THE AUSTRALIAN NATIONAL UNIVERSITY

TR-CS-97-11

Two New Factors of Fermat Numbers

Richard P. Brent, Richard E. Crandall and Karl Dilcher

May 1997

Joint Computer Science Technical Report Series

Department of Computer Science
Faculty of Engineering and Information Technology

Computer Sciences Laboratory
Research School of Information Sciences and Engineering

This technical report series is published jointly by the Department of Computer Science, Faculty of Engineering and Information Technology, and the Computer Sciences Laboratory, Research School of Information Sciences and Engineering, The Australian National University.

Please direct correspondence regarding this series to:

Technical Reports
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200
Australia

or send email to:

`Technical.Reports@cs.anu.edu.au`

A list of technical reports, including some abstracts and copies of some full reports may be found at:

`http://cs.anu.edu.au/techreports/`

Recent reports in this series:

TR-CS-97-10 Andrew Tridgell, Richard Brent, and Brendan McKay. *Parallel integer sorting*. May 1997.

TR-CS-97-09 M. Manzur Murshed and Richard P. Brent. *Constant time algorithms for computing the contour of maximal elements on the Reconfigurable Mesh*. May 1997.

TR-CS-97-08 Xun Qu, Jeffrey Xu Yu, and Richard P. Brent. *A mobile TCP socket*. April 1997.

TR-CS-97-07 Richard P. Brent. *A fast vectorised implementation of Wallace's normal random number generator*. April 1997.

TR-CS-97-06 M. Manzur Murshed and Richard P. Brent. *RMSIM: a serial simulator for reconfigurable mesh parallel computers*. April 1997.

TR-CS-97-05 Beat Fischer. *Collocation and filtering — a data smoothing method in surveying engineering and geodesy*. March 1997.

TWO NEW FACTORS OF FERMAT NUMBERS

R. P. BRENT, R. E. CRANDALL, AND K. DILCHER

ABSTRACT. We report the discovery of new 27-decimal digit factors of the thirteenth and sixteenth Fermat numbers. Each of the new factors was found by the elliptic curve method. After division by the new factors and other known factors, the quotients are seen to be composite numbers with 2391 and 19694 decimal digits respectively.

1. INTRODUCTION

For a nonnegative integer n, the n-th *Fermat number* is $F_n = 2^{2^n} + 1$. It is known [12] that F_n is prime for $0 \le n \le 4$, and composite for $5 \le n \le 23$. For a brief history of attempts to factor Fermat numbers, we refer to [3, §1] and [5].

In recent years several factors of Fermat numbers have been found by the elliptic curve method (ECM). Brent [2, 3, 4] completed the factorization of F_{10} (by finding a 40-digit factor) and F_{11}. He also "rediscovered" the 49-digit factor of F_9 and the five known prime factors of F_{12}. Crandall [10] discovered two 19-digit factors of F_{13}.

This paper reports the discovery of 27-digit factors of F_{13} and F_{16} (the factor of F_{13} was announced in [3, §8]). Both factors were found by ECM, although the implementations and hardware were different. In fact, we used Dubner Crunchers (see §3) on F_n for $12 \le n \le 15$, and Sun workstations with DWT multiplication (see §4.2) for $16 \le n \le 21$. Details of the computations on F_{13} and F_{16} are given in §5 and §6 respectively.

F_{16} is probably the largest number for which a nontrivial factor has been found by ECM. Factors of larger numbers are customarily found by trial division [16, 18].

2. THE ELLIPTIC CURVE METHOD

ECM was invented by H. W. Lenstra, Jr. [22]. Various practical refinements were suggested by Brent [1], Montgomery [23, 24], and Suyama [31]. We refer to [3, 14, 21, 25, 30] for a description of ECM and some of its implementations.

In the following, we assume that ECM is used to find a prime factor $p > 3$ of a composite number N, not a prime power [20, §2.5]. The first-phase limit for ECM is denoted by B_1.

Although p is unknown, it is convenient to describe ECM in terms of operations in the finite field $K = GF(p) = \mathbf{Z}/p\mathbf{Z}$. In practice we work modulo N (or sometimes modulo a multiple of N, if the multiple has a convenient binary representation),

---

1991 *Mathematics Subject Classification.* 11Y05, 11B83, 11Y55; Secondary 11-04, 11A51, 11Y11, 11Y16, 14H52, 65Y10, 68Q25.

Key words and phrases. discrete weighted transform, DWT, ECM, elliptic curve method, factorization, Fermat number, F_{13}, F_{16}, integer factorization.

Version of May 26, 1997. Submitted to *Mathematics of Computation.*
 rpb175tr.tex typeset using $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX.

and occasionally perform GCD computations which will detect a nontrivial factor of N.

The computations reported here used two parameterizations of elliptic curves. These are the symmetrical Cauchy form [9, §4.2]:

$$x^3 + y^3 + z^3 = \kappa xyz \,, \tag{1}$$

and the homogeneous form recommended by Montgomery [23]:

$$by^2 z = x^3 + ax^2 z + xz^2 \,. \tag{2}$$

Here κ, a and b are constants satisfying certain technical conditions. For details we refer to [3, §2.1].

The points (x, y, z) satisfying (1) or (2) are thought of as representatives of elements of $P^2(K)$, the projective plane over K, i.e. the points (x, y, z) and (cx, cy, cz) are regarded as equivalent if $c \not\equiv 0 \bmod p$. We write $(x : y : z)$ for the equivalence class containing (x, y, z). When using (2) it turns out that the y-coordinate is not required, and we can save work by not computing it. In this case we write $(x : : z)$.

2.1. **The starting point.** An advantage of using (2) over (1) is that the group order is always a multiple of four (Suyama [31]; see [23, p. 262]). It is possible to ensure that the group order is divisible by $8, 12$ or 16. For example, if $\sigma \notin \{0, 1, 5\}$,

$$\begin{aligned} u &= \sigma^2 - 5, \;\; v = 4\sigma, \\ x_1 &= u^3, \;\; z_1 = v^3, \\ a &= \frac{(v-u)^3(3u+v)}{4u^3 v} - 2, \end{aligned} \tag{3}$$

then the curve (2) has group order divisible by 12. As starting point we can take $(x_1 : : z_1)$. It is not necessary to specify b or y_1. When using (2) we assume that the starting point is chosen as in (3), with σ a pseudo-random integer.

3. The Dubner Cruncher

The Dubner Cruncher [8, 15] is a board which plugs into an IBM-compatible PC. The board has a digital signal processing chip (LSI Logic L64240 MFIR) which, when used for multiple-precision integer arithmetic, can multiply two 512-bit numbers in 3.2 μsec. A software library has been written by Harvey and Robert Dubner [15]. This library allows a C programmer to use the Cruncher for multiple-precision integer arithmetic. Some limitations are:

1. Communication between the Cruncher and the PC (via the PC's ISA bus) is relatively slow, so performance is much less than the theoretical peak for numbers of less than say 1000 bits.
2. Because of the slow communication it is desirable to keep operands in the on-board memory, of which only 256 KByte is accessible to the C programmer.

The combination of a cheap PC and a Cruncher board ($US2,500) is currently very cost-effective for factoring large integers by ECM. The effectiveness of the Cruncher increases as the integers to be factored increase in size. However, due to memory limitations, we have not attempted to factor Fermat numbers larger than F_{15} on a Cruncher. A number the size of F_{15} requires 4 KByte of storage.

4. Arithmetic

4.1. **Multiplication and division.** Most of the cost of ECM is in performing multiplications mod N. Our Cruncher programs all use the classical $O(w^2)$ algorithm to multiply w-bit numbers. Karatsuba's algorithm [19, §4.3.3] or other "fast" algorithms [11, 13] are preferable for large w on a workstation. The crossover point depends on details of the implementation. Morain [26, Ch. 5] states that Karatsuba's method is worthwhile for $w \geq 800$ on a 32-bit workstation. On a Cruncher the crossover is much larger because the multiplication time is essentially linear in w for $w < 10000$ (see [3, Table 4]).

Our programs avoid division where possible. If the number N to be factored is a composite divisor of F_n, then the elliptic curve operations are performed mod F_n rather than mod N. At the end of each phase we compute a GCD with N. The advantage of this approach is that we can perform the reductions mod F_n using binary shift and add/subtract operations, which are much faster than multiply or divide operations. Thus, our Cruncher programs run about twice as fast on Fermat (or Mersenne) numbers as on "general" numbers.

4.2. **Use of the discrete weighted transform.** For F_n with $n > 15$ we found it more efficient overall to employ standard workstations with an asymptotically fast multiplication algorithm rather than special hardware. For these larger F_n we employed the "discrete weighted transform" (DWT) of Crandall and Fagin [10, 13]. In this scheme, one exploits the fact that multiplication modulo F_n is essentially a negacyclic convolution [11] which can be effected via three DWTs. For two integers x, y to be multiplied modulo F_n, one splits each of x, y into D digits in some base W, with $D \log_2 W = 2^n$. We actually used $W = 2^{16}$, and employed the "balanced digit" scheme which is known to reduce floating-point convolution errors [10]. The three length-$D/2$ DWTs were then performed using a split-radix complex-FFT algorithm. The operation complexity is $O(D \log D)$, and 64-bit IEEE floating point arithmetic is sufficiently precise to attack Fermat numbers at least as large as F_{21} in this way. Our DWT approach becomes more efficient than "grammar-school" $O(D^2)$ methods in the region $n \sim 12$. However, the Cruncher hardware is so fast that a Cruncher performs faster than a workstation for $n \leq 15$.

The advantage of DWT methods is not restricted to multiplication. The elliptic curve algebra using the Montgomery parameterization (2) can be sped up in a fundamental way via transforms. The details are given in [10]; for present purposes we give one example of this speedup. For the point-adding operation

$$\frac{x_{m+n}}{z_{m+n}} = \frac{z_{|m-n|}(x_m x_n - z_m z_n)^2}{x_{|m-n|}(x_m z_n - x_n z_m)^2}$$

it is evident that one can compute the transforms of x_m, x_n, z_m, z_n, then compute the relevant cross-products in spectral space, then use the (stored) transforms of $x_{|m-n|}, z_{|m-n|}$ to obtain x_{m+n}, z_{m+n} in a total of 14 DWTs, which is equivalent to $14/3 \simeq 4.67$ multiplies. Similar enhancements are possible for point-doubling.

Memory capacity is a pressing concern for the largest Fermat numbers under consideration (F_{16} through F_{21}). Another enhancement for the ECM/DWT implementation is to perform the second stage of ECM in an efficient manner. We note that a difference of x-coordinates can be calculated from:

$$x_m z_n - x_n z_m = (x_m - x_n)(z_m + z_n) - x_m z_m + x_n z_n,$$

and a small table of $x_m z_m$ can be stored. Thus the coordinate difference requires only one multiply, plus one multiply for accumulation of all such differences. Again, if DWTs are used, one stores the transforms of $x_m, z_m, x_m z_m$, whence the difference calculation comes down to 2/3 of a multiply, plus the accumulation multiply. The accumulation of differences can likewise be given a transform enhancement, with the result that each coordinate difference in stage two consumes only 4/3 of a multiply. In practice, this second stage efficiency allows the choices of stage two limit B_2 at least as large as $50B_1$.

4.3. **GCD computation.** It is nontrivial to compute GCDs for numbers in the F_{21} region. We used a recursive GCD implementation by J. P. Buhler [6], based on the Schönhage algorithm [7, 27]. The basic idea is to recursively compute a 2×2 matrix M such that if $v = (a,b)^T$ is the column vector containing the two numbers whose GCD we desire, then $Mv = (0, \gcd(a,b))^T$. The matrix M is a product of 2×2 matrices and is computed by finding the "first half" of the product recursively. The first-half function calls itself twice recursively (for details see [7]). In practice it is important to revert to a classical algorithm (such as Euclid's) for small enough integers. We found that GCDs taken during factorization attempts on numbers as small as F_{13} could be speeded up by using the recursive algorithm. In the region of F_{21} the recursive approach gives a speedup by a factor of more than 100 over the classical GCD.

The Cruncher programs use the classical (non-recursive) GCD but only perform two GCDs per curve (one at the end of each phase). This is possible, at a small cost in additional multiplications, because the programs use the homogeneous forms (1) and (2) and never divide by the z-coordinate.

5. A new factor of F_{13}

Our first Cruncher ECM program [3, Program F] was implemented and debugged early in December 1994. It used the Cauchy form (1) with a "birthday paradox" second phase. In the period January – June 1995 we used a Cruncher in an 80386/40 PC to attempt to factor F_{13} (and some other numbers). We mainly used phase 1 limit $B_1 = 100000$. On F_{13} each curve took 137 minutes (91 minutes for phase 1 and 46 minutes for phase 2). At the time three prime factors of F_{13} were known:

$$F_{13} = 2710954639361 \cdot 2663848877152141313 \cdot 3603109844542291969 \cdot c_{2417} \, .$$

The first factor was found by Hallyburton and Brillhart [17]. The second and third factors were found by Crandall [10] on Zilla net (a network of about 100 workstations) in January and May 1991, using ECM.

On June 16, 1995 our Cruncher program found a fourth factor

$$p_{27} = 319546020820551643220672513 = 2^{19} \cdot 51309697 \cdot 11878566851267 + 1$$

after a total of 493 curves with $B_1 = 100000$. The overall machine time was about 47 days. We note that $p_{27} + 1 = 2 \cdot 3 \cdot 7^3 \cdot 59 \cdot p_{22}$. The factorizations of $p_{27} \pm 1$ explain why Pollard's $p \pm 1$ methods could not find the factor p_{27} in a reasonable time.

The successful curve was of the form (1), with initial point $(x_1 : y_1 : z_1) = (150400588188733400929847531 : 277194908510676462587880207 : 1) \bmod p_{27}$ and group order

$$g = 3^2 \cdot 7^2 \cdot 13 \cdot 31 \cdot 3803 \cdot 6037 \cdot 9887 \cdot 28859 \cdot 274471 \, .$$

Using Fermat's little theorem [5, p. lviii], we found the 2391-digit quotient c_{2417}/p_{27} to be composite. Thus, we now know that

$$\begin{aligned} F_{13} \;=\; & 2710954639361 \cdot 2663848877152141313 \cdot \\ & 3603109844542291969 \cdot 319546020820551643220672513 \cdot c_{2391} \, . \end{aligned}$$

At about the time that the p_{27} factor of F_{13} was found, our Cruncher ECM program was modified to use the Montgomery form (2) with the "improved standard continuation" second phase [3, §3.2]. Testing the new program with $B_1 = 500000$ and second-phase limit $B_2 = 35B_1$, we found p_{27} seven times, with a total of 579 curves. The expected number of curves, predicted as in [3, § 4.4], is $7 \times 137 = 959$. The successful curves are defined by (3), with σ and the group order g given in Table 1.

TABLE 1. Some curves finding the p_{27} factor of F_{13}

σ	g
1915429	$2^4 \cdot 3 \cdot 29 \cdot 857 \cdot 12841 \cdot 42451 \cdot 48299 \cdot 10173923$
2051632	$2^3 \cdot 3 \cdot 17 \cdot 19 \cdot 1031 \cdot 23819 \cdot 65449 \cdot 86857 \cdot 295277$
2801740	$2^2 \cdot 3 \cdot 7 \cdot 79 \cdot 157 \cdot 19813 \cdot 89237 \cdot 122819 \cdot 1412429$
4444239	$2^2 \cdot 3 \cdot 23 \cdot 173 \cdot 191 \cdot 907 \cdot 1493 \cdot 3613 \cdot 4013 \cdot 1784599$
6502519	$2^2 \cdot 3 \cdot 23 \cdot 131 \cdot 14011 \cdot 305873 \cdot 433271 \cdot 4759739$
8020345	$2^3 \cdot 3 \cdot 17 \cdot 23 \cdot 41 \cdot 113 \cdot 271 \cdot 3037 \cdot 10687 \cdot 12251 \cdot 68209$
8188713	$2^2 \cdot 3^2 \cdot 17 \cdot 41 \cdot 47 \cdot 139 \cdot 181 \cdot 34213 \cdot 265757 \cdot 1184489$

The fact that our programs found the same 27-digit factor many times suggests (but does not prove) that the unknown factors of F_{13} are larger than p_{27}.

When testing our program with $B_1 = 500000$, we also "rediscovered" both of Crandall's 19-digit factors using the *same* elliptic curve (mod F_{13}). In fact, our program returned the 39-digit product of Crandall's factors. Taking $\sigma = 6505208$ in (3), the group corresponding to the factor 2663848877152141313 has order $2^2 \cdot 3^2 \cdot 1879 \cdot 2179 \cdot 3677 \cdot 4915067$, and the group corresponding to the factor 3603109844542291969 has order $2^4 \cdot 3 \cdot 7^2 \cdot 22003 \cdot 79601 \cdot 874661$, so both factors will be found if $B_1 \geq 79601$ and $B_2 \geq 4915067$.

6. A NEW FACTOR OF F_{16}

The DWT/ECM program was run on a small network of SPARCstations at Dalhousie University from June 1996, in an attempt to find factors of $F_{16}, \ldots, F_{20}$. It was known that

$$F_{16} = 825753601 \cdot c_{19720}$$

where the 9-digit factor was found by Selfridge [28].

From the middle of September 1996 to the end of December 1996 an average of 6 SPARCstations ran the DWT/ECM program exclusively on F_{16} and in December found a new factor

$$p_{27} = 188981757975021318420037633$$

of F_{16}. Since

$$p_{27} - 1 = 2^{20} \cdot 3^2 \cdot 31 \cdot 37 \cdot 13669 \cdot 1277254085461$$

and

$$p_{27} + 1 = 2 \cdot 240517 \cdot 1389171559 \cdot 282805744939,$$

it would have been very difficult to find p_{27} by Pollard's $p \pm 1$ methods.

We remark that p_{27} was found twice – the first time with $B_1 = 400000$, $B_2 = 50B_1$, $\sigma = 1944934539$, and group order

$$g = 2^2 \cdot 3 \cdot 5^3 \cdot 7 \cdot 13 \cdot 19 \cdot 83 \cdot 113 \cdot 2027 \cdot 386677 \cdot 9912313\ ,$$

and the second time with $B_1 = 200000$, $B_2 = 50B_1$, $\sigma = 125546653$, and group order

$$g = 2^2 \cdot 3^2 \cdot 7^2 \cdot 109 \cdot 761 \cdot 2053 \cdot 20297 \cdot 101483 \cdot 305419\ .$$

The ECM limits were set so that each curve required roughly four days of CPU. Altogether, we ran 130 curves with various $B_1 \in [50000, 400000]$.

The quotient $q = c_{19720}/p_{27}$ was a 19694-digit number. We computed $x = 3^q \bmod q$ and found $x \neq 3$. Thus, q is composite, and we now know that

$$F_{16} = 825753601 \cdot 188981757975021318420037633 \cdot c_{19694}$$

As a check, the computation of q and x was performed independently by Brent and Crandall (using different programs on different machines in different continents). In both cases the computations found $x \bmod 2^{16} = 12756$.

7. Acknowledgements

Harvey Dubner provided a Dubner Cruncher and encouraged one of us to implement ECM on it. Dennis Andriolo and Robert Dubner provided assistance with aspects of the Cruncher hardware and software. We are indebted to J. P. Buhler for his inestimable aid in the optimization of various large-integer algorithms.

References

[1] R. P. Brent, *Some integer factorization algorithms using elliptic curves,* Australian Computer Science Communications **8** (1986), 149–163. Also Report CMA-R32-85, Centre for Mathematical Analysis, Australian National University, Canberra, Sept. 1985, 20 pp.

[2] R. P. Brent, *Factorization of the eleventh Fermat number (preliminary report),* AMS Abstracts **10** (1989), 89T-11-73. `ftp://nimbus.anu.edu.au/pub/Brent/rpb113a.dvi.gz` .

[3] R. P. Brent, *Factorization of the tenth and eleventh Fermat numbers,* Report TR-CS-96-02, Computer Sciences Laboratory, Australian National Univ., Canberra, Feb. 1997. `ftp://nimbus.anu.edu.au/pub/Brent/rpb161tr.dvi.gz` .

[4] R. P. Brent, *Factorization of the tenth Fermat number,* Math. Comp., 1998, to appear. `ftp://nimbus.anu.edu.au/pub/Brent/rpb161.dvi.gz` .

[5] J. Brillhart, D. 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*, 2nd ed., Amer. Math. Soc., Providence, RI, 1988.

[6] J. P. Buhler, personal communication to Crandall, 1993.

[7] P. Bürgisser, M. Clausen and M. A. Shokrollahi, *Algebraic Complexity Theory*, Springer-Verlag, 1997.

[8] C. Caldwell, *The Dubner PC Cruncher – a microcomputer coprocessor card for doing integer arithmetic*, review in J. Rec. Math. **25**(1), 1993.

[9] D. V. and G. V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests,* Adv. in Appl. Math. **7** (1986), 385–434.

[10] R. E. Crandall, *Projects in scientific computation,* Springer-Verlag, New York, 1994.

[11] R. E. Crandall, *Topics in advanced scientific computation,* Springer-Verlag, New York, 1996.

[12] R. Crandall, J. Doenias, C. Norrie, and J. Young, *The twenty-second Fermat number is composite,* Math. Comp. **64** (1995), 863–868.

[13] R. Crandall and B. Fagin, *Discrete weighted transforms and large-integer arithmetic,* Math. Comp. **62** (1994), 305–324.

[14] B. Dixon and A. K. Lenstra, *Massively parallel elliptic curve factoring,* Proc. Eurocrypt '92, Lecture Notes in Computer Science **658**, Springer-Verlag, Berlin, 1993, 183–193.

[15] H. Dubner and R. Dubner, *The Dubner PC Cruncher: Programmers Guide and Function Reference,* February 15, 1993.

[16] G. B. Gostin, *New factors of Fermat numbers,* Math. Comp. **64** (1995), 393–395.

[17] J. C. Hallyburton and H. Brillhart, *Two new factors of Fermat numbers,* Math. Comp. **29** (1975), 109–112. Corrigendum, *ibid* **30** (1976), 198.

[18] W. Keller, *Factors of Fermat numbers and large primes of the form $k \cdot 2^n + 1$,* Math. Comp. **41** (1983), 661–673. Also part II, preprint, Universität Hamburg, Sept. 27, 1992 (available from the author).

[19] D. E. Knuth, *The art of computer programming, Volume 2: Seminumerical algorithms* (2nd ed.), Addison-Wesley, Menlo Park, CA, 1981.

[20] 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.

[21] A. K. Lenstra and M. S. Manasse, *Factoring by electronic mail,* Proc. Eurocrypt '89, Lecture Notes in Computer Science **434**, Springer-Verlag, Berlin, 1990, 355–371.

[22] H. W. Lenstra, Jr., *Factoring integers with elliptic curves,* Ann. of Math. (2) **126** (1987), 649–673.

[23] P. L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization,* Math. Comp. **48** (1987), 243–264.

[24] P. L. Montgomery, *An FFT extension of the elliptic curve method of factorization,* Ph. D. dissertation, Mathematics, University of California at Los Angeles, 1992. `ftp://ftp.cwi.nl/pub/pmontgom/ucladissertation.psl.Z` .

[25] P. L. Montgomery, *A survey of modern integer factorization algorithms,* CWI Quarterly **7** (1994), 337–366.

[26] F. Morain, *Courbes elliptiques et tests de primalité,* Ph. D. thesis, Univ. Claude Bernard – Lyon I, France, 1990. `ftp://ftp.inria.fr/INRIA/publication/Theses/TU-0144.tar.Z` .

[27] A. Schönhage, *Schnelle Berechnung von Kettenbruchentwicklungen,* Acta Inf. **1** (1971), 139–144.

[28] J. L. Selfridge, *Factors of Fermat numbers,* MTAC **7** (1953), 274–275.

[29] J. H. Silverman, *The arithmetic of elliptic curves,* Graduate Texts in Mathematics **106**, Springer-Verlag, New York, 1986.

[30] R. D. Silverman and S. S. Wagstaff, Jr., *A practical analysis of the elliptic curve factoring algorithm,* Math. Comp. **61** (1993), 445–462.

[31] H. Suyama, *Informal preliminary report* (8), personal communication to Brent, October 1985.

Computer Sciences Laboratory, Australian National University, Canberra, ACT 0200, Australia

E-mail address: `Richard.Brent@anu.edu.au`

Center for Advanced Computation, Reed College, Portland, OR 97202, USA

E-mail address: `crandall@reed.edu`

Department of Mathematics and Statistics, Dalhousie University, Halifax, Nova Scotia B3H 3J5, Canada

E-mail address: `dilcher@cs.dal.ca`