RÉSUMÉ OF RESEARCH

RICHARD P. BRENT
APRIL 1997

Introduction. Brent's work is well-known and widely quoted[1] in several areas of Computer Science and Mathematics. This résumé outlines his contributions to parallel algorithms, area-time bounds for VLSI, number-theoretic and high-precision computations, computational complexity, analysis of algorithms, algorithms for Toeplitz systems, optimisation, and the solution of nonlinear algebraic systems. Brent's work has been influential in the development of several of these fields.

Parallel Algorithms. Parallel computation is now one of the most active areas of Computer Science. Brent is a pioneer in the development and analysis of parallel algorithms, i.e. algorithms suitable for parallel computers.

Brent's result [22][2] on parallel evaluation of arithmetic expressions shows that arithmetic expressions in n distinct variables can be evaluated in $O(\log n)$ parallel steps using $O(n/\log n)$ processors. The result is the best possible, up to small constant factors. It should be compared with the well-known result of Valiant *et al*[3] that any multivariate polynomial of degree d which can be evaluated sequentially in n steps can be evaluated in $O(\log d)(\log n + \log d)$ parallel steps using $O(n^3 d^6)$ processors. The latter result is more general, but weaker, because of the $\log d$ terms in the time bound and the much larger number of processors. Both results apply to Boolean expressions, so they have implications for the design of circuits with small depth, i.e. small delay. Brent's result improved on earlier joint work with Kuck and Maruyama [15][4] and various results for restricted classes of expressions[5]. It inspired

[1] There are about 1,725 references to Brent's work in *Science Citation Index* up to 1996, including 1,155 for the period 1985-1996. His work is implicitly referred to several times in *Computer Science: Achievements and Opportunities* (Report of the NSF Advisory Committee for Computer Research), SIAM, Philadelphia, 1989.

[2] References to Brent's publications use the same numbering as his c.v., available from `http://cslab.anu.edu.au/~rpb/` . References to recent work and other authors are given in footnotes such as this.

[3] L. G. Valiant, S. Skyum, S. Berkowitz and C. Rackoff, "Fast parallel computation of polynomials using few processors", *SIAM J. Computing* 12, 1983, 641-644.

[4] [15] did not allow division, and required $O(n^2)$ processors.

[5] For example: I. Munro and M. Paterson, "Optimal algorithm for parallel polynomial evaluation", *Proc. IEEE Twelfth Annual Symp. on Switching and Automata Theory*, 1971, 132-139.

 résumé typeset using $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX further work by Winograd[6], Muller and Preparata[7], Miller and Reif[8] and others.

The *systolic array*[9] is a particularly simple form of parallel computer. In joint work with H. T. Kung and F. T. Luk, Brent has obtained many important results for computations on systolic arrays[10]. In [73,77], he showed how integer and polynomial greatest common divisor (GCD) problems could be evaluated in linear time on a linear systolic array. The polynomial GCD problem has applications to decoding of Reed-Solomon and other error-detecting codes[11]. The integer GCD problem is of interest because it is not known whether it is in the class NC[12]. In fact, the fastest known parallel algorithm for the integer GCD problem is only slightly faster than the Brent-Kung algorithm, although it uses many more processors[13].

In joint work with Luk, Brent [84] pioneered the use of Jacobi-like algorithms for the singular value decomposition (SVD) and symmetric eigenvalue problem on systolic arrays[14]. These algorithms are competitive with more traditional algorithms[15] on local-memory computers

---

[6]S. Winograd, "On the parallel evaluation of certain arithmetic expressions", *J. ACM* 22, 1975, 477-492.

[7]D. E. Muller and F. P. Preparata, "Restructuring of arithmetic expressions for parallel evaluation", *J. ACM* 23, 1976, 534-543; also F. P. Preparata, D. E. Muller and A. B. Barak, "Reduction of depth of Boolean networks with a fan-in constraint", *IEEE Transactions on Computers* C-26, 1977, 474.

[8]G. L. Miller and J. H. Reif, "Parallel tree contraction, Part I: Fundamentals", in *Randomness and Computation* (edited by S. Micali), JAI Press, 1988; G. L. Miller, V. Ramachandran and E. Kaltofen, "Efficient parallel evaluation of straight-line code and arithmetic circuits", *SIAM J. Computing* 17, 1988, 687-695; G. Miller and S.-H. Teng, "Dynamic parallel complexity of computational circuits", *Proc. 19th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1987, 254-263; *etc.*

[9]Closely related to Von Neumann's *cellular automata*. See H. T. Kung, "Why systolic architectures ?", *IEEE Computer* 15, Jan. 1982, 37-45.

[10]The results for systolic arrays are also applicable to more general models of parallel computation.

[11]I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields", *J. SIAM* 8, 1960, 300-304; see also R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Mass., 1983. See also [109,112,114,124].

[12]NC is the class of problems which can be solved in poly-log time using a polynomial number of processors. The basic arithmetic operations are in NC. See S. A. Cook, "An overview of computational complexity", *Comm. ACM* 26, 1983, 401-408.

[13]A. Borodin, J. von zur Gathen and J. Hopcroft, "Parallel matrix and GCD computations", *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science*, IEEE, New York, 1982, 65-71.

[14]See also [77,82,96] for GCDs, and [80,83,87] for the symmetric eigenvalue problem, the SVD and various generalisations.

[15]For example, reduction to tridiagonal or Hessenberg form followed by the QR algorithm, or the "divide and conquer" algorithm of Dongarra and Sorensen (*SIAM J. Scientific and Statistical Computing* 8, 1987, 139-154).

and have good numerical properties[16]. Brent and Luk [78,79] also gave practical linear-time algorithms for the solution of Toeplitz systems on linear systolic arrays. The results of Brent and Luk have many applications in digital signal processing[17].

Brent's recent work on parallel algorithms includes the development of practical algorithms for linear algebra [136] and sorting [158] on distributed-memory MIMD computers.

VLSI Design and Area-Time Bounds. Early results[18] on the complexity of computational circuits counted *gates* and *gate delays* but ignored *wires* and *propagation delays*. With the introduction of "very large scale integration" (VLSI), a more realistic model which took wires into account was required. Brent and Kung introduced such a model in an influential paper [55] which established a lower bound $AT = \Omega(n^{3/2})$ on the product of the area A and time T required for the fundamental problem of n-bit binary multiplication[19]. The technique was extended to decision problems in [64]. The results give nontrivial examples of a common phenomenon: there is usually a tradeoff between the area A and time T required for a computation on a VLSI chip[20].

Binary trees are fundamental to many algorithms. Brent and Kung [56] used geometric arguments to obtain a lower bound on the area of binary tree layouts, with significant practical implications for VLSI design.

Parallel prefix computation is a fundamental technique in parallel processing[21]. Brent and Kung [60] showed how to apply it to the design of VLSI carry-lookahead adders. At the time the use of carry-lookahead

---

[16] J. Demmel and K. Veselić, "Jacobi's method is more accurate than QR", *SIAM J. Sci. Stat. Computing* 11, 1992, 1204-1246.

[17] G. H. Golub and P. Van Dooren, *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, Springer-Verlag, 1990, 93-110.

[18] For example: Yu. Ofman, "On the algorithmic complexity of discrete functions", *Dokl. Akad. Nauk SSSR* 145, 1962, 48-51 (translation: *Sov. Phys. Dokl.* 7, 1968, 589-591); S. Winograd, "On the time required to perform multiplication", *J. ACM* 14, 1967, 793-802; J. E. Savage, *The Complexity of Computing*, Wiley, New York, 1976, Chs. 2-3; and Brent's early paper [3].

[19] A similar result was obtained independently by H. Abelson and P. Andreae, "Information transfer and area-time tradeoffs for VLSI multiplication", *Comm. ACM* 23, 1980, 20-23.

[20] See J. D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, Maryland, 1984.

[21] R. Ladner and M. Fischer, "Parallel prefix computation", *J. ACM* 27, 1980, 831-838; also W. D. Hillis and G. Steele, "Data parallel algorithms", *Comm. ACM* 12, 1986, 1170-1183.

in VLSI designs was unpopular[22], but the Brent-Kung design technique has recently been applied widely in VLSI implementations of adders[23].

Floating-Point Arithmetic and High-Precision Computations. Brent's interest in *arithmetic* covers a broad range, from computer hardware to software for high-precision computations. Results on binary addition [3,60] are discussed above.

[17] was one of the first papers to show clearly the advantages of binary rounded arithmetic over other systems (e.g. base 16) which were popular at the time for floating-point arithmetic[24].

[34] established the best known upper bounds on the complexity of high precision evaluation of elementary functions. Essentially, an n-bit result can be obtained with $O(\log n)$ multiplications of $O(n)$-bit numbers. Special cases include the evaluation of constants such as π. The idea is to use quadratically convergent iterations such as Gauss's arithmetic-geometric mean or Landen transformations. The elegant quadratically convergent algorithm for π (discovered independently by Brent [28,34] and Salamin[25]) motivated the recent book *Pi and the AGM* [26].

No quadratically convergent algorithm is known for the computation of Euler's constant γ. A fast algorithm[27] was given by Brent and McMillan[28] in [49]. It has never been proved that γ is irrational, but [49] shows that, if $\gamma = p/q$ is rational, then $|q| > 10^{15000}$. This result was recently improved to 10^{250000} (using essentially the same

[22] See C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980, Sec. 5.5.

[23] See N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison Wesley, Reading, Mass., 1985, Sec. 8.2.6; J. L. Hennessy, D. A. Patterson and D. Goldberg, *Computer Architecture: A Quantitative Approach*, second edition, Morgan Kaufmann, San Mateo, California, 1996, A67-A69; and A. Tyagi, "A reduced-area scheme for carry-select adders", *IEEE Trans. on Computers* 42 (1993), 1163-1170.

[24] Since the widespread acceptance of the IEEE floating-point standard it has become the conventional wisdom that base two is preferable to higher bases for floating-point arithmetic. However, in 1972 the subject was controversial enough to delay the publication of [17].

[25] E. Salamin, "Computation of π using arithmetic-geometric mean", *Math. of Comp.* 30, 1976, 565-570.

[26] J. M. Borwein and P. B. Borwein, *Pi and the AGM*, John Wiley and Sons, New York, 1987; also "Ramanujan and Pi", *Scientific American*, February 1988, 66-73.

[27] Faster than an earlier algorithm of Brent [40], which in turn was faster than algorithms of Sweeney (*Math. Comp.* 17, 1963, 170-178) and Knuth (*Math. Comp.* 16, 1962, 275-281). For an historical perspective, see [139].

[28] Nobel laureate, 1951. See J. D. Jackson and W. K. H. Panofsky, "Edwin Mattison McMillan 1907–1991", *Biographical Memoirs of the National Academy of Sciences (USA)*, 69 (1996).

method combined with binary splitting and the ideas of [166]) by Papanikolaou[29].

The paper [52] described the principal approaches possible for *unrestricted* numerical algorithms for the computation of elementary and special functions. Here *unrestricted* means that no *a priori* bounds are placed on the precision required[30]. The algorithms of [52] are generally more practical and cover a wider class of functions than those of [34].

Algorithms intended for digital computers are of little practical use without reliable software implementations. Brent's *MP* package [42,43] for multiple-precision floating-point arithmetic implemented the algorithms of [52]. The algorithms used in *MP* have been adopted in several other software packages[31]. *MP* is still used extensively, for example in recent work by Briggs[32] on the Feigenbaum constants, Odlyzko and te Riele[33] in their disproof of the Mertens conjecture, Odlyzko[34] on the zeros of the Riemann zeta function, and Varga *et al* [35] on bounding the de Bruijn-Newman constant (relevant to the Riemann Hypothesis).

Number-Theoretic Computations. The *Riemann Hypothesis* (RH) is a famous open problem with many consequences in analytic number theory. If false, it could (in principle) be disproved by numerical computations with rigorous error bounds. Using an improvement of a method due to Lehmer, Turing and Littlewood, Brent[47] showed that the first 75,000,000 zeros $\sigma + it$ of the Riemann zeta function in the upper half-plane are simple and lie on the line $\sigma = 1/2$. This is as expected if RH is true. Various interesting empirical results on the

[29] Personal communication from T. Papanikolaou, March 1997.

[30] The term *unrestricted* was used in this sense by C. W. Clenshaw and F. W. J. Olver, "An unrestricted algorithm for the exponential function", *SIAM J. Numer. Anal.* 17, 1980, 310-331.

[31] For example: D. H. Bailey, "Multiprecision translation and execution of Fortran programs", *ACM Trans. on Math. Software* 19 (1993), 288-319; also in J. Cannon's *Magma* and (probably) in commercial packages such as *Mathematica*.

[32] K. Briggs, "A precise calculation of the Feigenbaum constants", *Math. Comp.* 57, 1991, 435-439.

[33] A. M. Odlyzko and H. J. J. te Riele, "Disproof of the Mertens conjecture", *J. Reine Angew. Mathematik* 357, 1985, 138-160.

[34] A. M. Odlyzko, "On the distribution of spacings between zeros of the zeta function", *Math. Comp.* 48, 1987, 273-308. Also A. M. Odlyzko, *The* 10^{20}*-th zero of the Riemann zeta function and 175 million of its neighbors*, to appear.

[35] G. L. Csordas, A. M. Odlyzko, W. Smith and R. S. Varga, "A new Lehmer pair of zeros and a new lower bound for the de Bruijn-Newman constant Λ", *Electronic Transactions on Numerical Analysis* 1 (1993), 104-111. Available by anonymous ftp from `etna@@mcs.kent.edu`. See also T. S. Norfolk, A. Ruttan and R. S. Varga, *A Detailed Numerical Examination of the Tracking of the Zeros of* $F_\lambda(z)$ *to Produce a Lower Bound for the de Bruijn-Newman Constant* Λ, preprint, 1991; and R. S. Varga, *Scientific Computation on Mathematical Problems and Conjectures*, SIAM, Philadelphia, 1990.

distribution of the zeros were obtained in the course of the computation. Using essentially the same method, the result was extended to the first 200,000,000 zeros in [70,81], and to the first 1,500,000,001 zeros by te Riele *et al* [36].

[24] is one of several papers on the empirical distribution of (rational) primes. It showed that primes and twin primes behave in a very different manner (at least up to 10^{11}), and obtained an estimate for Brun's constant[37], only recently improved by Nicely[38] and (independently) Kutrib and Richstein[39].

Brent has made significant contributions to the development of integer factorisation algorithms, a topical subject since the introduction of "public key" cryptosystems whose security depends on the (assumed) difficulty of integer factorisation[40]. The papers [113,115] announced the complete factorisation of the Fermat number $F_{11} = 2^{2^{11}} + 1$ by Brent's variant [97,102] of Lenstra's elliptic curve method. This was a surprise, since Brent and Pollard [61] had factored F_8 (by a different method), but F_9 and F_{10} had not been completely factored[41]. The factorisation of F_{10} was finally completed by Brent in October 1995: see [161].

The existence of odd perfect numbers is an old open problem[42]. Brent, Cohen and te Riele [100,106,116] extended the lower bound on an odd perfect number (if one exists) from 10^{50} to 10^{300}. This required both new mathematical results[43] and good implementations of modern integer factorisation algorithms[44].

[36] J. van de Lune, H. J. J. te Riele and D. T. Winter, "On the zeros of the Riemann zeta function in the critical strip, IV", *Math. of Comp.* 46, 1986, 667-681.

[37] Related papers are [21] on a consequence of the conjecture of Hardy and Littlewood (*Acta Math.* 44, 1923, 1-70) on the distribution of *small* gaps between primes, and [19,57] on *large* gaps between consecutive primes.

[38] T. R. Nicely, "Enumeration to 10^{14} of the Twin Primes and Brun's Constant", *Virginia Journal of Science* 46, 1995, 195-204. This paper describes the computation which uncovered the notorious bug in the Intel Pentium floating-point divide operation. A review is available from
`ftp://nimbus.anu.edu.au/pub/Brent/Nicely.ps.Z` .

[39] Personal communication from J. Richstein, September 21, 1995.

[40] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Comm. ACM* 21, 1978, 120-126; see also [120].

[41] The factorisation of F_9 was completed in 1990 by a world-wide collaboration (in which Brent was a participant) using the *Number Field Sieve* algorithm: see A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The factorization of the ninth Fermat number", *Math. Comp.* 61 (1993), 319-349.

[42] A positive integer is *perfect* if it is equal to the sum of its divisors (other than itself), e.g. $28 = 1 + 2 + 4 + 7 + 14$.

[43] For example, the $q^{5k/2}$ Theorem of [116].

[44] Specifically, the *elliptic curve method* (ECM) and the *multiple polynomial quadratic sieve* (MPQS).

The factorisations required for [116] formed the basis for a significant extension, by Brent, Montgomery and te Riele [134], of the "Cunningham" tables[45] of factors of integers of the form $a^n \pm 1$.

The cyclotomic polynomials $\Phi_n(z)$ satisfy well-known identities of Gauss, Aurifeuille and Lucas[46]. Brent [127,135] has recently given efficient algorithms for computing the polynomials occurring in these identities. He has also given explicit generating functions for these polynomials, and closed-form expressions for the corresponding Aurifeuillian factors of certain integers of the form $a^n \pm 1$. The generating functions display an interesting connection with Dirichlet L-functions and the theory of quadratic fields.

Computational Complexity and Analysis of Algorithms. In an early and significant paper on the *analytic*[47] computational complexity of approximating zeros of functions of one variable, Brent, Winograd and Wolfe [16] showed that locally convergent iteration schemes which use only evaluations of $f, f', \ldots, f^{[d]}$ have order bounded above by $d+2$. This result is best possible[48] and attainable.

Symbolic computation often involves the manipulation of formal power series in one or more variables. Brent and Kung [45] established the best known upper bounds on the complexity of *composition* and *reversion* of power series in one variable, and showed the equivalence of the composition and reversion problems, up to a constant factors. For power series satisfying first and second-order ordinary differential equations, which includes most power series of interest in computation, $O(n \log n)$ algorithms were given. The results were later extended to dense multivariate power series [39] and to generalised composition [50].

Solution of Structured Linear Systems. Brent has made several significant contributions to algorithms for the solution of Toeplitz[49] systems of linear equations and Toeplitz least squares problems. Linear-time algorithms for systolic arrays are mentioned above.

By classical results of Kolmogorov, Wiener and Levinson[50], n by n Toeplitz linear systems can be solved in $O(n^2)$ operations. Brent,

[45] J. Brillhart, H. H. Lehmer, J. L. Selfridge, B. Tuckerman and S. S. Wagstaff, Jr., *Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers*, American Math. Soc., Providence, Rhode Island, 2nd edition, 1985.

[46] See H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Birkhäuser, Boston, 1985; and P. Stevenhagen, "On Aurifeuillian factorizations", *Nederl. Akad. Wetensch. Indag. Math.* 49, 1987, 451-468.

[47] As opposed to *discrete*. See, for example, J. F. Traub and H. Wozniakowski, *A General Theory of Optimal Algorithms*, Academic Press, New York, 1980.

[48] The order is bounded by $d+1$ for iteration schemes *without memory*, e.g. Newton's method.

[49] The algorithms can be generalised to other classes of low displacement-rank matrices, e.g. Hankel, block Toeplitz, etc.

[50] For historical references, see J. R. Bunch, "Stability of methods for solving Toeplitz systems of equations", *SIAM J. Sci. Statist. Comput.* 6 (1985), 349-364.

Gustavson and Yun [59] showed that $O(n(\log n)^2)$ algorithms exist. The algorithms depend on the close connection between computation of the Padé table, or the continued fraction expansion, of a power series, and the solution of a Toeplitz system. The result motivated the development of several other $O(n(\log n)^2)$ algorithms[51].

The algorithms just mentioned are generally numerically unstable unless special conditions (e.g. positive definiteness) are imposed. There is much interest in the stable solution of unsymmetric or indefinite n by n Toeplitz systems $Ax = b$ in $o(n^3)$ operations. Bojanczyk, Brent and de Hoog [92] showed how the orthogonal factorisation $A = QR$ of the Toeplitz matrix A could be computed in $O(n^2)$ steps on a sequential computer, and in $O(n)$ steps on a parallel computer with $O(n)$ processors[52]. The algorithm is a significant improvement over an earlier (unstable and inherently sequential) algorithm of Sweet[53], and gives a weakly stable algorithm for the solution of Toeplitz linear systems and linear least squares problems [111]. More recent work on the stability of fast algorithms for Toeplitz and related matrix problems, in collaboration with Bojanczyk, de Hoog and Sweet, may be found in [143,144,157]. For example, in [144] the stability of the Bareiss (Schur) algorithm was shown to be similar to that of Gaussian elimination[54].

Optimisation and Solution of Nonlinear Algebraic Systems. The monograph *Algorithms for Minimization without Derivatives* [11] considers algorithms for zero-finding and optimisation under the constraint that only function (not derivative) evaluations are permitted. It improves on Dekker's algorithm for the problem of finding a zero of a function of one variable, and gives an analogous algorithm for minimisation of a function of one variable. Implementations of these algorithms are still in wide use. Other problems considered in [11] include the global optimisation of a function of a small number of variables, given bounds on derivatives; and the unconstrained optimisation of a

[51] R. R. Bitmead and B. D. O. Anderson, "Asymptotically fast solution of Toeplitz and related systems of linear equations", *J. Linear Algebra Appl.* 34, 1980, 103-116; F. R. de Hoog, "A new algorithm for solving Toeplitz systems of equations", *J. Linear Algebra Appl.* 88/89, 1987, 122-138; G. S. Ammar and W. B. Gragg, "Superfast solution of real positive definite Toeplitz systems", *SIAM J. Matrix Anal. Appl.* 9, 1988, 61-76; and others [108].

[52] R is upper triangular. For simplicity we assume that the matrix A is square, although this is not necessary.

[53] D. R. Sweet, "Fast Toeplitz orthogonalization", *Numerische Mathematik* 43, 1984, 1-21.

[54] For generalisations of this result, see S. Chandrasekaran and A. H. Sayed, "Stabilizing the generalized Schur algorithm", *SIMAX* 17 (1996), 950-983; also M. Stewart and P. Van Dooren, "Stability issues in the factorization of structured matrices", *SIAM J. Matrix Anal. Appl.*, 1997, to appear.

function of several variables. Some modifications to Powell's (1964) algorithm[55] are suggested.

In [14] Brent considered efficient methods for approximating the solution of a system of nonlinear equations, using only function evaluations. He improved on the method of Brown and Conte[56] by using (numerically stable) orthogonal transformations and optimising the Ostrowski efficiency. Brent's method is still widely used and recommended[57].

Other Recent Research. In [119], Brent described a fast training algorithm for multi-layer feed-forward neural nets[58]. The algorithm is much faster than the popular but inefficient "back propagation" algorithm. It also demonstrates a close connection between neural nets and older classification and data retrieval methods in common use by statisticians and computer scientists.

The fundamental results on linear recurrences of integers mod m were obtained many years ago[59]. Recently, Brent [133] obtained a surprising new result on the period of three-term linear recurrences mod 2^w. The result has applications to the popular class of "generalised Fibonacci" pseudo-random number generators. Brent has made significant contributions to the development of uniform [132] and normal [23,141,170] random number generators.

Brent is joint leader (with R. B. Stanton) of the ANU-Fujitsu CAP Project[60], which is a collaborative R&D project to develop systems and applications software for MIMD computers such as the Fujitsu AP1000/AP3000. Some recent highlights are the implementation of parallel Linux, parallel file systems, and parallel sorting algorithms. Brent is also joint leader (with M. R. Osborne) of a project[61] to develop scientific subroutine libraries on parallel/vector computers such as the Fujitsu VPP300.

---

[55] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives", *Computer J.* 7, 1964, 155-162.

[56] K. M. Brown and S. D. Conte, "The solution of simultaneous nonlinear equations", *Proc. 22nd National Conference of the ACM*, Thompson Book Co., Washington, D.C., 1967, 111-114.

[57] J. J. Moré and M. Y. Cosnard, "Numerical solution of nonlinear equations", *ACM Transactions on Mathematical Software* 5, 1979, 64-85.

[58] F. Crick, "The recent excitement about neural networks", *Nature* 337, 1989, 129-132.

[59] M. Ward, "The arithmetical theory of linear recurring series", *Trans. Amer. Math. Soc.* 35, 1933, 600-628.

[60] R. P. Brent and R. B. Stanton (editors), *ANU/Fujitsu CAP Research Program Report 1996*, Department of Computer Science and Computer Sciences Laboratory, Australian National University, October 1996.

[61] See, for example, [148] and various *Area 4 Working Notes*.