NUMERICAL MATHEMATICS AND COMPUTER SCIENCE

J. F. Traub
Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pa.

This work was supported in part by the Office of Naval Research under contract N 00014-67-A-0314-0010, NR 044-422 and by the National Science Foundation under grant GJ-32111.

1. INTRODUCTION

Numerical Mathematics is a very broad field. In this paper we focus on aspects of numerical mathematics which are related to computer science. If our focus were on numerical mathematics as a mathematical subject, we would cover rather different ground, for that aspect of the subject has a vast literature which we do not touch on here. The reader interested in the relation between numerical mathematics, applied mathematics, and computation is referred to Cohen [4].

Computer Science has strong mathematical components such as automata theory, computational complexity, numerical mathematics, and symbolic mathematics. However, much of computer science is rather close to engineering in the sense that the creations must work in the real world. Similiarly, numerical mathematics ranges from strongly mathematical areas to those with a strong engineering flavor.

One of the central concerns of computer science is algorithmic analysis or algorithmics (Traub [25, p. 1]). As we shall see, numerical mathematics deals with algorithms for continuous processes.

2. HISTORICAL PERSPECTIVE

The roots of numerical mathematics are ancient dating back thousands of years (Davis [7]). Yet modern numerical mathematics as a separate discipline dates back less than 25 years. The modern discipline is distinguished by:

(1) synthesis and analysis of algorithms rather than a focus on synthesis alone

(2) analysis of algorithms using finite precision numbers

Some have dated the modern field back to 1947. That year marked the establishment of an Institute for Numerical Analysis at UCLA as well as the publication of a paper by Von Neumann and Goldstine [27] on the error analysis of matrix inversion using finite precision arithmetic. Indeed it had been believed (Householder [11], Traub [24], Wilkinson [29]) that 1947 marked the first use of the phrase "numerical analysis", but Businger [6] pointed out that the phrase had been used by Lagrange. Nevertheless 1947 does seem to be a reasonable year to mark the beginning of modern numerical mathematics, and in 1972 a joint SIAM-SIGNUM Symposium will celebrate the 25th anniversary.

Both numerical analysis and numerical mathematics are used to denote the discipline although the latter appellation seems to be superseding the former.

3. A DEFINITION OF NUMERICAL MATHEMATICS

We give a formal definition of the discipline, as it is generally practiced today and then suggest how the scope of the field should be broadened.

Numerical mathematics is the theory and practice of the efficient calculation and error appraisal of approximate solutions of continuous mathematical problems.

An equivalent name for the subject is the Analysis of Continuous Algorithms. Two of the words in this definition, continuous and efficient, warrant some amplification. By continuous we mean the problem input and/or output belong to a continuum. For example, the solution of polynomial equations with integer coefficients belongs to numerical mathematics since the zeros belong to a continuum. The solution of diophantine equations does not belong since the input and output are integers. The distinction between these two problems is a real one since the second problem is combinatorial in nature and totally different techniques are used.

The adjective efficient is critical. One of the primary differences between mathematics and numerical mathematics is that mathematics lacks the concept of efficiency. There is elegance and simplicity of proof, but not efficiency.

Mathematics distinguishes between constructive and non-constructive proofs. A constructive proof may be used as an algorithm but it is usually not an efficient one. For example, the proof of the theorem that continuous functions on a closed interval may be approximated arbitrarily closely by polynomials is based on Bernstein polynomials. This construction is not used in numerical practice since it may require a polynomial of degree ten million to achieve eight-place accuracy. A best-fit polynomial of degree perhaps six would achieve such accuracy.

The adjective continuous serves to delineate the discipline as it is generally practiced today. However, the algorithmic approach, long traditional in numerical mathematics, is useful for problems where the data does not belong to a continuum. Numerical mathematicians can contribute to the study of such problems.

Before leaving our discussion of the extent of numerical mathematics, we should mention some neighboring fields. Discrete mathematics (also called combinatorial mathematics) deals with discrete data (such as integers). Symbolic mathematics deals with symbolic data. Numerical mathematics, discrete mathematics, and symbolic mathematics are all subfields of the mathematics of computation which consists of the intersection of mathematics and computer science.

For the remainder of this paper, we discuss four major components of numerical mathematics. They are

1. Foundations
2. Synthesis and Analysis of Algorithms
3. Analysis of Error
4. Programs and Program Libraries

Others might choose different components. Many numerical mathematicians believe (2) and (3) comprise the field. We believe that the study of (1) provides theoretical underpinnings while (4) makes results achieved in numerical mathematics accessible to other fields.

4. FOUNDATIONS

We discuss two examples of foundation subjects: finite precision number systems and computational complexity.

a. Finite Precision Number Systems

The finite precision floating point numbers are the numbers used in computation and their properties are quite different from the real numbers of mathematical analysis. We give two simple illustrations:

The familiar algebraic laws need not hold. Thus $a \neq b$, $x + a = x + b$ may be simultaneously true. A perceptive discussion is given by Van Wijngaarden [26]. Knuth [14, Vol. 2, Chapter 4] contains a wealth of material.

Another phenomenon is that the spacing between adjacent numbers is not uniform. Consider eight place numbers. The nearest neighbor to a number of size 10^{-8} is 10^{-15} distant. The nearest neighbor to a number of size 10^{8} is 10^{1} distant. Since $10 > \pi$, the calculation of the sine of a rounded number of magnitude 10^{8} is meaningless with such numbers.

The "real numbers" and "real analysis" may be viewed as a limiting case of finite precision numbers and numerical mathematics. In this sense numerical mathematics may be regarded as "richer". For example, Taylor's formula is a special case of Newton's interpolation formula from which it follows in the confluent limit.

The real numbers of analysis were introduced to simplify mathematics. They are an abstraction placed on a firm foundation by Dedekind, Weierstrass and other analysts of the last century. The implications of real numbers for real analysis took decades to develop. The implications of finite precision numbers for numerical mathematics are just beginning to be explored. Brent [3] and Matula [16] are doing interesting work here.

b. Computational Complexity

We emphasized above the central role that efficiency plays in numerical mathematics. This focus on efficiency naturally leads us to study what are the most efficient, that is, the optimal processes. This area of study is contained in computational complexity. Some of the most exciting current work in numerical mathematics lies in this area. In March 1972, an IBM Symposium on Complexity of Computer Computations summarized recent progress [12]. In [22] we suggested that optimality theory for algebraic (or combinatorial) processes be called algebraic computational complexity, while optimality theory for analytic (or continuous) processes be called analytic computational complexity.

Much, but not all, of algebraic computational complexity deals with algorithms for problems which are of interest to numerical mathematics. Such problems include the multiplication of matrices and the evaluation of polynomials. Although these processes are combinatorial, they are very common components of the algorithms of numerical mathematics. A survey may be found in Borodin [2].

Research on analytic computational complexity dates to the early sixties and predates most of the algebraic results. A survey may be found in Traub [22]. The results achieved so far deal almost exclusively with optimality of iteration algorithms for the calculation of fixed-points.

We cite here only one very recent result due to Kung [15]. Kung defines a multiplication efficiency index for any multivariate rational iteration which takes into account the rate of convergence of the iteration and the number of multiplications or divisions per iteration step. He proves that the multiplication efficiency is always bounded by unity.

5. SYNTHESIS AND ANALYSIS OF ALGORITHMS

This has been a core area for numerical mathematics and will continue to be so. It is relatively easy to invent algorithms (synthesis). The trick is to invent good ones and prove they are good (analysis). Knuth considers the analysis of algorithms a central one in computer science and is writing a seven volume set on this subject [14]. He is not attempting to include the algorithms of numerical mathematics.

In practically every problem area, the best algorithm we have today has been introduced in the last 15 years. Some examples are the QR algorithm of Francis and Kublanovskaya with improvements by Parlett, Kahan, and Wilkinson; the algorithms of Fletcher and Powell and their co-workers on minimization; rational extrapolation algorithms for ordinary differential equations due to Bulirsch, Gragg, and Stoer; Golub's analysis of Householder Transformations for least squares problems, the analysis of finite element methods for partial differential equations due to Ciarlet, Fix, Schultz, Strang, and Varga; the efficient calculation of finite Fourier Series by the Fast-Fourier Transform of Cooley and Tukey. These are only a few instances of the great progress that has been made. Even when the algorithms are classical, as in the case of Gaussian elimination, it is only recently that a complete analysis has been performed.

We list some general criteria for a good algorithm:

1. The algorithm should do well when performed in finite precision.

2. It should be robust, that is, it should degrade gracefully if it is used on a problem which is near the boundary of the problem space for which the algorithm is designed.

3. The hypotheses under which the algorithm works should be easily verifiable. For example, since the symmetry of a matrix is easily verified, an eigenvalue algorithm which converges for any symmetric matrix is desirable. On the other hand, it is hard to verify a hypothesis on the positivity of a second derivative over an interval.

4. The algorithm should be efficient by some appropriate measure.

5. The algorithm should always work for a reasonable "large" class of problems.

Such algorithms are natural candidates for implementation as computer routines.

In certain areas we are now at the stage where we have excellent algorithms meeting these specifications. A major recent achievement has been to design such algorithms for such areas as systems of linear algebraic equations, algebraic eigenvalues, and polynomial zeros. Elsewhere we have a long way to go. Examples are most non-linear problems, most multivariate problems such as constrained and unconstrained optimization, partial differential equations, ordinary differential equations. For some problems we may never find algorithms satisfying our five criteria. Gentleman [8] has surveyed numerical algorithms and selected areas for which good algorithms do or do not exist.

Almost without exception algorithms have been created for sequential machines. Only recently have algorithms been designed for parallel and vector machines. A survey of work on parallel algorithms may be found in Miranker [17].

6. ANALYSIS OF ERROR

An important element in numerical mathematics is the appraisal of error. We consider both mathematical and machine-aided error analysis.

Although mathematical error analysis began with the celebrated paper by Von Neumann and Goldstine [27] published in 1947, it was not until the mid to late fifties that activity really picked up. Extensive bibliographies may be found in Rall [20].

One of the most important ideas in mathematical error analysis is that of backward error analysis proposed by Givens [10] and brilliantly exploited by Wilkinson [28] to analyze the rounding error of algebraic processes. In backward error analysis we ask what problem have we solved exactly and how far is this problem from the one we set out to solve? This may be contrasted with <u>forward error analysis</u> where we ask by how much the calculated answer differs from the true answer. Backward analysis is often easier to perform and the answers are often more useful. Using backward analysis, Wilkinson [30] was able to give a complete a priori analysis of the solution of a system of linear algebraic equations by Gaussian elimination with pivoting.

We turn to machine-aided error analysis. A dream has been automatic error analysis -- let the computer do the work. Papers in machine-aided error analysis first appeared in the mid to late fifties. An annotated bibliography is given by Bright [5] and extensive references appear in Rall [20].

Special arithmetics are <u>tools</u> used in machine-aided error analysis. The purpose of a special arithmetic is to enable us to appraise the error at the same time that we perform the calculation. Examples are significance arithmetic (Ashenhurst [1]) and range arithmetic (Moore [18]). There are some serious problems. For example, a difficulty with range arithmetic is that the intervals in which the answers are guaranteed to lie increase greatly in size as the computation proceeds.

Work has started on systems for automatic error analysis. P. Richman has been studying the limitations, design and implementation of a system for computing numerical results to within any desired tolerance. His basic tool is variable-precision range arithmetic.

7. PROGRAMS AND PROGRAM LIBRARIES

Algorithms may be specified in various ways. One mode is by a computer program. This has a number of advantages. It is the nature of a program that all details must be spelled out completely. When the program is run, one may verify that the algorithm performs as required. The publication of algorithms as programs may be viewed as a means of getting frontier results to consumers of numerical mathematics in a useful way. Indeed the creators of the scientific programming language Algol [19] were well aware of the usefulness of programs as a medium of information dissemination.

Refereed programs are published in BIT, the Communications of the ACM, the Computer Journal and Numerische Mathematik. Those in the latter journal are of exceptionally high quality and will appear in a Handbook of Automatic Computing. There are a number of difficulties. The programs have been written in Algol and hence are not immediately useable by Fortran programmers. There are often a number of programs purporting to solve the same problem and a selection of a particular program must still be made.

This leads us to computer program libraries. Conceptually, this is a fine idea. In practice, however, these libraries are often inadequate because the latest and best algorithms are not used and the programs are not carefully tested, certified, and documented. When users find their local library routines cannot be trusted, they start to write their own. But a great deal has been learned in the last decade and the user who writes his own routine is generally not aware of the latest techniques.

A number of recent projects have been aimed at the construction of high quality portable libraries. Examples are the Bell Laboratories Library One project (Gentleman and Traub [9], Traub [23]), the Argonne Laboratories-

Stanford-Texas joint project on the Wilkinson matrix codes, and the IMSL program library [13]. A collection of recent papers on mathematical software may be found in the proceedings of a Purdue conference edited by Rice [21].

REFERENCES

[1] Ashenhurst, R. L., Techniques for Automatic Error Monitoring and Control. Appears in *Error in Digital Computation*, Vol. 1, Edited by L. B. Rall, pp. 43-59, Wiley, 1964.

[2] Borodin, A., Computational Complexity - Theory and Practice. To appear in *Currents in the Theory of Computing*, Edited by A. Aho, Prentice-Hall, 1972.

[3] Brent, R. P., On the Precision Attainable with Various Floating-point Number Systems. IBM Report, 1972.

[4] Cohen, H., Mathematical Applications, Computation, and Complexity. Special Issue of Quarterly of Applied Mathematics on "The Future of Applied Mathematics", April, 1972.

[5] Bright, H. S., A Proposed Numerical Accuracy Control System. Appears in *Interactive Systems for Experimental Applied Mathematics*, edited by M. Klerer and J. Reinfelds, pp. 314-334, Academic Press, 1968.

[6] Businger, P. A., How Old is Numerical Analysis? *Communications of the ACM 14*, 9 (Sept. 1971), 609.

[7] Davis, P. J., Numerical Analysis. Appears in *The Mathematical Sciences*, pp. 128-137, edited by COSRIMS and G. Boehm, MIT Press, 1969.

[8] Gentleman, W. M., Off-the-Shelf Black Boxes for Programming. *IEEE Transactions on Education*, E-12, (1969), pp. 43-50.

[9] Gentleman, W. M., and Traub, J. F., The Bell Laboratories Numerical Mathematics Program Library Project. *Proceedings ACM National Conference*, 1968, pp. 485-490.

[10] Givens, W., Numerical Computation of the Characteristic Values of a Real Symmetric Matrix. *Report ORNL - 1574*, Oak Ridge National Laboratory, 1954.

[11] Householder, A. S., Numerical Analysis. Appears in *Lectures on Modern Analysis*, Volume I, pp. 59-97, edited by T. L. Saaty, Wiley, 1963.

[12] Proceedings of Symposium on Complexity of Computer Computation. To Appear.

[13] IMSL Library I, Mathematical-Statistical Computer Subroutines. International Mathematical and Statistical Libraries, Inc., Houston, Texas, 1971.

[14] Knuth, D. E., *The Art of Computer Programming*. Addison-Wesley, 1968.

[15] Kung, H. T., A Bound on the Multiplication Efficiency of Iteration, Computer Science Department Report, Carnegie-Mellon University, 1972.

[16] Matula, D. W., Significant Digits: Numerical Analysis or Numerology. Proceedings of the IFIP Conference, Ljubljana, (Aug. 1971), 33-37.

[17] Miranker, W. L., A Survey of Parallelism in Numerical Analysis. SIAM Review, Vol. 13, No. 4, (Oct. 1971), 524-547.

[18] Moore, R. E., Interval Arithmetic, Prentice-Hall, 1966.

[19] Naur, P., Revised Report on the Algorithmic Language Algol 60. Communications ACM 6 (1963), pp. 1-17.

[20] Rall, L. B., editor, Error in Digital Computation, Vol. I and II, Wiley, 1965.

[21] Rice, J. R., editor, Mathematical Software. Academic Press, 1971.

[22] Traub, J. F., Computational Complexity of Iterative Processes. Computer Science Department Report, Carnegie-Mellon University, 1971. To appear in SIAM Journal of Computing, 1972.

[23] Traub, J. F., High Quality Portable Numerical Mathematics Program. Appears in Mathematical Software, edited by Rice, J. R., Academic Press, 1971, 131-139.

[24] Traub, J. F., Twenty-fifth Anniversary of Numerical Mathematics. ACM Signum Newsletter 5, 2 (Aug. 1970).

[25] Traub, J. F., Iterative Methods for the Solution of Equations. Prentice-Hall, 1964.

[26] Van Wijngaarden, A., Numerical Analysis as an Independent Science. BIT 6 (1966), pp. 66-81.

[27] Von Neumann, J. and Goldstine, H. H., Numerical Inverting of Matrices of High Order. Bull Amer. Math. Soc. 53, (1947), pp 1021-1099.

[28] Wilkinson, J. H., Rounding Errors in Algebraic Process. Prentice-Hall, 1963.

[29] Wilkinson, J. H., Some Comments From A Numerical Analyst. J. ACM 18, 2 (Apr. 1971), 137-147.

[30] Wilkinson, J. H., The Algebraic Eigenvalue Problem. Oxford University Press, 1965.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Computer Science Department Carnegie-Mellon University Pittsburgh, Pennsylvania 15213	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

NUMERICAL MATHEMATICS AND COMPUTER SCIENCE

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*
J. F. Traub

6. REPORT DATE April, 1972	7a. TOTAL NO. OF PAGES 16	7b. NO. OF REFS 30

8a. CONTRACT OR GRANT NO. N 00014-67-A-0314-0010 b. PROJECT NO. NR 044-422 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
Unlimited distribution.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Mathematics Program Office of Naval Research Arlington, Virginia 22217

13. ABSTRACT