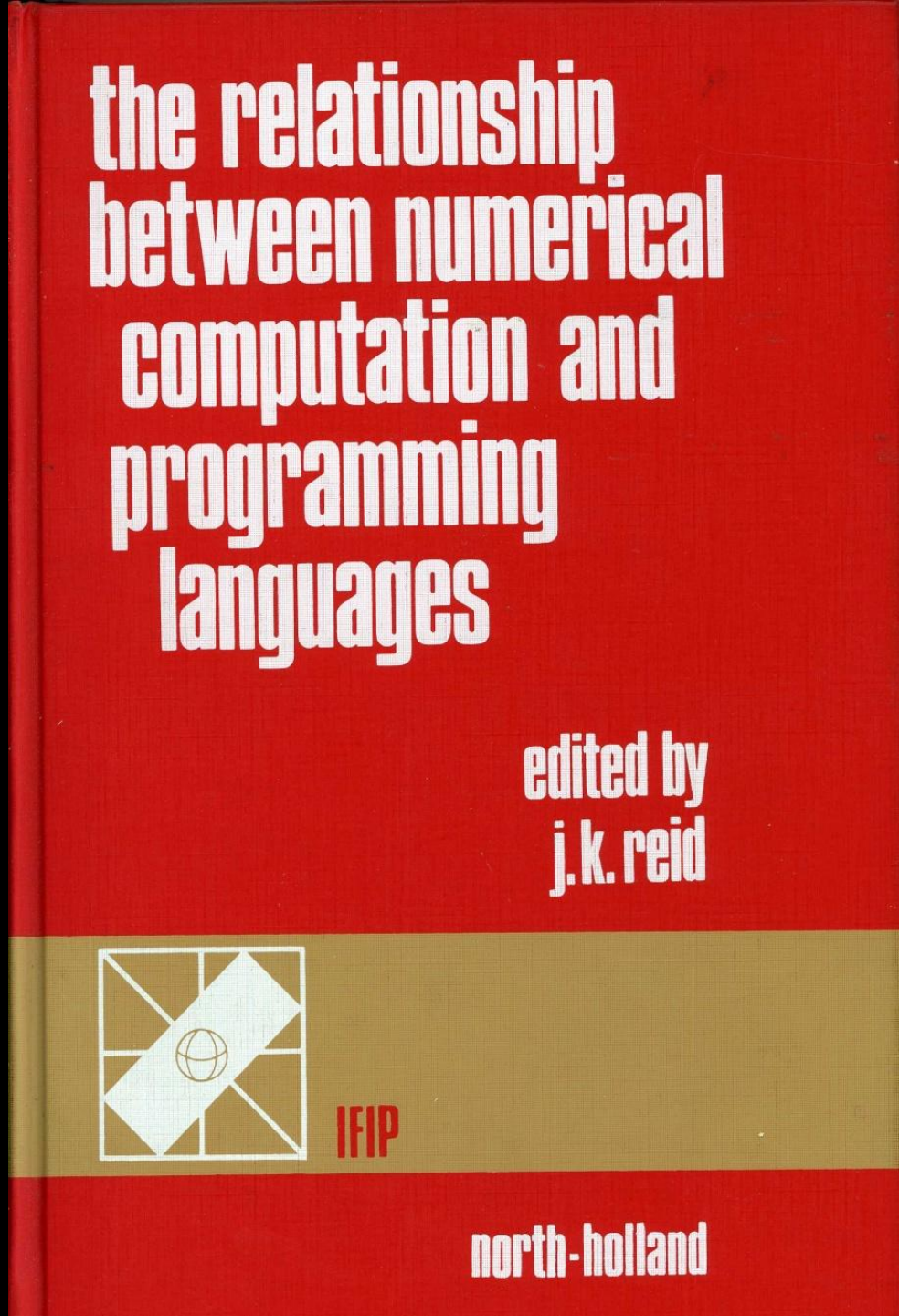
VI. 3 OPTIMIZATION IN PRACTICE by S.E. Hersom 385
1. Introduction 385
2. The Routines 385
3. Applications 392
4. Future 399
5. Conclusion 399
6. References 400



Proceedings of IFIP Conference, Boulder 1981

Published 1982

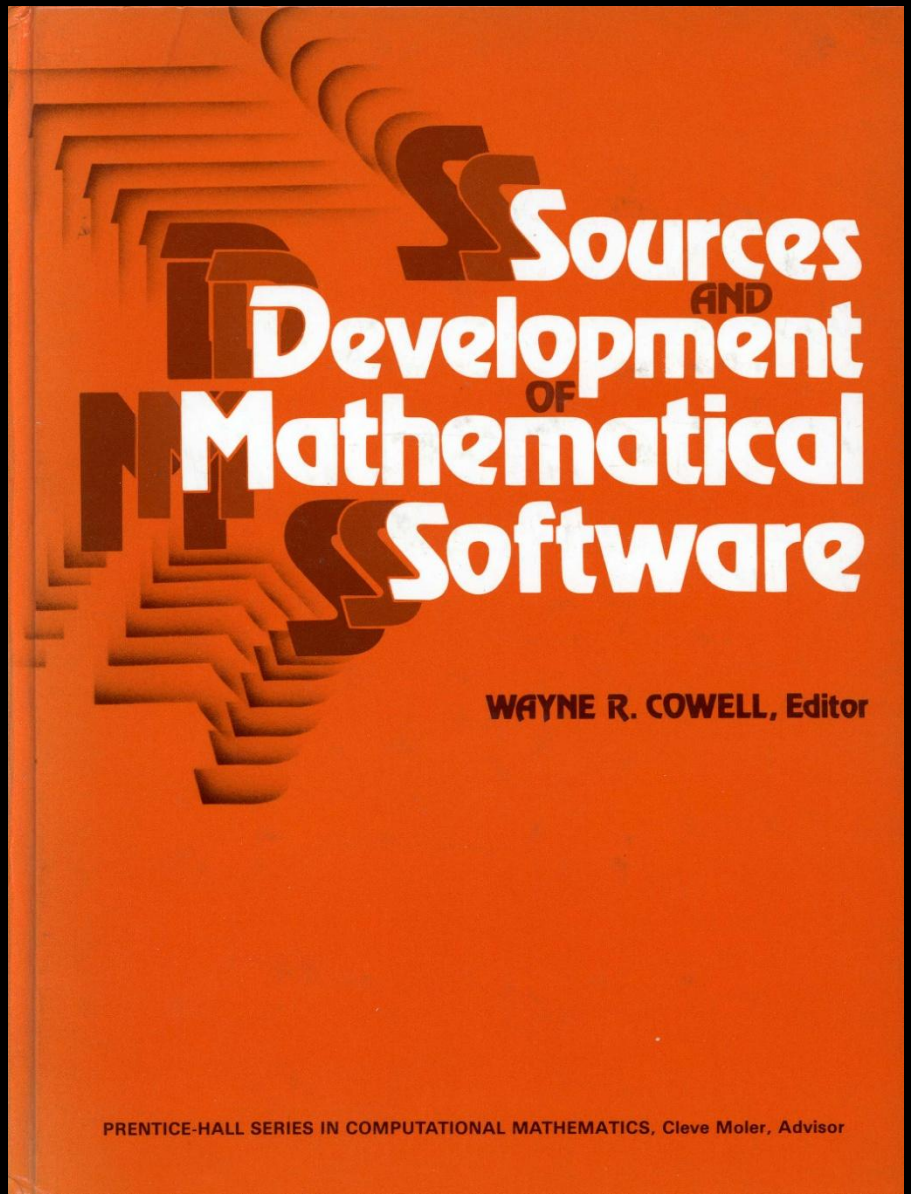
Included “Floating-point Parameters, Models and Standards by Jim Cody



Jim Cody in 1969

Sadly passed away
on June 24th, 2009

State of Libraries and Packages in 1984



Articles on:

- Observations on Mathematical Software Effort (Cody)
- LINPACK (Dongarra, Stewart)
- FUNPACK (Cody)
- EISPACK (Dongarra, Moler)
- MINPACK (Moré, Sorensen, Garbow, Hillstrom)
- Software for ODEs (Shampine, Watts)

... (cont'd)

- Sources of Information on Quadrature Software (Kahaner)
- A Survey of Sparse Matrix Software (Duff)
- Mathematical Software for Elliptic Boundary Value Problems (Boisvert, Sweet)
- The IMSL Library (Aird)
- The SLATEC Common Mathematical Library (Buzbee)

... (cont'd)

- The Boeing Mathematical Software Library (Erisman, Neves, Philips)
- The PORT Mathematical Subroutine Library (Fox)
- The Evolving NAG Library Service (Ford, Pool)
- ----------
- HSL

Today

- BCSLIB, IMSL, HSL and NAG are all alive
- A number of the PACKs are still available; LAPACK is probably the most active (but needs more funding)

- PETSc
- Trilinos (maintained repository), ...

Big Challenge for the Future

- Multicore and hybrid chips
- Dongarra: “There is no Moore’s law for software”

- See the final talk by Jack Dongarra