CONFERENTIE VAN NUMERIEK WISKUNDIGEN

26, 27 en 28 september 1983

CONFERENTIEOORD WOUDSCHOTEN
ZEIST

Werkgemeenschap Numerieke Wiskunde

CONFERENTIE VAN NUMERIEK WISKUNDIGEN

26, 27 en 28 september 1983

CONFERENTIEOORD WOUDSCHOTEN
ZEIST

Werkgemeenschap Numerieke Wiskunde

Uitgave verzorgd door
CENTRUM VOOR WISKUNDE EN INFORMATICA TE AMSTERDAM

ACHTSTE CONFERENTIE NUMERIEKE WISKUNDE

DOEL VAN DE CONFERENTIE

De Conferentie Numerieke Wiskunde wordt eenmaal per jaar gehouden onder auspiciën van de Werkgemeenschap Numerieke Wiskunde. Het doel van de conferentie is om kennis te nemen van recente ontwikkelingen binnen de numerieke wiskunde. Hiertoe worden jaarlijks twee thema's vastgesteld. Lezingen over deze thema's worden normaliter verzorgd door uitgenodigde buitenlandse deskundigen.

THEMA

De thema's zijn:

- Defectcorrecties en a posteriori foutschattingen.
- De rol van software, interval-aritmetiek en supercomputers in de numerieke analyse.

ORGANISATIE

De organisatie is in handen van de voorbereidingscommissie bestaande uit de heren Van der Houwen (CWI), Axelsson (KUN), van Veldhuizen (VU) en Verwer (CWI), en van het Centrum voor Wiskunde en Informatica.

UITGENODIGDE SPREKERS

K. BÖHMER, Universität Marburg.
I.S. DUFF, AERE Harwell.
T. DUPONT, University of Chicago.
M.J. KASCIC, Control Data, Minnesota.
H.J. STETTER, Technische Universität Wien.

Op uitnodiging van de voorbereidingscommissie en ter aanvulling van het programma hebben P.W. HEMKER en H. VAN DER VORST zich bereid verklaard een lezing te verzorgen over hun ervaringen met supercomputers.

PROGRAMMA

Maandag 26 september 1983

10.00-11.15	aankomst en koffie	15.15-15.45	thee
11.15-12.30	opening, Stetter	15.45-16.45	Dupont
12.45	lunch	18.00	diner
14.15-15.15	Kascic		

Dinsdag 27 september 1983

8.00	ontbijt	12.45	lunch
9.00-10.00	Duff	14.15-15.15	Hemker
10.00-10.30	koffie	15.15-15.45	thee
10.30-11.30	Böhmer	15.45-16.45	Stetter
11.30-12.30	Dupont	18.00	diner

Woensdag 28 september 1983

8.00	ontbijt	11.30-12.30	van der Vorst
9.00-10.00	Duff	12.45	lunch
10.00-10.30	koffie	13.45-14.45	Kascic
10.30-11.30	Böhmer	14.45	sluiting
			thee,vertrek

De bar is geopend van 17.00-18.00 uur en van 20.30-24.00 uur.

TITELS EN SAMENVATTINGEN VOORDRACHTEN *)

Maandag 26 september 1983

11.15	opening	
	H.J. Stetter:	"Error-free solution of algebraic problems in floating-point arithmetic: Defect correction and interval computation in the assesment of computational errors"
14.15	M.J. Kascic:	"An introduction to vector processing with application to numerical methods"
15.45	T. Dupont:	"A posteriori error estimation for evolution equations with time-dependent meshes"

Dinsdag 27 september 1983

9.00	I.S. Duff:	"Basic aspects of numerical software"
10.30	K. Böhmer:	"Defect correction methods: general principle, software-considerations, application to Hartree-Fock methods"
11.30	T. Dupont:	"A posteriori error estimation for evolution equations with time-dependent meshes"
14.15	P.W. Hemker:	"A portable vector code for autonomous multigrid modules"
15.45	H.J. Stetter:	"Error-free solution of algebraic problems in floating-point arithmetic: High-accuracy algorithms with guaranteed results"

Woensdag 28 september 1983

9.00	I.S. Duff:	"Organization of numerical software libraries"
10.30	K. Böhmer:	"Defect correction methods: general principle, software-considerations, application to Hartree-Fock methods"
11.30	H. van der Vorst:	"Comparative performance tests on the CRAY-1 and CYBER-205"
13.45	M.J. Kascic:	"Vorton dynamics: a case study of developing a fluid dynamics model for a vector processor"

*) De samenvatting van T. Dupont zal zo mogelijk bij de aanvang van de conferentie worden rondgedeeld.

DEFECT CORRECTION METHODS: GENERAL PRINCIPLE, SOFTWARE-CONSIDERATIONS, APPLICATION TO HARTREE-FOCK-METHODS

K. Böhmer, Universität Marburg.

The defect of a given approximation with respect to a given equation may be used either to estimate the error of this approximation or to correct it into higher accuracy. Especially the second approach is used in many numerical methods, e.g. iterative improvement in systems of linear equations, in multigrid-methods, in boundary and initial value problems software for ordinary and partial differential equations.

After comparing some different approaches we concentrate to the case, where asymptotic expansions are available and properly describe the situation. Discrete Newton methods are then the easiest way to determine very general discretization methods of variable order and stepsize. The realization of these strategies in modern software e.g. PASVA, is described. Applications in the energy level computation of many electron atoms via Hartree-Fock methods are discussed.

References

[1] Fox, L., Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations, Proc. Royal Soc. London A 190, 31-59 (1947).

[2] Pereyra, V., Accelerating the convergence of discretization algorithms, SIAM J. Numer.Anal. 4, 508-533 (1967).

[3] _____, Iterated deferred corrections for operator equations, Numer. Math. 10, 316-323 (1967).

[4] _____, Iterated deferred corrections for nonlinear BVP's, Numer. Math. 11, 111-125 (1968).

[5] _____, PASVA 3: An adaptive finite difference FORTRAN program for first order NLOB's, in "Codes for BVP's in ODE's", Lecture Notes in Computer Science 76, Springer Verlag, 67-88 (1979).

[6] Böhmer, K., Discrete Newton Methods and iterated defect corrections, Numer. Math. 37, 167-192 (1981).

[7] _____, Asymptotic expansion for the discretization error in linear elliptic boundary value problems on general regions, Math. Z. 177, 235-255 (1981).

[8] Brakhage, H., Über die numerische Behandlung von Integralgleichungen nach der Quadraturformelmethode, Numer. Math. 2, 183-196 (1960).

[9] Frank, R., The method of iterated defect correction and its application to two point boundary value problems I, II, Numer.Math. 25, 409-419 (1976), 27, 407-420 (1977).

[10] Frank, R. & Ueberhuber, C.W., Iterated defect corrections for differential equation, Part I: Theoretical results, Computing 20, 207-228 (1978).

[11] Lindberg, B., Error estimation and iterative improvement for discretization algorithms, BIT 20, 486-500 (1980).

[12] Stetter, H.J., The defect correction principle and discretization methods, Numer. Math. 29, 425-443 (1978).

[13] Skeel, R.D., A theoretical framework for proving accuracy results for deferred connections, SIAM J. Numer. Anal. 19, 171-196 (1982).

[14] Zadunaisky, P.E., On the estimation of errors propagated in the numerical integration of ordinary differential equations, Numer. Math. 27, 21-40 (1976).

[15] Hackbusch, W., Bemerkungen zum iterierten Defektkorrektur und zu ihrer Kombination mit Mehrgitterverfahren, Report 79-13, Mathem. Institut, Universität Köln (1979).

[16] Brandt, A., Multilevel adaptive solutions to boundary value problems, Math. Comp. 31, 333-390 (1977).

[17] Hemker, P.W. & Schippers H., Multiple grid methods for the solution of Fredholm integral equations of the second kind, Math. Comp. 36, 215-232 (1981).

BASIC ASPECTS OF NUMERICAL SOFTWARE

I.S. Duff, AERE Harwell.

As labour costs increase and material costs decrease, the balance of expenditure in scientific computation continues to shift from hardware to software. Indeed the cost of computing equipment measured against performance has shown a rapid decline in recent years. Unfortunately, the labour-intensive software market both numerical and non-numerical has not shown a similar decrease. Some years ago, a careful study of numerical software development in the United States calculated that the development costs were over $20 per card and it is our belief that they are much greater today.

The purpose of this preamble is to emphasise the importance of good numerical software practices to the scientific computing community. If some well-considered procedures are adopted from the outset in the development of numerical software, the savings, both immediate and long term, can be enormous. It is the purpose of this first lecture to establish guidelines for developing software and to indicate what procedures should be adopted. Although our remarks will be primarily geared to numerical software and will be addressed to the numerical analysis community, many are relevant to non-mathematical software and to users as well as developers of the software.

We begin by identifying some of the basic tenets of numerical software. Included among these will be robustness, reliability, efficiency, portability, flexibility, modularity, clarity and usability. We discuss these tenets in detail and emphasize their differences and relative importance. In this discussion and in all subsequent ones we will relate our remarks to actual examples of mathematical software.

The second part of the first lecture will discuss the procedures that should be adopted when writing numerical software. We will concentrate on aspects of design, language, coding, packaging, documentation and distribution. We will touch on the incorporation of such software in a numerical software library although libraries will be the primary focus of our second lecture.

Some basic references:

Aird, T.J., Battiste, E.L. and Gregory, W.C. (1977). Portability of mathematical software coded in Fortran. TOMS 3, pp.113-127.

Evans, D.J. (Ed.) (1974). Software for numerical mathematics. Academic Press.

Fosdick, L.D. (Ed.)(1979). Peformance Evaluation of Numerical Software. North-Holland.

Cowell, W.R. (Ed.)(1976). Portability of Numerical Software. Lecture Notes in Computer Science 57. Springer Verlag.

Cowell, W.R. (Ed.)(1983). Sources and development of mathematical software. Prentice Hall. To appear.

Delves, L.M. (Ed.)(1980). Production and assessment of numerical software. Academic Press.

Jacobs, D.A.H. (1978). Numerical Software - Needs and Availability. Academic Press.

Kernighan, B.W. and Plauger, P.J. (1978). The elements of programming style. Second edition. McGraw-Hill.

Rice, J.R. (1971). Mathematical Software. Academic Press.

Rice, J.R. (1975). Papers from Mathematical Software II, May 29-31, 1974: Part 1. TOMS 1, pp.7-12.

Rice, J.R. (1977). Mathematical Software III. Academic Press.

Ryder, B.G. (1974). The PFORT verifier. Software Practice and Experience, 4, pp.359-377.

ORGANIZATION OF NUMERICAL SOFTWARE LIBRARIES

I.S. Duff, AERE Harwell.

We begin the second lecture with a kind of consumer's guide to existing numerical software libraries. We discuss the major commercial libraries (for example NAG, IMSL and PORT), smaller libraries which are at least partly driven by the research interests of local numerical analysts (for example, HSL, BCS and NUMAL), cooperative efforts (for example, SLATEC), manufacturers libraries (for example, CRAY SCILIB, IBM SLMATH), public domain software (for example, EISPACK, LINPACK) and published collections of software (for example, CALGO of the ACM).

We illustrate the diversity of sources of software by considering the area of sparse matrices. Here there is much active and good research but relatively little good software which is spread thinly over many software developers and research groups.

It could be very easy to be swamped with an unmanageable amount of software. We discuss how one might avoid this and indicate how to select appropriate software and organize the resulting multi-library.

Whatever choice one makes about the basis of the numerical library support, in any active research institution there will be many in-house contributions to the available software. We consider how to discipline and incorporate these contributions. In particular, we discuss the documentation, distribution and accessibility of locally developed libraries drawing heavily on our experience with the Harwell Subroutine Library. An important related area which we also consider is that of general numerical support and advice to the user community. In connection with this we discuss the recent use of on-line databases and/or decision trees to aid both the user and the numerical consultant.

Even on a single site, it is becoming increasingly common to have a multi-machine environment. For example, at Harwell an IBM 3081K, CRAY 1-S and a VAX 11/780 are all used extensively by computational scientists and there are many other smaller machines used by more specialized groups. Many people will wish to use the same subroutines on different machines with no change to their calling program. We discuss such issues of internal portability as they affect numerical software libraries. We illustrate how we are trying to overcome this problem at Harwell by using a masterfile system. We discuss the design and implementation of our masterfile system and report on our experience with it.

Some basic references

Barinka, L.L. (1975). Some experience with constructing, testing, and certifying a standard mathematical subroutine library. TOMS 1, pp.165-177.

Chan, T.F., Coughran, W.M..Jr., Grosse, E. and Heath, M.T. (1980). A numerical library and its support. TOMS 6, pp.135-145.

Cody, W.J. (1974). The construction of numerical subroutine libraries. SIAM Rev. 16, pp.36-46.

Duff. I.S. (1979). Some notes on numerical library construction. AERE Report M.3057.

Fletcher, R. and Hebden, M.D. (1974). Setting up a numerical advisory service, pp.413-421. In Software for Numerical Mathematics. D.J. Evans (Ed.). Academic Press.

Ford, B. and Hague, S.J. (1974). The organisation of numerical algorithms libraries, pp.357-372. In Software for Numerical Mathematics. D.J. Evans (Ed.). Academic Press.

Ford, B. and Sayers, D.K. (1976). Developing a single numerical algorithms library for different machine ranges. TOMS 2, pp.115-131.

Gaffney, P.W. (1981). Information and advice on numerical software. Report ORNL/CSD/TM-147, Computer Sciences Division, Oak Ridge National Laboratory.

Gill, P.E., Murray, W., Picken, S.M. and Wright, M.H. (1979). The design and structure of a Fortran Program Library for Optimization. TOMS 5, pp.259-283.

Hopper, M.J. (1976). TSSD, a typesetting system for scientific documents. Report AERE-R.8574. AERE Harwell. HMSO. London.

Hopper, M.J. (Compiler)(1981). Harwell Subroutine Library. A catalogue of subroutines (1981). Report AERE-R.9185. AERE Harwell. HMSO. London.

Kemp, P. (1978). Libraries: The user interface, pp.39-56. In Numerical Software - Needs and Availability. D.A.H. Jacobs (Ed.). Academic Press.

A SHORT BIBLIOGRAPHY OF TEXTS ON NUMERICAL SOFTWARE AND NUMERICAL SOFTWARE LIBRARIES (compiled in December 1979)

I.S. Duff, AERE Harwell

Reports of Conferences

Evans, D.J. (Ed.)(1974). Software for numerical mathematics. Academic Press.

Fosdick, L.D.(Ed.)(1979). Performance evaluation of numerical software. North-Holland.

Cowell, W. (Ed.)(1976). Portability of Numerical Software. Lecture Notes in Computer Science 57. Springer-Verlag.

Delves, L.M. (Ed.)(1980). Production and assessment of numerical software. Academic Press.

Jacobs, D.A.H. (1978). Numerical Software - Needs and Availability. Academic Press.

Rice, J.R. (1975). Papers from Mathematical Software II, May 29-31, 1974: Part 1. TOMS 1, pp.7-12.

Rice, J.R. (1971). Mathematical software. Academic Press.

Rice, J.R. (1977). Mathematical software III. Academic Press.

Boyle, J.M., Cody, W.J., Cowell, W.R., Garbow, B.S., Ikebe, Y., Moler, C.B. and Smith B.T. (1972). NATS, A collaborative effort to certify and disseminate mathematical software, pp.630-635. Proc. 1972 National ACM Conference, Vol. II. Assoc. for Computing Machinery, New York.

Cody, W.J. (1971). Software for the elementary functions, pp.171-186. In Mathematical Software. J.R. Rice (Ed.). Academic Press.

Software. Collections, Efforts, General

Cody, W.J. (1975). The FUNPACK package of special function subroutines. TOMS 1, pp.13-25.

Cowell, W.R. and Fosdick, L.D. (1977). Mathematical software production, pp.195-224. In Mathematical Software III. J.R. Rice (Ed.). Academic Press.

Dongarra, J.J., Moler, C.B., Bunch, J.R. and Stewart, G.W. (1979). LINPACK - Users' Guide. SIAM Press.

Einarsson, B. (1977). Bibliography on numerical software. Report UCB/ERL M77/19 Electronics Res. Lab., Berkeley.

Forsythe, G.E., Malcolm, M.A. and Moler, C.B. (1977).
Computer methods for mathematical computations. Prentice-Hall.

Forsythe, G.E. and Moler, C.B. (1967). Computer solution of linear algebraic systems. Prentice Hall.

Garbow, B.S., Boyle, J.M., Dongarra, J.J. and Moler, C.B. (1977).
Matrix Eigensystem Routines - EISPACK Guide Extension. Lecture Notes in Computer Science 51. Springer-Verlag.

Kuki, H. (1971). Mathematical function sub-programs for basic system libraries. Objectives, constraints, and trade-off, pp.187-199.
In Mathematical Software. J.R. Rice (Ed.). Academic Press.

Lawson, C.L. and Hanson, R.J. (1967). Solving least squares problems. Prentice-Hall.

Lawson, C.L., Hanson, R.J., Kincaid, D.R. and Krogh, F.T. (1979).
Basic linear algebra subprograms for Fortran usage. TOMS 5, pp.308-323.

More, J.J. (1979). On the design of optimization software. Report DAMPT 79/NA8. Dept. Appl. Math. and Theoretical Phys.
University of Cambridge.

Reid, J.K. and Hopper, M.J. (1979). Production testing and documentation of mathematical software. Harwell report CSS 75.

Rice, J.R. (1977). ELLPACK: A research tool for elliptic partial differential equations software, pp.319-342. In Mathematical Software III.
J.R. Rice (Ed.). Academic Press.

Rice, J.R. (1978). Software for numerical computation. In Research Directions in Software Technology. Ed. P. Wegner. MIT Press.

Rice, J.R., Gear, C.W., Ortega, J., Parlett, B., Schultz, M., Shampine, L.F., Wolfe, P. and Traub, J.F. (1979). Numerical Computation: Its nature and research directions. SIGNUM Newsletter.
Special Issue. February 1979.

Shampine, L.F. and Gordon, M.K. (1975). Computer solution of ordinary differential equations: the initial value problem. Freeman.

Smith, B.T., Boyle, J.M. and Cody, W.J. (1974). The NATS approach to quality software, pp.393-405. In Software for Numerical Mathematics.
D.J. Evans (Ed.). Academic Press.

Smith, B.T., Boyle, J.M., Garbow, B.S., Ikebe, Y., Klema, V.C. and Moler, C.B. (1974). Matrix Eigensystem Routines - EISPACK Guide.
Lecture Notes in Computer Science 6. Springer-Verlag.

Stewart, G.W. (1977). Research, development, and LINPACK, pp.1-14.
In Mathematical Software III. J.R. Rice (Ed.). Academic Press.

Sweet, R.A. (1979). A survey of the state of software for partial differential equations. Report STAN-CS-79-704, Computer Science Dept. Stanford University.

Wilkinson, J.H. and Reinsch, C. (1971). Handbook for automatic computation. Vol. II. Linear Algebra. Springer-Verlag. Berlin.

Libraries. General, Specific Libraries

Barinka, L.L. (1975). Some experience with constructing, testing, and certifying a standard mathematical subroutine library. TOMS 1, pp.165-177.

Bolstad, J.H., Chan, T.F., Coughran, W.M. Jr., Grosse, E.H., Heath, M.T. and Luk, F.T. (1977). Numerical Analysis Program Library Users' Guide. Report SCS-SCIP User Note 82. Stanford.

Chan, T.F., Coughran, W.M. Jr., Grosse, E. and Heath, M.T. (1978). A numerical library and its support. Report STAN-CS-78-673. Computer Science Dept. Stanford.

Cody, W.J. (1974). The construction of numerical subroutine libraries. SIAM Rev. 16, pp.36-46.

Cooper, J.R.A. (1979). NPL Algorithms Library Brief Guide - 1979. Report CSU 3/79. NPL, Teddington, England.

Dickinson, A.W., Herbert, V.P., Pauls, A.C. and Rosen, E.M. (1971). The development and maintenance of a technical subprogram library, pp. 141-151. In Mathematical Software. J.R. Rice (Ed.). Academic Press.

Ford, B. and Bentley, J. A library design for all parties, pp.3-20. In Numerical Software - Needs and Availability. D.A.H. Jacobs (Ed.). Academic Press.

Ford, B. (1972). Developing a numerical algorithms library. Bulletin of the Inst. Maths. and its Applics. 8, pp.332-336.

Ford, B. and Sayers, D.K. (1976). Developing a single numerical algorithms library for different machine ranges. TOMS 2, pp.115-131.

Ford, B. and Hague, S.J. (1974). The organisation of numerical algorithms libraries, pp.357-372. In Software for Numerical Mathematics. D.J. Evans (Ed.). Academic Press.

Fox, P.A., Hall, A.D. and Schryer, N.L. (1978). The PORT mathematical subroutine library. TOMS 4, pp.104-126.

Gentleman, W.M. and Traub, J.F. (1968). The Bell Laboratories Numerical Mathematics Program Library Project, pp.485-490. In Proc. ACM National Conf. 1968.

Gill, P.E., Murray, W., Picken, S.M. and Wright, M.H. (1979). The design and structure of a Fortran Program Library for Optimization. TOMS 5, pp.259-283.

Haskell, K.H. and Jones, R.E. (1977). Brief instructions for using the Sandia Mathematical Subroutine Library (version 7.2). Report SAND77-1441, Sandia Labs. Albuquerque.

Hopper, M.J. (Compiler)(1979). Harwell Subroutine Library. A catalogue of subroutines (1979). Report AERE-R.9185, Harwell, UKAEA, HMSO, London.

IBM (1974). Subroutine Library - Mathematics. User's Guide. SH12-5300-1, IBM.

JPL (1975). JPL Fortran V. Subprogram Directory. Edition 5. Report 1846-23. Rev. A. Jet Propulsion Laboratory.

IMSL (1975). IMSL Library 1. Reference Manual (in 2 volumes). IMSL LIB1.0005. International Mathematical and Statistical Libraries Inc. Houston, Texas.

Newbery, A.C.R. (1971). The Boeing Library and Handbook of Mathematical Routines, pp.153-169. In Mathematical Software. J.R. Rice (Ed.). Academic Press.

NUMAL (1977). NUMAL. A library of numerical procedures in Algol 60. Volume 0. General information and indices. Second version. Mathematical Centre, Amsterdam.

Wilkes, M.V., Wheeler, D.J. and Gill, S. (1951). The preparation of programs for an electronic digital computer. Addison-Wesley.

Coding, testing, portability, certification, etc.

ANSI (1966). American National Standard programming language FORTRAN. ANSI X3.9-1966. Published by American National Standards Institute. Inc.

ANSI (1978). American National Standard programming language FORTRAN. ANSI X3.9-1978. Published by American National Standards Institute. Inc.

Bentley, J. and Ford, B. (1976). On the enhancement of portability within the NAG project - a statistical survey, pp.505-528, in Portability of Numerical Software. W. Cowell (Ed.). Springer-Verlag.

Crowder, H., Dembo, R.S. and Mulvey, J.M. (1979). On reporting computational experiments with mathematical software. TOMS 5, pp.193-203.

Duff, I.S. (1979a). Practical comparisons of codes for the solution of sparse linear systems, pp.107-134. In Duff and Stewart (1979).

Duff, I.S. and Reid, J.K. (1979). Performance evaluation of codes for sparse matrix problems, pp.121-135. In Performance Evaluation of Numerical Software, L.D. Fosdick Ed. North-Holland.

Einarsson, B. (1978). Bibliography on the evaluation of numerical software. Report from FOA, Tumba, Sweden.

Ford, B. (Editor)(1978). Machine characteristics and their parameterisation in numerical software. TOMS 4, pp.100-103.

Ford, B., Reid, J.K. and Smith, B.T. (Editors)(1977). Three proposed amendments to the Draft Proposed ANS Fortran Standard. SIGNUM Newsletter, 12, pp.18-20.

Fosdick, L.D. (1979). Detecting errors in programs, pp.77-87. In Performance Evaluation of Numerical Software. L.D. Fosdick (Ed.). North-Holland.

Gregory, R.T. and Karney, D.L. (1969). A collection of matrices for testing computational algorithms. Wiley.

Kernighan, B.W. and Plauger, P.J. (1978). The elements of programming style. Second edition. McGraw-Hill.

Lawson, C.L. and Reid, J.K. (1976). Two numerical analysts' views on the draft proposed ANS Fortran, pp.257-268. In Portability of Numerical Software. W. Cowell (Ed.). Springer-Verlag.

Niblett, B. (1979). Portability considerations in software design. Report AERE-R.9437, AERE Harwell, HMSO, London.

Parlett, B.N. and Wang, Y. (1975). The influence of the compiler on the cost of mathematical software - in particular on the cost of triangular factorization. TOMS 1, pp.35-46.

Reid, J.K. (Editor)(1979). Functions for manipulating floating-point numbers. Report CSS 76, AERE Harwell.

Smith, B.T. (1976). Fortran poisoning and antidotes, pp.178-256, in Portability of Numerical Software. W. Cowell (Ed.). Springer-Verlag.

Software Tools

Aird, T.J., Battiste, E.L. and Gregory, W.C. (1977). Portability of mathematical software coded in Fortran. TOMS 3, pp.113-127.

Boyle, J.M. and Dritz, K.W. (1974). An automated programming system to facilitate the development of quality mathematical software, pp.542-546. Proc. IFIP Congress 1974. North Holland.

Dorrenbacher, J., Paddock, D., Wisneski, D. and Fosdick, L.D. (1974). POLISH, A Fortran program to edit Fortran programs. Report CU-CS-050-74. Dept. Computer Sci. University of Colorado, Boulder.

Du Croz, J.J., Hague, S.J. and Siemieniuch, J.L. (1976). Aids to portability within the NAG project, pp.389-404. In Portability of Numerical Software. W. Cowell (Ed.). Springer-Verlag.

Fosdick, L.D. (1974). BRNANL, a Fortran program to identify basic blocks in Fortran programs. Report CM-CS-040-74. Dept. of Computer Science, University of Boulder, Colorado.

Hague, S.J. (1978). Software Tools, pp.57-79. In Numerical Software - Needs and Availability. D.A.H. Jacobs (Ed.). Academic Press.

Kernighan, B.W. and Plauger, P.J. (1976). Software Tools. Addison-Wesley.

Krogh, F.T. (1976). Features for FORTRAN portability, pp.361-367. In Portability in Numerical Software. W. Cowell (Ed.). Springer-Verlag.

Marlow, S. and Reid, J.K. (1978). A Fortran subroutine for comparing two files. Report AERE-R.8971, AERE, Harwell.

Miller, W. (1975). Software for roundoff analysis. TOMS 1, pp.108-128.

Ryder, B.G. (1974). The PFORT verifier. Software Practice and Experience, 4, pp.359-377.

Woodward, M.R., Hedley, D. and Hennell, M. (1978). Observations and experience of path analysis and testing of programs. Report. Univ. of Liverpool, England.

Interrelationship with Users

Bailey, C.B. and Jones, R.E. (1975). Usage and argument monitoring of mathematical library. TOMS 1, pp.196-209.

Erisman, A.M. (Chairman)(1979). Panel Discussion: The use of mathematical software outside the mathematical software community, pp.273-283. In Performance Evaluation of Numerical Software. L.D. Fosdick (Ed.). North-Holland.

Fletcher, R. and Hebden, M.D. (1974). Setting up a numerical advisory service, pp.413-421. In Software for Numerical Mathematics. D.J. Evans (Ed.). Academic Press.

Gaffney, P.W. (1979). Information and advice on the numerical software available for the fusion energy program at Oak Ridge. Report ORNL/CSD/TM-93. Computer Sciences Division, Oak Ridge, Tenn.

Hopper, M.J. (1976). TSSD, a typesetting system for scientific documents. Report AERE-R.8574, AERE Harwell, HMSO, London.

Kemp, P. (1978). Libraries: The user interface, pp.39-56. In Numerical Software - Needs and Availability. D.A.H. Jacobs (Ed.). Academic Press.

Lill, S.A. (1974). User documentation for a general numerical library: the NAG approach, pp.423-432. In Software for Numerical Mathematics. D.J. Evans (Ed.). Academic Press.

McIntyre, R.W. and Tate, T. (1978). The value of a numerical library to a large industrial organization, pp.81-95. In Numerical Software - Needs and Availability. D.A.H. Jacobs (Ed.). Academic Press.

Newman, N. and Lang, T. (1976). Documentation for computer users. Software Practice and Experience 6, pp.321-326.

Whelan, J.P. (1978). Using the NAG library in industrial research, pp.97-105. In Numerical Software - Needs and Availability, D.A.H. Jacobs (Ed.). Academic Press.

A PORTABLE VECTOR-CODE FOR AUTONOMOUS MULTIGRID MODULES

P.W. Hemker, P. Wesseling and P.M. de Zeeuw

We present the description of the implementation of two multigrid modules for the efficient solution of discretized elliptic PDEs.
The equations may be non-selfadjoint and the coefficients are arbitrary.

The modules can be used just like any standard subroutine for solving systems of linear equations. The user has to specify only the matrix and the rhs. Using these autonomous (black-box) subroutines the user remains unaware of the underlying multigrid method (still obtaining the multigrid efficiency).

An implementation in standard ANSI-FORTRAN which vectorizes to a satisfactory degree by auto-vectorization is prepared for use in vector computers. Results are given for the CRAY1 and the CD CYBER 205. Comparisons are made for the two algorithms and the two machines.

Lit.: P.W. Hemker, P. Wesseling and P.M. de Zeeuw, "A portable vector-code for autonomous multigrid modules", Report NW 154/83, Mathematical Centre, Amsterdam.

AN INTRODUCTION TO VECTOR PROCESSING WITH APPLICATION TO NUMERICAL METHODS

M.J. Kascic, Control Data, Minnesota.

Numerical Analysis has been revolutionized in the twentieth century by the invention of the computer. It has become routine to perform actual calculations several orders of magnitude faster and far more reliably than by hand. Naturally, this has changed the nature of what a computationally solvable problem is. This, in turn (ought to), affect the development of numerical methods, at least in so far as their goal is practical calculation.

The development of vector processors, such as the CDC CYBER 205 has changed this scenario in two significant ways.

1. Vector processing on the CYBER 205 theoretically, can achieve a two order of magnitude increase in performance over scalar (traditional) processing. This is larger than that achieved by the first computers over hand calculation.

2. In order to achieve this performance, a vector discipline must be imposed on the data structure of the problem.

We will discuss how the concept of vector processing is implemented on the architecture of the CYBER 205. This discussion will not assume any previous knowledge of computer architecture.

We will then present results for the performance of several elementary algorithms of Numerical Analysis. The goal here is to give a "feel" for what is a hard/expensive problem to solve on such a class of computer and what is not. Allusion will be made to the possibility of developing a semantic vectorization methodology which allows a conceptually direct link between mathematical structure and vector computational ones.

VORTON DYNAMICS: A CASE STUDY OF DEVELOPING A FLUID DYNAMICS MODEL FOR A VECTOR PROCESSOR

M.J. Kascic, Control Data, Minnesota.

One of the fundamental problems of three-dimensional fluid dynamics is the study of the growth of enstrophy, i.e., the L^2 norm of the vorticity. Unlike two dimensional flows, not only is enstrophy not conserved, but it does to infinity even in an inviscid fluid, provided that there is vorticity present at time = 0.

Ideally one would like to know at what rate the enstrophy goes to infinity since this impacts the question of well-posedness of the Euler equations. One is also interested in the Hausdorff dimension of the subset of the fluid where the enstrophy grows since it is now believed that its dimension lies between 2 and 3. Finally one would like to study the conformity (or lack thereof) with Kolmogoroff's rolloff exponent in the inertial range of wave numbers.

Various aspects of this problem have been discussed by several authors [1], [2], [3], [4]. The Vorton Dynamics model has been designed to study this problem using the resources of the CDC CYBER 205 computer.

The vorticity field is modelled as a set of independent mobile Dirac measures which carry the vorticity. These "particles" are called vortons Conceptually they differ from either the blobs or one-dimensional filaments studied previously. The vortons are Lagrangian in the sense that their position is not fixed. They can wander throughout the grid as the Euler equations move them. The velocity is considered an ancillary field which is measured on an Eulerian grid so that velocity gradients can be numerically calculated.

The Euler equations then use the velocity gradient to update the strength of the vortons. Poisson's equation is then solved to recoupe the new velocity from the curl of the vorticity.

Since we are interested in the transient behaviour of the vortons, simple explicit algorithms are used whenever possible. The integral of vorticit and the total kinetic energy are used as stability checks.

There are several seeming drawbacks to implementing this model on a vector computer such as the CYBER 205.

1. Whatever computer one uses, the use of explicit algorithms demands small time steps. Unless the model runs incredibly fast, its cost is prohibitive. In particular three Poisson equations have to be solved each time step on an N^3 grid. The thought of solving Poisson's equation on a 64^3 grid, say 30,000 times, is enough to discourage some.
2. The interaction between the Lagrangian vortons and the Eulerian grid is computationally non-linear. This is usually considered a bad thing to do on a vector computer. In particular, the seemingly trivial problem of converting floating point numbers that signify position into integers that represent "box" coordinates within a grid structure is often more costly than the calculation of physical processes.

We shall discuss how these problems have been attacked on the CYBER 205 and to what degree they have been "solved". We shall also discuss preliminary results of the model.

REFERENCES

1. A. J. Chorin, Estimates of Intermittency, Spectra, and Blow-Up in Developed Turbulence, Comm. on Pure and Applied Math. Vol. XXXIV, 1981
2. B. Couet, Evolution of Turbulence by Three-Dimensional Numerical Particle-Vortex Tracing, SU-IPR Report #793, Inst. for Plasma Research, Stanford U., 1979
3. A. Dold & B. Eckmann, Turbulence Seminar, Berkeley 1976, Springer Verlag ISBN 3-540-08445-2
4. P. Saffman, Dynamics of Vorticity, J. Fluid Mech. Vol. 106, 1981

ERROR-FREE SOLUTION OF ALGEBRAIC PROBLEMS IN FLOATING-POINT ARITHMETIC: DEFECT CORRECTION AND INTERVAL COMPUTATION IN THE ASSESMENT OF COMPUTATIONAL ERRORS

H.J. Stetter, Technische Universität Wien.

Many of the usual algorithms for numerical problems have an iterative structure: For a "given" approximation $x^{(i)}$ of the true solution x^* of the specified problem, the algorithm computers a "better" approximation $x^{(i+1)}$. On the other hand, the assessment of the error of an approximation $\tilde{x}$ for x^* is equivalent to the following "secondary problem":

Given an approximation $\tilde{x}$ of x^* ,

compute $\left.\begin{matrix}\text{an approximation}\\ \text{a bound}\end{matrix}\right\}$ for the error $\tilde{x} - x^*$.

Obviously, error assessment and iterative improvement are two sides of the same thing: The correction $\Delta x^{(i)} := x^{(i)} - x^{(i+1)}$ in a convergent iteration algorithm for x^* may also be regarded as an approximation ("estimate") of the error $x^{(i)} - x^*$ of $x^{(i)}$.

For the computation of strict bounds for the results of numerical problems, interval mathematics has been the classical approach: Computations with real numbers are replaced by computations with intervals of real numbers such that the result interval contains the true solution x^*.

Naive interval algorithms produce unrealistically large result intervals due to

a) the unavoidable expansion of the spans of intervals in addition and subtraction,

b) the occurence of "dependent" intervals.

Spans of intervals remain small only in multiplicative operations with small intervals close to the origin. Therefore, interval mathematics should only be used in the secondary problem, i.e. for the computation of bounds for errors or corrections.

The classical approach to the computational solution of the secondary problem is the following:

a) Compute the defect (residual) $d(\tilde{x})$ of the approximation $\tilde{x}$,

b) compute the approximate effect of this defect on the solution.

The result of b) may be used to correct $\tilde{x}$ so that an iteration is possible; the approach is often called (iterative) defect correction. The accuracy which it may potentially achieve depends on the accuracy of a) while shortcomings of b) may be compensated by further iterations.

For the application of interval arithmetic, a formulation of the secondary problem in fixed point form is more appropriate. If

$$\Delta x = T(d(\tilde{x}))\ \Delta x$$

in such a formulation, then the establishing of

$$T(d(\tilde{x}))\ X \subset X$$

for an interval X implies

$$\Delta x \in X \qquad (\text{or } x^* \in \tilde{x} - X)\ .$$

under weak assumptions on the mapping T.

This approach leads to feasible algorithms for the computation of realistic strict error bounds for the results of numerical computations.

ERROR-FREE SOLUTION OF ALGEBRAIC PROBLEMS IN FLOATING-POINT ARITHMETIC: HIGH-ACCURACY ALGORITHMS WITH GUARANTEED RESULTS.

H.J. Stetter, Technische Universität Wien.

In a fixed wordlength, floating point number system, part of the information in a correction Δx is lost when this correction is added to a current approximation $\tilde{x}$. It is therefore essential to retain the correction values in a suitable form. Furthermore, it must be possible to compute the defect of $\tilde{x} = x_0 + \Delta_1 x + \Delta_2 x + .. + \Delta_m x$ from the set of quantities x_0, $\Delta_1 x$, $\Delta_2 x$,.., $\Delta_m x$ to a high relative accuracy even in the presence of strong cancellations.

In most algebraic problems (linear equations, nonlinear equations, evaluation of rational expressions, eigenproblems), the computation of the defect may be reduced to the computation of several scalar products. Thus the representation of $\tilde{x}$ as a sum creates no problems.

The accurate evaluation of a scalar product with strong cancellations - as they are to be expected in the defect of $\tilde{x} \approx x^*$ - may be achieved recursively (Bohlender's algorithm) or in a quasi-fixed-point computation with a "long accumulator".

With the approach described in the first lecture and the tools just mentioned, it becomes possible to design algorithms which generate solutions of algebraic problems which are accurate "to the last bit" and whose accuracy is guaranteed. Actually the computational solutions arise in the form of intervals whose bounds are neighboring floating-point numbers.

For systems of linear (algebraic) equations, we establish the secondary problem in fixed point form; various details of the implemented algorithm are explained.

The same approach may be extended to systems of nonlinear (algebraic) equations via an interval version of Newton's method.

The evaluation of ill-conditioned rational expressions, e.g. of a polynomial near one of its zeros, may also be reduced to a similar high-accuracy algorithm.

Naturally, all these "last bit accurate", guaranteed results refer to the mathematical problem with the floating-point data specified in the call of the subroutine. This raises the question of the relevance of these results in an ill-conditioned situation, with low-accuracy data.

It is clear that in such a situation the results of the specified mathematical problem have a low meaningful accuracy; but it is also clear that it is better not to add further perturbations by uncontrolled (and probably large) round-off errors. On the contrary, the elimination of these effects in the high-accuracy algorithms permits safe experimental studies of the sensitivity of the results with respect to data variations.

Thus the true meaningful accuracy induced by the data may often be determined, perhaps even automatically in simpler situations.

The extension of this approach to composite problems and to analytic problems in function spaces has been proposed and studied; the inherent difficulties and some partial solutions will be exhibited.

VECTORIZATION OF NUMERICAL SOFTWARE ON THE CRAY 1 AND THE CYBER 205

H.A. van der Vorst, Academisch Computer Centrum, Utrecht.

The currently available supercomputers CRAY 1 and CYBER 205 allow us to solve much more complicated and bigger numerical problems as the conventional computers do. Both types of computers are particularly suited for many linear algebra problems, which are often the time-consuming kernels in many large problems.

In order to achieve a favourable performance (i.e., a large MFLOPS rate) one usually has to reformulate the method of solution, taking into account the features of the particular computer. Though both computers are so-called vector computers (pipeline-concept), their inital differences are so fundamental that not only the formulation of an algorithm may depend on the choice of computers, but also the choice of algorithm may depend there upon.

In this contribution we regard the relevant properties of both super-computers and we will show their effects on the treatment of some numerical linear algebra problems.

DEELNEMERS

1.	Axelsson, A.O.H.	Katholieke Universiteit	Nijmegen
2.	Boerstoel, J.W.	NLR	Amsterdam
3.	Böhmer, K.	Universität Marburg	BRD
4.	Bollen, J.A.M.	Technische Hogeschool	Enschede
5.	Dekker, K.	CWI	Amsterdam
6.	Dekker, Th.J.	Universiteit van Amsterdam	Amsterdam
7.	Duff, I.	AERE, Harwell	UK
8.	Dupont, T.	University of Chicago	USA
9.	Geurts, A.J.	Technische Hogeschool	Eindhoven
10.	Ginneken, C.J.J.M.	Technische Hogeschool	Eindhoven
11.	Griend, J.A. van de	Rijks Universiteit	Leiden
12.	Groen, P.P.N. de	Vrije Universiteit	Brussel
13.	Groot, J. de	Nat. Lab. Philips	Eindhoven
14.	Hemker, P.W.	CWI	Amsterdam
15.	Heijer, C. den	Philips	Eindhoven
16.	Houwen, P.J. van der	CWI	Amsterdam
17.	Hundsdorfer, W.H.	Rijks Universiteit	Leiden
18.	Jansen, J.K.M.	Technische Hogeschool	Eindhoven
19.	Kascic, M.J.	Control Data	Minnesota
20.	Kok, J.	CWI	Amsterdam
21.	Kraaijevanger, J.F.B.M.	Rijks Universiteit	Leiden
22.	Laan, C.G. van der	Rijks Universiteit	Groningen
23.	Loon, P.H. van	Technische Hogeschool	Eindhoven
24.	Maten, E.J.W. ter	Rijks Universiteit	Utrecht
25.	Mattheij, R.M.M.	Katholieke Universiteit	Nijmegen
26.	Ouden, A.C.B. den	ENR	Petten
27.	Peeck, H.H.	KNMI	De Bilt
28.	Piers, W.J.	NLR	Amsterdam
29.	Riele, H.J.J. te	CWI	Amsterdam
30.	Schippers, H.	Nat. Lucht en Ruimtevaartlab.	Marknesse
31.	Sluis, A. van der	Rijks Universiteit	Utrecht
32.	Sommeijer, B.P.	CWI	Amsterdam
33.	Spijker, M.N.	Rijks Universiteit	Leiden
34.	Stelling, G.S.	Rijkswaterstaat	Rijswijk
35.	Stetter, H.J.	Technische Universität, Wien	Oostenrijk
36.	Sullivan, F.E.	NBS, Washington	USA
37.	Toint, Ph.	FNDP	Namen
38.	Vaatstra, W.	Nat. Lucht en Ruimtevaartlab.	Marknesse
39.	Veldhuizen, M. van	Vrije Universiteit	Amsterdam
40.	Veltkamp, G.W.	Technische Hogeschool	Eindhoven
41.	Verwer, J.G.	CWI	Amsterdam
42.	Vooren, A.I. van de	Rijks Universiteit	Groningen
43.	Vorst, H.A. van der	Academisch Computer Centrum	Utrecht
44.	Vuik, C.	Rijks Universiteit	Utrecht
45.	Weber, C.	Nat. Lab. Philips	Eindhoven
46.	Wesseling, P.	Technische Hogeschool	Delft
47.	Wolkenfelt, P.H.M.	Technische Hogeschool	Eindhoven
48.	Wubs, F.W.	CWI	Amsterdam
49.	Zeeuw, P.M. de	CWI	Amsterdam