MONASH UNIVERSITY

THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF

Doctor of Science

ON 6 November 1981

B S Barwood

FACULTY SECRETARY

TOPICS IN COMPUTATIONAL COMPLEXITY AND THE ANALYSIS OF ALGORITHMS

A COLLECTION OF PUBLISHED WORKS

SUBMITTED TO THE FACULTY OF SCIENCE

OF MONASH UNIVERSITY

FOR THE DEGREE OF DOCTOR OF SCIENCE

By

Richard P. Brent

October 1980

Acknowledgements

This collection would have been incomplete without the inclusion of three papers published jointly with David Kuck, Kiyoshi Maruyama, Joe Traub, Shmuel Winograd, and Phil Wolfe, whom I thank for their kind permission to republish our joint papers here. Special thanks are due to H. T. Kung for his permission to republish four of our joint papers.

Numerous people, mentioned in the acknowledgements in the individual papers, have contributed in various ways to this collection. I gratefully acknowledge my debt to my teachers at Monash and Stanford Universities, especially P. D. Finch, R. W. Floyd, G. E. Forsythe, G. H. Golub, Z. Janko, D. E. Knuth, J. B. Miller, G. Polya, G. B. Preston, M. Schiffer and E. Strzelecki. Thanks are also due to M. R. Osborne for providing a good research environment during my period in the Computer Centre at the Australian National University, to J. F. Traub for his encouragement and stimulating questions, and to my wife Erin for her patience and understanding over the last twelve years.

Table of Contents

	Page
0. Preliminaries	
0.1 Introduction	7
0.2 Bibliography	15
1. Parallel evaluation of arithmetic expressions	27
1.1 The parallel evaluation of arithmetic expressions without division (by R. P. Brent, D. Kuck & K. Maruyama)	29
1.2 The parallel evaluation of general arithmetic expressions	37
1.3 The parallel evaluation of arithmetic expressions in logarithmic time	45
2. Circuits for arithmetic operations	67
2.1 On the addition of binary numbers	69
2.2 A regular layout for parallel adders (by R. P. Brent & H. T. Kung)	75
2.3 The area-time complexity of binary multiplication (by R. P. Brent & H. T. Kung)	95
3. Continuous models for discrete algorithms	123
3.1 Analysis of the binary Euclidean algorithm	125
3.2 Reducing the retrieval time of scatter storage techniques	163
4. Algorithms for manipulating formal power series	175
4.1 Fast algorithms for manipulating formal power series (by R. P. Brent & H. T. Kung)	177
4.2 Fast algorithms for composition and reversion of multivariate power series (by R. P. Brent & H. T. Kung)	195
4.3 On the complexity of composition and generalized composition of power series (by R. P. Brent & J. F. Traub)	207
5. The complexity of algorithms for solving nonlinear equations	221
5.1 Optimal iterative processes for rootfinding (by R. P. Brent, S. Winograd & P. Wolfe)	223
5.2 A class of optimal-order zero-finding methods using derivative evaluations	241
5.3 The computational complexity of iterative methods for systems of nonlinear equations	259
5.4 Some efficient algorithms for solving systems of nonlinear equations	273
6. Asymptotically fast algorithms for high-precision computations	293
6.1 Multiple-precision zero-finding methods and the complexity of elementary function evaluation	295
6.2 The complexity of multiple-precision arithmetic	323
6.3 Fast multiple-precision evaluation of elementary functions	365

The theme of this collection of papers is the derivation of rigorous bounds on the cost of certain computations. Cost may be measured in several different ways. For example, in [37, 39, 45, 50] we identify cost with the number of arithmetic operations performed on integers or real numbers, in [32, 34] we consider the number of Boolean operations, and in [12, 14, 16] we count function and derivative evaluations. In [3, 15, 18, 22] it is more appropriate to consider the time required to evaluate an expression on a parallel machine, and in [55, 60] we introduce an area-time product which is motivated by practical cost measures for VLSI circuits [106].

Most of the papers are concerned with upper bounds, which are established by exhibiting an algorithm and analysing its performance. In [16, 55] nontrivial lower bounds are established. Proofs of lower bounds are generally more difficult than those for upper bounds, since it is necessary to consider all possible algorithms for the problem at hand, rather than just one carefully selected algorithm. The aim when establishing upper and lower bounds is, of course, to bring them as close together as possible, but this is difficult unless the "trivial" lower bound is almost attainable, as in [3].

The collection is divided into the following six sections, each containing several papers.

1. Parallel evaluation of arithmetic expressions.

2. Circuits for arithmetic operations.

3. Continuous models for discrete algorithms.

4. Algorithms for manipulating formal power series.

5. The complexity of algorithms for solving nonlinear equations.

6. Asymptotically fast algorithms for high-precision computations.

In Sections 1 and 2, the problems considered are discrete and may be solved exactly in a finite number of steps. The two sections are closely related, for the problem of performing arithmetic operations in hardware is essentially equivalent to the problem of evaluating certain Boolean expressions on a parallel machine.

The problems considered in Section 3 (computation of greatest common divisors, information retrieval) are also discrete, but continuous models are used to make the analysis tractable. In Section 4 the underlying problem (computing with formal power series) is in principle infinite, but is made finite by restricting attention to initial segments of power series. For example, we may ask how to compute the first n terms in the reversion of a formal power series, for given n.

In Sections 5 and 6 we consider the computation of approximate solutions to problems whose exact solutions are given by limiting processes and can not, in general, be computed exactly in a finite number of steps. Traub [131] coined the term "analytic computational complexity" to distinguish this area from "discrete" or "algebraic"

computational complexity [67, 76, 139]. Section 5 deals with algorithms for the solution of one or a system of nonlinear equations, and Section 6 with algorithms for the high-precision computation of elementary and special functions. These two sections are closely related, as the algorithms studied in Section 5 may often be used with advantage to solve the computational problems of Section 6.

In the next few pages we attempt to summarise the main contributions and inter-relationships of the papers contained in Sections 1 to 6, as well as briefly mentioning some recent developments.

1. Parallel evaluation of arithmetic expressions

The three papers in this section are concerned with algorithms for the parallel evaluation of arithmetic expressions on a "multiple-instruction, multiple-data" (MIMD) machine [75]. The expressions may be real, integer or Boolean, and it is assumed that the associative, distributive and commutative laws may be used freely to rearrange them into a form suitable for parallel evaluation. Early results of Baer and Bovet [65], Muraoka [113] and others are weak unless the depth of parenthesis nesting is small. Paper 1.1 [15] shows that n-variable expressions involving only addition/subtraction and multiplication can be evaluated in time $2.465\lg(n)$ if enough processors (nonlinear in n) are available. (Here and below we assume that an arithmetic operation can be performed in unit time, and write $\lg(n)$ for $\log_2(n)$. Our usage of the "big O" notation follows the suggestions of Knuth [93].)

Paper 1.2 [22] improves the result of [15] in two ways: expressions involving division are allowed, and the number of operations required is linear in n. Using a "simulation" argument (Lemma 2 of [22]), it follows that any n-variable expression can be evaluated with p processors in time $4.\lg(n) + 10(n-1)/p$, which is within a constant factor (14) of the trivial lower bound $\max(\lg(n), (n-1)/p)$. (A sharper lower bound has been given by Hyafil and Kung [85] for the case of small p.)

Paper 1.3 [18] specialises the result of [22] to the case of arithmetic expressions without division (as considered earlier in [15]), giving a time bound $3.\lg(n) + 5(n-1)/(2p)$, of the same form as the bound in [22], but with smaller and more realistic constants. The constant "3" here is still larger than the constant 2.465 obtained in [15], so the result of [15] is better if p is sufficiently large. The algorithm given in [18] is of interest because it is numerically stable in the sense of backward error analysis [135], and the use of the associative, commutative and distributive laws does not cause any significant amplification of the effect of rounding errors. Unfortunately, this is not the case (or at least has not been established) for most other parallel algorithms [109, 120].

Subsequently, the constants given in [22] were improved by Winograd [138] in the case of small p, and by Muller and Preparata [111] in the case of large p. No improvement which is valid uniformly for all p is known. In a series of papers, Barak, Muller, Preparata and Shamir [66, 117, 118] obtained sharper results than those of [15, 18] for the special case of Boolean expressions and large p. This case is of interest for its

application to combinational circuit design: see Section 2. All these improvements used refinements of the methods introduced in the three papers [15, 18, 22] of Section 1.

Towle and Brent [38] showed that the proofs given in [18, 22] could easily be transformed into efficient procedures for compiling arithmetic expressions for execution on a parallel machine. In fact, the compilation can be performed in time O(n.lg(n)) on a serial machine.

2. Circuits for arithmetic operations

Paper 2.1 [3] considers the time required to add binary numbers, using circuit elements with bounded fan-in. The upper bound obtained is asymptotically equal to Winograd's lower bound [136], and improves by a factor of almost two on the obvious upper bound. A similar result was obtained independently by Krapchenko [94]. Fan-out restrictions are not considered in [3], but in practice these tend to be less severe than fan-in restrictions: see for example [96].

[3] provides a bridge with Section 1, for a corollary of (the proof of) its main result is a good upper bound on the time required for the parallel evaluation of a polynomial. This result predated work by Maruyama, Munro and Paterson [104, 112], and improved earlier results of Dorn, Estrin and Ofman [72, 73, 115]. The corollary follows from the observation (due to Maruyama) that the construction used in [3] is valid for variables over any commutative ring, not just Boolean variables, and that a polynomial in one variable is a special case of the "carry function" considered in [3].

Papers 2.2 [60] and 2.3 [55] use a new computational model, appropriate for modern LSI and VLSI technology [105], in which the chip area is a more realistic measure of cost than the number of gates. There is a trade-off between time and area, and it is possible to give nontrivial upper and lower bounds on the area-time product for certain computations. In [60] we consider the problem of binary addition, while binary multiplication is considered in [55]. From the upper bounds for addition and the lower bounds for multiplication, it follows that multiplication is "harder" than addition, in the sense that it requires a larger area-time product [53].

In practical VLSI designs, as in the model of [55, 60], the cost of communicating results between gates may be more significant than the cost of computing logical functions at the gates. Earlier models [3, 134, 136, 137] ignored communication costs because they were not significant in the days of discrete component technology. Results related to those of [55, 60] have recently been obtained by Abelson, Andreae, Thompson and others [62, 63, 80, 128]. At present there is much interest in this new area of complexity theory.

3. Continuous models for discrete algorithms

The classical Euclidean algorithm for the computation of greatest common divisors (GCDs) is simple to state but difficult to analyse. The main results were conjectured by Gauss [78], but the proofs were not completed until 160 years later [140]. The classical algorithm involves

divisions, but shifts (i.e. multiplications or divisions by powers of two) are faster than divisions on most computers. Consequently, several "binary" Euclidean algorithms, which use shifts instead of divisions, have been proposed [91]. In paper 3.1 [37] we analyse two of these algorithms, and obtain results which are complete for all practical purposes, although some intriguing theoretical questions remain unresolved.

[37] considers only the problem of computing GCDs of (single-precision) integers. See [59, 70, 91] and the references given there for the more difficult problem of computing polynomial GCDs.

Paper 3.2 [13] describes and analyses a scatter storage (i.e. hash coding) method which is more effective than previously known methods if the table is nearly full and keys are (on average) looked up several times. This is often true in practical applications. For a comparison with other methods, see Knuth [92]. The method has been widely used, and has led to further research by Gonnet [81], Mallach [102] and others.

4. Algorithms for manipulating formal power series

A basic problem of symbolic algebraic computation is the manipulation of formal power series in one or more variables. The three papers of Section 4 give asymptotically fast algorithms for the operations of reversion, composition, and iterated composition of dense (as opposed to sparse) power series in a small number of variables. For applications of such algorithms, see [82, 83, 114].

Paper 4.1 [45] shows that the first n terms in the reversion of a power series in one variable can be computed in $O((n.\log(n))^{3/2})$ arithmetic operations in the coefficient domain. (The classical algorithms [91, 114] require order n^3 operations.) Similar results hold for the composition of two power series. In fact, it is shown in [45] that the composition and reversion problems have the same complexity (modulo constant factors). It is an open question whether the exponent 3/2 can be reduced for the general problem. However, at least in many cases of practical interest, the composition problem can be solved in $O(n.\log(n))$ operations [45].

Paper 4.2 [39] outlines how the results of [45] can be extended to power series in several commuting variables. As the algorithms do not take advantage of sparsity (i.e. zero coefficients), they are unlikely to be useful in practice for power series in more than two or three variables.

Paper 4.3 [50] considers the well-known problem of iterated self-composition of a power series [68, 99, 127], and shows that this problem can be solved in time $O((n.\log(n))^{3/2})$, independent of the degree of self-composition. An analogous result for exponentiation of power series had been obtained in [28]. The results of [45] were extended by Kung and Traub [100], and in a different direction by Brent, Gustavson and Yun [59].

The algorithms considered in Section 4 are asymptotically fast, i.e. they are good when n (the number of terms required) is sufficiently large. Empirical estimates of how large n needs to be for the algorithms to be faster than the classical ones have been given by Jones [88]. In this connection, see also [67, 74, 79].

5. The complexity of algorithms for solving nonlinear equations

Section 5 includes four papers on the complexity of iterative methods for the solution of nonlinear equations and systems of equations. Paper 5.1 [16] gives one of the first significant results in the area of analytic computational complexity. Essentially, the result states that a method for the solution of the nonlinear equation $f(x) = 0$, using only evaluations of f and its first d derivatives, can have order of convergence at most $d+2$. This result is the best possible, for order $d+2$ is attained by certain interpolatory methods.

In paper 5.2 [27] we consider certain classes of algorithms which use more evaluations of f' than of f. These algorithms generalise one of Jarratt [87], and improve on other known algorithms using the same information [95, 129]. The algorithms were proved to be optimal by Meersman [107]. Additional details and applications, e.g. to the efficient evaluation of inverse probability distribution functions, are given in [26].

Results for systems of equations [86, 116] are much less satisfactory than those for a single nonlinear equation. Paper 5.3 [12] attempts to extend some of the results of [16] to systems of equations, and paper 5.4 [14] describes several classes of algorithms which are both practically useful [103, 110] and theoretically interesting. Other practical methods are given in [30, 44]. Traub, Wozniakowski and their students have done much further work on questions of optimality of iterative methods for nonlinear equations and systems of equations [89, 133, 141, 142].

6. Asymptotically fast algorithms for high-precision computations

The final section contains three papers on the complexity of high-precision computation of arithmetic operations (division, square root etc.) and elementary functions (log, exp, sin, cos etc.). Paper 6.1 [28] considers the complexity of high-precision zero-finding methods, and thus provides a bridge between Sections 5 and 6. Some power series algorithms are formally similar to multiple-precision algorithms (except for the lack of carries), so there is some overlap with the material of Section 4. [28] also includes slightly faster alternatives to the algorithms described in [34].

Paper 6.2 [32] analyses in detail the complexity of the basic high-precision arithmetic operations. It also includes some practical (though not asymptotically fastest) high-precision algorithms for elementary functions, and further analysis of high-precision zero-finding methods. The algorithm suggested for computation of the exponential function was later analysed in great detail by Clenshaw and Olver [69].

Paper 6.3 [34] gives the best known asymptotic bounds on the time required for high-precision evaluation of log(x), exp(x), sin(x) etc. (The constant factors may be improved: see [28].) To obtain n-bit accuracy requires only O(log(n)) multiplications of n-bit numbers, and each of these can be performed with O(n.log(n)log(log(n))) bit-operations if the Schönhage-Strassen algorithm [124] is used. Similar results were obtained independently by Gosper, Salamin and Schroeppel, but apparently were never published.

The "Gauss-Legendre" algorithm for fast high-precision computation of π was first published in [28, 34], although discovered independently by Salamin [119]. Its name comes from the fact that it depends on identities known to Gauss [77] and Legendre [101], but it was not discovered by them, probably because its usefulness depends on the availability of a fast multiplication algorithm such as the Karatsuba-Ofman or Schönhage-Strassen algorithm [64, 67, 90, 91, 124]. The algorithms considered in [34] are theoretically interesting because they are asymptotically the fastest known (modulo constant factors), and are within a factor O(log(n)) of the lower bounds [32]. There are certainly more practical algorithms for low and moderate precision computation, see e.g. [32, 35, 42, 52, 121, 125].

This concludes our brief introduction. In the space available it has not been possible to present a complete survey. The reader is referred to the books and papers listed in the Bibliography below, and particularly [64, 67, 84, 122, 132, 133, 139], for a broader coverage of the field. Some of the papers included in this collection also contain a substantial amount of introductory or survey material.

Bibliography

Notes

[1] to [61] are publications of R. P. Brent (not all cited above), [62] to [142] are other publications cited above or containing relevant background material. The following abbreviations are used:

ACM:	Association for Computing Machinery.
ANU:	Australian National University, Canberra.
CMU:	Carnegie-Mellon University, Pittsburgh, Pennsylvania.
CR:	Computing Reviews.
DCS:	Department of Computer Science.
IBM Research:	IBM Thomas J. Watson Research Center, Yorktown Heights, New York.
IEEE:	Institute of Electrical and Electronics Engineers.
MR:	Mathematical Reviews.
NTIS:	National Technical Information Service (USA).
SIAM:	Society for Industrial and Applied Mathematics.
Stanford:	Stanford University, Stanford, California.
TR:	Technical Report.
*:	Appears in this collection.

1. M. P. C. Legg & R. P. Brent, Automatic contouring, Proceedings of the Fourth Australian Computer Conference, Australian Computer Society, Adelaide, 1969, 467-468. CR 12#21982.

2. R. P. Brent, Algorithms for matrix multiplication, TR CS 157, DCS, Stanford (March 1970), 52pp. (Available from NTIS, #AD705509.)

*3. R. P. Brent, On the addition of binary numbers, IEEE Trans. Comp. C-19 (1970), 758-759. CR 12#20898. [See Section 2.1]

4. R. P. Brent, Error analysis of algorithms for matrix multiplication and triangular decomposition using Winograd's identity, Numer. Math. 16 (1970), 145-156. MR 43#5702, CR 12#21408.

5. R. P. Brent, An algorithm with guaranteed convergence for finding a zero of a function, Comp. J. 14 (1971), 422-425. MR 49#4234.

6. R. P. Brent, Algorithms for Minimization without Derivatives, Prentice-Hall, Englewood Cliffs, New Jersey, 1973, 195pp. Errata: Math. Comp. TE 520, 29 (1975), 1166. Reviewed in: American Scientist 61 (May-June 1973), 374; Math. Programming 4 (1973), 349; Comp. J. 16 (1973), 314; Math. Comp. 28 (1974), 865-866; CR 15#26544; MR 49#4251, 51#7283. Preliminary version appeared as "Algorithms for finding zeros and extrema of functions without calculating derivatives", TR CS 198, DCS, Stanford (Feb. 1971), 313pp. (Ph. D. thesis, available from NTIS, #AD726170.)

7. R. P. Brent, A new algorithm for minimizing a function of several variables without calculating derivatives, in "Optimization" (edited by R. S. Anderssen, L. S. Jennings & D. M. Ryan), University of Queensland Press, Brisbane, 1972, 14-25. MR 52#2574, 52#9601.

8. R. P. Brent, On the Davidenko-Branin method for solving simultaneous nonlinear equations, IBM J. Res. and Dev. 16 (1972), 434-436. CR 14#24419, MR 48#12817. Also appeared as "A note on the Davidenko-Branin method for solving simultaneous nonlinear equations", TR RC 3506, IBM Research (Aug. 1971), 7pp.

9. R. P. Brent, A modified linear scatter storage technique, IBM Tech. Disclosure Bull. 14, 11 (1972), 3489.

10. R. P. Brent, An optimal secant method for solving systems of nonlinear equations, IBM Tech. Disclosure Bull. 15, 4 (1972), 1216.

11. R. P. Brent, An optimal orthogonal triangularization method for solving systems of nonlinear equations, IBM Tech. Disclosure Bull. 15, 4 (1972), 1217.

*12. R. P. Brent, The computational complexity of iterative methods for systems of nonlinear equations, in [108], 61-71. MR 51#9575, 52#4703. [See Section 5.3]

*13. R. P. Brent, Reducing the retrieval time of scatter storage techniques, Comm. ACM 16 (1973), 105-109. Also appeared as "A modified linear scatter storage technique", TR RC 3460, IBM Research

(July 1971), 20pp. See also "Comment on Brent's scatter storage algorithm", Comm. ACM 16 (1973), 703. [See Section 3.2]

*14. R. P. Brent, Some efficient algorithms for solving systems of nonlinear equations, SIAM J. Numer. Anal. 10 (1973), 327-344 (G. E. Forsythe memorial issue). MR 48#10096, CR 17#29965. Preliminary version appeared as "On maximizing the efficiency of algorithms for solving systems of nonlinear equations", TR RC 3725, IBM Research (Feb. 1972), 33pp. [See Section 5.4]

*15. R. P. Brent, D. Kuck & K. Maruyama, The parallel evaluation of arithmetic expressions without division, IEEE Trans. Comp. C-22 (1973), 532-534. MR 50#11843. [See Section 1.1]

*16. R. P. Brent, S. Winograd & P. Wolfe, Optimal iterative processes for rootfinding, Numer. Math. 20 (1973), 327-341. CR 15#26753, MR 47#6079. Preliminary version appeared as TR RC 3960, IBM Research (Aug. 1972), 29pp. [See Section 5.1]

17. R. P. Brent, On the precision attainable with various floating-point number systems, IEEE Trans. Comp. C-22 (1973), 601-607. CR 14#25960. Also appeared as TR RC 3751, IBM Research (Feb. 1972), 28pp.

*18. R. P. Brent, The parallel evaluation of arithmetic expressions in logarithmic time, in [130], 83-102. CR 15#26540, 15#27335, MR 50#15432. [See Section 1.3]

19. R. P. Brent, The first occurrence of large gaps between successive primes, Math. Comp. 27 (1973), 959-963. MR 48#8360.

20. R. P. Brent, Sources of error in computation, in "Error, Approximation and Accuracy" (edited by F. de Hoog & C. L. Jarvis), University of Queensland Press, Brisbane, 1973, 122-128. MR 52#2124, 54#9069.

21. R. P. Brent, The distribution of small gaps between successive primes, Math. Comp. 28 (1974), 315-324. MR 48#8356. Preliminary version appeared as "Empirical evidence for a proposed distribution of small prime gaps", TR CS 127, DCS, Stanford (Feb. 1969), 18pp.

*22. R. P. Brent, The parallel evaluation of general arithmetic expressions, J. ACM 21 (1974), 201-206. CR 15#27055. [See Section 1.2]

23. R. P. Brent, Algorithm 488: A Gaussian pseudo-random number generator (G5), Comm. ACM 17 (1974), 704-706.

24. R. P. Brent, Irregularities in the distribution of primes and twin primes, Math. Comp. 29 (1975), 43-56 (D. H. Lehmer special issue). Errata: ibid 30 (1976), 148. MR 51#5522, 53#302. See also "Tables concerning irregularities in the distribution of primes and twin primes", UMT 4, Math. Comp. 29 (1975), 331; and "Tables concerning irregularities in the distribution of primes and twin

primes to 10^{11} ", UMT 21, Math. Comp. 30 (1976), 379.

25. R. P. Brent, Numerical solution of nonlinear equations, DCS, Stanford (March 1975), 189pp.

26. R. P. Brent, Some high-order zero-finding methods using almost orthogonal polynomials, J. Austral. Math. Soc. (Ser. B) 19 (1975), 1-29. MR 52#16000. Also appeared as "Efficient methods for finding zeros of functions whose derivatives are easy to evaluate", TR, DCS, CMU (Dec. 1974), 62pp.

*27. R. P. Brent, A class of optimal-order zero-finding methods using derivative evaluations, in [131], 59-73. MR 52#15938, 54#9073. Reviewed in Austral. Comp. J. 9 (1977), 100. Also appeared as a TR, DCS, CMU (June 1975), 15pp. [See Section 5.2]

*28. R. P. Brent, Multiple-precision zero-finding methods and the complexity of elementary function evaluation, in [131], 151-176. MR 52#15938, 54#11843. Reviewed in Austral. Comp. J. 9 (1977), 100. Also appeared as a TR, DCS, CMU (July 1975), 26pp. [See Section 6.1]

29. R. P. Brent & H. T. Kung, $O((n \log n)^{3/2})$ algorithms for composition and reversion of power series, in [131], 217-225. MR 52#15938, 55#1699. Reviewed in Austral. Comp. J. 9 (1977), 100. Also appeared as a TR, DCS, CMU (May 1975), 7pp.

30. J. P. Abbott & R. P. Brent, Fast local convergence with single and multistep methods for nonlinear equations, J. Austral. Math. Soc. (Ser. B) 19 (1975), 173-199. Errata: ibid 20 (1977), 254. MR 55#4677, 58#13673.

31. R. S. Anderssen & R. P. Brent (editors), The Complexity of Computational Problem Solving, University of Queensland Press, Brisbane, 1976, 262pp. LC 76-374278, ISBN 0-7022-1213-X. Reviewed in Comp. J. 21 (1978), 242.

*32. R. P. Brent, The complexity of multiple-precision arithmetic, in [31], 126-165. [See Section 6.2]

33. R. S. Anderssen, R. P. Brent, D. J. Daley & P. A. P. Moran, Concerning $\int_0^1 \cdots \int_0^1 (x_1^2 + \cdots + x_k^2)^{1/2} dx_1 \cdots dx_k$ and a Taylor series method, SIAM J. Appl. Math. 30 (1976), 22-30. MR 52#15773.

*34. R. P. Brent, Fast multiple-precision evaluation of elementary functions, J. ACM 23 (1976), 242-251. MR 52#16111. Also appeared as TR STAN-CS-75-515, DCS, Stanford (Aug. 1975), 22pp. [See Section 6.3]

35. R. P. Brent, MP users guide, TR 54, Computer Centre, ANU (Sept. 1976), 53pp. Revisions published as "MP users guide (second edition)", Computing Research Group, ANU (Aug. 1978), 44pp.; and "MP user's guide (third edition)", TR-CS-79-08, DCS, ANU (Dec. 1979), 73pp.

36. R. P. Brent, Knuth's constants to 1000 decimal and 1100 octal places, TR 47, Computer Centre, ANU (Sept. 1975), 25pp. See also UMT 30, Math. Comp. 30 (1976), 668.

*37. R. P. Brent, Analysis of the binary Euclidean algorithm, in [132], 321-355. MR 54#14417, 55#11701. Reviewed in Austral. Comp. J. 10 (1978), 76-77. Also appeared as a TR, DCS, CMU (June 1976), 35pp. Extended abstract appeared in SIGSAM Bulletin (May 1976). [See Section 3.1]

38. R. Towle & R. P. Brent, On the time required to parse an arithmetic statement for parallel processing, in "Proceedings of the 1976 International Conference on Parallel Processing" (edited by P. H. Enslow), IEEE, New York, 1976, 254. (IEEE Catalog #76CH1127-0C.)

*39. R. P. Brent & H. T. Kung, Fast algorithms for composition and reversion of multivariate power series, in "Proceedings of a Conference on Theoretical Computer Science held at the University of Waterloo", DCS, University of Waterloo, Ontario (Aug. 1977), 149-158. [See Section 4.2]

40. R. P. Brent, Computation of the regular continued fraction for Euler's constant, Math. Comp. 31 (1977), 771-777. MR 55#9490. See also "γ and $\exp(\gamma)$ to 20700D and their regular continued fractions to 20000 partial quotients", UMT 1, Math. Comp. 32 (1978), 311.

41. A. H. Sameh & R. P. Brent, Solving triangular systems on a parallel computer, SIAM J. Numer. Anal. 14 (1977), 1101-1113. MR 56#17026. Also appeared as TR UIUCDCS-R-75-766, DCS, University of Illinois, Urbana, Illinois (Nov. 1975), 18pp.

42. R. P. Brent, A Fortran multiple-precision arithmetic package, ACM Trans. Math. Software 4 (1978), 57-70. CR 20#34962. Also appeared as a TR, DCS, CMU (May 1976), 29pp.

43. R. P. Brent, Algorithm 524: MP, a Fortran multiple-precision arithmetic package, ACM Trans. Math. Software 4 (1978), 71-81. See also "Remark on Algorithm 524", ibid 5 (1979), 578-579.

44. J. P. Abbott & R. P. Brent, A note on continuation methods for the solution of nonlinear equations, J. Austral. Math. Soc. (Ser. B) 20 (1978), 157-164. MR 58#13672.

*45. R. P. Brent & H. T. Kung, Fast algorithms for manipulating formal power series, J. ACM 25 (1978), 581-595. CR 20#34535, MR 58#25090. Also appeared as a TR, DCS, CMU (Jan. 1976), 38pp. [See Section 4.1]

46. R. P. Brent, T. Dwyer, S. Edwards, A. Glenn, D. Hawking, A. J. Hurst, C. Johnson, N. Justusson, T. Kelly, B. Molinari, D. Poole, J. M. Robson & I. R. Simpson, Comments on the draft Pascal standard, Austral. Computer Science Communications 1 (1979), 310-317. Also appeared as TR-CS-79-09, DCS, ANU (Oct. 1979), 9pp.

47. R. P. Brent, On the zeros of the Riemann zeta function in the critical strip, Math. Comp. 33 (1979), 1361-1372. A preliminary version appeared as "Computation of the zeros of the Riemann zeta function in the critical strip", TR CMU-CS-78-148, DCS, CMU (Nov. 1978), 27pp. A progress report appeared as "The first 40,000,000 zeros of the Riemann zeta function lie on the critical line", AMS Notices 24 (1977), A-417.

48. R. P. Brent & R. A. Jarvis (editors), Proceedings of the Third Australian Computer Science Conference, special issue of Austral. Comp. Sci. Communications 2, 1 (Jan. 1980), 222 pp.

49. R. P. Brent & E. M. Mc Millan, Some new algorithms for high-precision computation of Euler's constant, Math. Comp. 34 (1980), 305-312. Also appeared as TR LBL-8729, Lawrence Berkeley Laboratory; and as TR-CS-79-03, DCS, ANU (Jan. 1979), 16pp. See also "Euler's constant and its exponential to 30,100 decimals", and "The first 29,000 partial quotients in the regular continued fraction for Euler's constant and its exponential", both submitted to Math. Comp. UMT File.

*50. R. P. Brent & J. F. Traub, On the complexity of composition and generalized composition of power series, SIAM J. Computing 9 (1980), 54-66. Also appeared as TR CMU-CS-78-128, DCS, CMU (May 1978), 34pp. Abstract appeared in SIGSAM Bulletin 12, 2 (May 1978), 9. [See Section 4.3]

51. R. P. Brent, An improved Monte Carlo factorization algorithm, BIT 20 (1980), to appear. Appeared as "Some new cycle finding and factorization algorithms", TR-CS-79-11, DCS, ANU (Nov. 1979), 10pp.

52. R. P. Brent, Unrestricted algorithms for elementary and special functions, Information Processing 80, North-Holland, Amsterdam, 1980, to appear. Appeared as TR-CS-79-13, DCS, ANU (Nov. 1979), 20pp.

53. R. P. Brent & H. T. Kung, Chip complexity of binary arithmetic, Proc. Twelfth Annual ACM Symposium on the Theory of Computing, ACM, New York, 1980, 190-200.

54. R. P. Brent, J. A. Hooper & J. M. Yohe, An Augment interface for Brent's multiple-precision arithmetic package, ACM Trans. Math. Software 6 (1980), 146-149. A longer version appeared as Technical Summary Report #1868, Mathematics Research Center, University of Wisconsin, Madison (Aug. 1978), 26pp.

*55. R. P. Brent and H. T. Kung, The area-time complexity of binary multiplication, TR-CS-79-05, DCS, ANU; and TR CMU-CS-79-136, DCS, CMU (July 1979), 25pp. (To appear in J. ACM.) [See Section 2.3]

56. R. P. Brent & H. T. Kung, On the area of binary tree layouts, Information Processing Letters, to appear. Appeared as TR-CS-79-07, DCS, ANU (July 1979), 5pp.

57. R. P. Brent, The first occurrence of certain large prime gaps, Math. Comp. 34 (1980), to appear.

58. G. M. Baudet, R. P. Brent & H. T. Kung, Parallel execution of a sequence of tasks on an asynchronous multiprocessor, Austral. Comp. J. 12 (1980), to appear. A preliminary version appeared as a TR, DCS, CMU (June 1977), 28pp.

59. R. P. Brent, F. G. Gustavson & D. Y. Y. Yun, Fast solution of Toeplitz systems of equations and computation of Pade approximants, J. Algorithms 1 (1980), to appear. Appeared as TR RC 8173, IBM Research (Jan. 1980); and as TR-CS-79-06, DCS, ANU (June 1980), 36pp.

*60. R. P. Brent & H. T. Kung, A regular layout for parallel adders, TR-CS-79-04, DCS, ANU; and TR CMU-CS-79-131, DCS, CMU (June 1979), 16pp. [See Section 2.2]

61. R. P. Brent, Factorization of the eighth Fermat number, AMS Notices 27 (1980), to appear.

62. H. Abelson, Lower bounds on information transfer in distributed computations, J. ACM 27 (1980), 384-392.

63. H. Abelson & P. Andreae, Information transfer and area-time tradeoffs for VLSI multiplication, Comm. ACM 23 (1980), 20-23.

64. A. V. Aho, J. E. Hopcroft & J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, 1976.

65. J. L. Baer & D. P. Bovet, Compilation of arithmetic expressions for parallel computations, Information Processing 68, North-Holland, Amsterdam, 1968, 340-346.

66. A. B. Barak & E. Shamir, On the parallel evaluation of Boolean expressions, SIAM J. Computing 5 (1976), 678-681.

67. A. Borodin & I. Munro, The Computational Complexity of Algebraic and Numeric Problems, American Elsevier, New York, 1975.

68. N. G. de Bruijn, Asymptotic Methods in Analysis, third edition, North-Holland, Amsterdam, 1970.

69. C. W. Clenshaw & F. W. J. Olver, An unrestricted algorithm for the exponential function, SIAM J. Numer. Anal. 17 (1980), 310-331.

70. G. E. Collins, The computing time of the Euclidean algorithm, SIAM J. Computing 3 (1974), 1-10.

71. J. W. Cooley & J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comp. 19 (1965), 297-301.

72. W. J. Dorn, Generalizations of Horner's rule for polynomial evaluation, IBM J. Res. and Dev. 4 (1962), 239-245.

73. G. Estrin, Organization of computer systems - the fixed plus variable computer structure, Proc. Fifth Western Joint Computer Conference, Spartan Books, New York, 1960, 33-44.

74. R. J. Fateman, Polynomial multiplication, powers, and asymptotic analysis: some comments, SIAM J. Computing 3 (1974), 196-213.

75. M. J. Flynn, Some computer organizations and their effectiveness, IEEE Trans. Comp. C-21 (1972), 948-960.

76. M. R. Garey & D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co., San Francisco, 1979.

77. C. F. Gauss, Carl Friedrich Gauss Werke, Bd. 3, Gottingen, 1876, 362-403.

78. C. F. Gauss, ibid, Bd. X_1, 371-374.

79. W. M. Gentleman, On the relevance of various cost models of complexity, in [130], 103-109.

80. W. M. Gentleman, Some complexity results for matrix computations on parallel processors, J. ACM 25 (1978), 112-115.

81. G. H. Gonnet & J. I. Munro, Efficient ordering of hash tables, SIAM J. Computing 8 (1979), 463-478.

82. P. Henrici, Applied and Computational Complex Analysis, Vol. 1, Wiley-Interscience, New York, 1974.

83. P. Henrici, Fast Fourier methods in computational complex analysis, SIAM Review 21 (1979), 481-527.

84. J. E. Hopcroft, Complexity of computer computations, Information Processing 74, North-Holland, Amsterdam, 1974, 620-626.

85. L. Hyafil & H. T. Kung, The complexity of parallel evaluation of linear recurrences, J. ACM 24 (1977), 513-521.

86. J. Jankowska, Theory of multivariate secant methods, SIAM J. Numer. Anal. 16 (1979), 547-562.

87. P. Jarratt, Some efficient fourth order multipoint methods for solving equations, BIT 9 (1969), 111-124.

88. H. P. G. Jones, The implementation and comparison of 'fast' power series algorithms, DCS, ANU, Nov. 1979 (Honours thesis).

89. B. Kacewicz, An integral-interpolatory iterative method for the solution of nonlinear scalar equations, Numer. Math. 26 (1976), 355-365.

90. A. Karatsuba & Yu. Ofman, Multiplication of multidigit numbers on automata, Soviet Physics Dokl. 7 (1963), 595-596.

91. D. E. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, Addison-Wesley, Menlo Park, 1969.

92. D. E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, Menlo Park, 1973.

93. D. E. Knuth, Big Omicron and Big Omega and Big Theta, SIGACT News 8, 2 (1976), 18-24.

94. V. M. Krapchenko, Asymptotic estimation of addition time of a parallel adder, Syst. Theory Res. 19 (1970), 105-122.

95. L. I. Kronsjo, Algorithms: Their Complexity and Efficiency, John Wiley and Sons, New York, 1979.

96. D. J. Kuck, The Structure of Computers and Computations, Vol. 1, John Wiley and Sons, New York, 1978.

97. D. J. Kuck, D. H. Lawrie & A. H. Sameh (editors), High Speed Computer and Algorithm Organization, Academic Press, New York, 1977.

98. D. J. Kuck & K. Maruyama, Time bounds on the parallel evaluation of arithmetic expressions, SIAM J. Computing 4 (1975), 147-162.

99. M. Kuczma, Functional Equations in a Single Variable, PWN - Polish Scientific Publishers, Warsaw, 1968.

100. H. T. Kung & J. F. Traub, All algebraic functions can be computed fast, J. ACM 25 (1978), 245-260.

101. A. M. Legendre, Exercices de Calcul Integral, Vol. 1, Paris, 1811, pg. 61.

102. E. G. Mallach, Scatter storage techniques: a unifying viewpoint and a method for reducing retrieval times, Comp. J. 20 (1977), 137-140.

103. J. M. Martinez, Generalizations of the methods of Brent and Brown for solving nonlinear simultaneous equations, SIAM J. Numer. Anal. 16 (1979), 434-448.

104. K. Maruyama, On the parallel evaluation of polynomials, IEEE Trans. Comp. C-22 (1973), 2-5.

105. C. A. Mead & L. A. Conway, Introduction to VLSI Systems, Addison-Wesley, Menlo Park, 1979.

106. C. A. Mead & M. Rem, Cost and performance of VLSI computing structures, IEEE J. Solid State Circuits SC-14 (1979), 455-462.

107. R. Meersman, Optimal use of information in certain iterative processes, in [131], 109-125.

108. R. E. Miller & J. W. Thatcher (editors), Complexity of Computer Computations, Plenum Press, New York, 1972.

109. W. Miller, Computational complexity and numerical stability, SIAM J. Computing 4 (1975), 97-107.

110. J. J. More & M. Y. Cosnard, Numerical solution of nonlinear equations, ACM Trans. Math. Software 5 (1979), 64-85.

111. D. E. Muller & F. P. Preparata, Restructuring of arithmetic expressions for parallel evaluation, J. ACM 23 (1976), 534-543.

112. I. Munro & M. Paterson, Optimal algorithms for parallel polynomial evaluation, J. Comp. Systems Sci. 7 (1973), 189-198.

113. Y. Muraoka, Parallelism exposure and exploitation in programs, Report 424, Digital Computing Lab., Univ. of Illinois, Urbana, Illinois, 1971 (Ph. D. thesis).

114. A. C. Norman, Computing with formal power series, ACM Trans. Math. software 1 (1975), 346-356.

115. Yu. Ofman, On the algorithmic complexity of discrete functions, Dokl. Akad. Nauk SSSR 145 (1962), 48-51 (in Russian).

116. J. M. Ortega & W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

117. F. P. Preparata & D. E. Muller, The time required to evaluate division-free arithmetic expressions, Information Proc. Letters 3 (1975), 144-146.

118. F. P. Preparata, D. E. Muller & A. B. Barak, Reduction of depth of Boolean networks with a fan-in constraint, IEEE Trans. Comp. C-26 (1977), 474-479.

119. E. Salamin, Computation of π using arithmetic-geometric mean, Math. Comp. 30 (1976), 565-570.

120. A. H. Sameh, Numerical parallel algorithms - a survey, in [97], 207-228.

121. T. Sasaki & Y. Kanada, Practically fast multiple-precision evaluation of log(x), Institute of Physical and Chemical Research, Wako-shi, Saitama 351, Japan, 1980 (preprint).

122. J. E. Savage, The Complexity of Computing, John Wiley and Sons, New York, 1976.

123. A. Schönhage, Partial and total matrix multiplication, Mathematisches Institut der Universitat Tubingen, 1979 (preprint).

124. A. Schönhage & V. Strassen, Schnelle Multiplikation grosser Zahlen, Computing 7 (1971), 281-292.

125. D. Shanks & J. W. Wrench, Calculation of π to 100,000 decimals, Math. Comp. 16 (1962), 76-99.

126. V. Strassen, Gaussian elimination is not optimal, Numer. Math. 13 (1969), 354-356.

127. G. Szekeres, Functional iteration of entire and rational functions, J. Austral. Math. Soc. 4 (1964), 129-142.

128. C. D. Thompson, Area-time complexity for VLSI, Proc. 11th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1979, 81-88.

129. J. F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

130. J. F. Traub (editor), Complexity of Sequential and Parallel Numerical Algorithms, Academic Press, New York, 1973.

131. J. F. Traub (editor), Analytic Computational Complexity, Academic Press, New York, 1975.

132. J. F. Traub (editor), New Directions and Recent Results in Algorithms and Complexity, Academic Press, New York, 1976.

133. J. F. Traub & H. Wozniakowski, General Theory of Optimal Error Algorithms and Analytic Complexity, Academic Press, New York, 1980.

134. C. S. Wallace, A suggestion for a fast multiplier, IEEE Trans. Elec. Comp. EC-13 (1964), 14-17.

135. J. H. Wilkinson, Rounding Errors in Algebraic Processes, HMSO, London, 1963.

136. S. Winograd, On the time required to perform addition, J. ACM 12 (1965), 277-285.

137. S. Winograd, On the time required to perform multiplication, J. ACM 14 (1967), 793-802.

138. S. Winograd, On the parallel evaluation of certain arithmetic expressions, J. ACM 22 (1975), 477-492.

139. S. Winograd, Arithmetic Complexity of Computations, SIAM, Philadelphia, 1980.

140. E. Wirsing, On the theorem of Gauss-Kusmin-Levy and a Frobenius-type theorem for function spaces, Acta Arith. 24 (1974), 507-528.

141. H. Wozniakowski, Maximal stationary iterative methods for the solution of operator equations, SIAM J. Numer. Anal. 11 (1974), 934-949.

142. H. Wozniakowski, Generalized information and maximal order of iteration for operator equations, SIAM J. Numer. Anal. 12 (1975), 121-135.