MATHEMATICS OF COMPUTATION
Volume 66, Number 217, January 1997, Pages 433–449
S 0025-5718(97)00791-6

A SEARCH FOR WIEFERICH AND WILSON PRIMES

RICHARD CRANDALL, KARL DILCHER, AND CARL POMERANCE

ABSTRACT. An odd prime p is called a *Wieferich prime* if

$$2^{p-1} \equiv 1 \pmod{p^2};$$

alternatively, a *Wilson prime* if

$$(p-1)! \equiv -1 \pmod{p^2}.$$

To date, the only known Wieferich primes are $p = 1093$ and 3511, while the only known Wilson primes are $p = 5, 13$, and 563. We report that there exist no new Wieferich primes $p < 4 \times 10^{12}$, and no new Wilson primes $p < 5 \times 10^8$. It is elementary that both defining congruences above hold merely (mod p), and it is sometimes estimated on heuristic grounds that the "probability" that p is Wieferich (independently: that p is Wilson) is about $1/p$. We provide some statistical data relevant to occurrences of small values of the pertinent Fermat and Wilson quotients (mod p).

0. INTRODUCTION

Wieferich primes figure strongly in classical treatments of the first case of Fermat's Last Theorem ("FLT(I)"). For an odd prime p not dividing xyz, Wieferich [26] showed that $x^p + y^p + z^p = 0$ implies $2^{p-1} \equiv 1 \pmod{p^2}$. Accordingly, we say that an odd prime p is a Wieferich prime if the Fermat quotient

$$q_p(2) = \frac{2^{p-1} - 1}{p}$$

vanishes (mod p). The small Wieferich primes $p = 1093$ and 3511 have long been known. Lehmer [16] established that there exist no other Wieferich primes less than 6×10^9, and David Clark [6] recently extended this upper bound to 6.1×10^{10}. This paper reports extension of the search limit to 4×10^{12}, without a single new Wieferich prime being found. The authors searched to 2×10^{12}. David Bailey used some of our techniques (and some machine-dependent techniques of his own; see §4: Machine considerations) to check our runs to 2×10^{12}, and to extend the search to the stated limit of 4×10^{12}. Richard McIntosh likewise verified results over several long intervals.

Wilson's classical theorem, that if p is prime, then $(p-1)! \equiv -1 \pmod{p}$, and Lagrange's converse, that this congruence characterizes the primes, are certainly

Received by the editor May 19, 1995 and, in revised form, November 27, 1995 and January 26, 1996.

1991 *Mathematics Subject Classification.* Primary 11A07; Secondary 11Y35, 11–04.

Key words and phrases. Wieferich primes, Wilson primes, Fermat quotients, Wilson quotients, factorial evaluation.

The second author was supported in part by a grant from NSERC. The third author was supported in part by an NSF grant.

©1997 American Mathematical Society

elegant as a definitive primality test, but are also certainly difficult to render practical. Even in the domain of factoring, if arbitrary $M!$ (mod N) were sufficiently easy to evaluate, one would be able to take the GCD of such factorials with N routinely to produce factors of N. One therefore expects factorial evaluation to be problematic. But their difficulty is not the only interesting aspect of factorial evaluations. Indeed, FLT(I) results can be cast in terms of certain binomial coefficients and the Wilson quotients

$$w_p := \frac{(p-1)!+1}{p}$$

(E. Lehmer [17]) whose vanishing (mod p) signifies the Wilson primes.

Though our Wieferich search involved successive refinements of one basic algorithm, namely the standard binary powering ladder, we found that many fascinating and disparate options abound for the factorial evaluation. What might be called the "straightforward" approach to a Wilson search is simply to accumulate the relevant product: set $s = 1$, then for $n = 2$ through $p-1$, accumulate $s := s * n$ (mod p^2). This straightforward approach turns out to admit of dramatic enhancements. By careful selection and testing amongst a wide range of options, we were able to extend W. Keller's search limit of 3×10^6 (see Ribenboim [22]), and a limit by Gonter and Kundert [13] of 1.88×10^7. Our conclusion is: save $p = 5, 13$, and 563 there are no Wilson primes less than 5×10^8. Again, Richard McIntosh verified our results in various disjoint regions below our search limit.

1. Wieferich primes

Note the following amusing numerological observation: if the positions of all the ones in the binary representation of p lie in arithmetic progression, then p cannot be a Wieferich prime. (To see this, note that if $p = 1 + 2^k + \cdots + 2^{(t-1)k} = (2^{kt}-1)/(2^k-1)$, then the exponent that 2 belongs to modulo p is kt, so kt divides $p-1$. Raising $2^{kt} = 1 + p(2^k-1)$ to the $(p-1)/(kt)$ power, we see that $2^{p-1} \equiv 1 + p(2^k-1)(p-1)/(kt) \not\equiv 1$ (mod p^2).) Thus the set of Wieferich primes cannot contain Fermat primes ($p = 10\ldots01$ (binary)) or Mersenne primes ($p = 11\ldots11$ (binary)) or primes such as

$$p = 10000001000000100000010000001000000010000001 \text{ (binary)}$$

(see also Ribenboim [21, p.154]). But such special forms are rare indeed, hardly affecting any exhaustive search for Wieferich primes. Generally speaking, there is no known way to resolve 2^{p-1} (mod p^2), other than through explicit powering computations. Let us denote a "straightforward" Wieferich search scheme as one in which a standard binary power ladder is employed, with all arithmetic done in standard high-precision fashion (mod p^2), meaning in particular that standard long division (by p^2) is employed for the mod operations. In this straightforward scheme, intermediate integer results would at times be nearly as large as p^4, always to be reduced, of course, (mod p^2). Beyond this straightforward scheme there exist various enhancements, to which we next turn.

The first enhancement we employed has been used by previous investigators (Lehmer [16], Montgomery [19]). This enhancement is useful if the low-level multiplication in the computing machinery cannot handle products of magnitude p^4. The idea is to invoke base-p representations and thereby "split" the multiplication of two numbers (mod p^2). Let any $x = a + bp$ (mod p^2) be represented by $\{a, b\}$,

with both a, b always reduced (mod p). Then the requisite doubling and squaring operations within a standard powering ladder may use the following formulae. We use the operation "$\%p$" to denote modular reduction to a residue between 0 and $p-1$ inclusive, and [ ] to denote greatest integer part:

$$\begin{aligned} 2x &= \{(2a)\%p, (2b + [(2a)/p])\%p\}, \\ x^2 &= \{(a^2)\%p, (2ab + [a^2/p])\%p\}. \end{aligned} \tag{1.1}$$

For primes $p \approx 10^{12}$, our machinery benefitted considerably from this base-p method. On some machines, in fact, the base-p scheme (1.1) is about twice as fast as the straightforward scheme.

A second, and in practice a telling enhancement, is to invoke what we shall call "steady-state" division. In this technique, which exploits the fact that a denominator (for example p or p^2) is fixed, a divide operation can be performed in roughly the same time as a single multiply. The basic idea is described with respect to large-integer arithmetic in [8], and was noticed within a floating-point scenario by Montgomery [19]. To divide by some N repeatedly, one computes and stores a fixed "reciprocal" of N, then uses this systematically to resolve arbitrary values of $[x/N]$. Choose some integer s such that $2^s > N^2$. Then consider, for arbitrary $0 \le x < N^2$, the quantity

$$\left[\frac{\left[\frac{2^s}{N}\right] x}{2^s}\right] = \left[\frac{\left(\frac{2^s}{N} - \theta\right) x}{2^s}\right] = \left[\frac{x}{N} - \frac{x\theta}{2^s}\right],$$

where $0 \le \theta < 1$. Clearly, the far right-hand side is either $[x/N]$ or $[x/N] - 1$. The point is, for given N the "reciprocal" $[2^s/N]$ need be computed only once. And once it is computed, the far left-hand side may be evaluated with a single multiplication and a shift. Let "$>> s$" denote a right-binary-shift by s bits. We express the steady-state division arithmetic in the following way:

Theorem 1. *Let $0 \le x < N^2$, and let $r = [2^s/N]$, where $2^s > N^2$. Then $[x/N]$ is either $(rx) >> s$ or $((rx) >> s) + 1$.*

Armed with the base-p arithmetic of (1.1) and with Theorem 1, we may now exhibit an efficient pseudocode sequence for the squaring of a representation $x = \{a, b\}$:

(1.2)

```
(* Assume r = [2^s/p], 2^s > p^2, has been computed once for given p.  *)
        b := b * a;
        c := (b * r) >> s;
        b := b - p * c;
        if (b >= p) b := b - p;
        a := a * a;
        d := (a * r) >> s;
        a := a - p * d;
        if (a >= p) {
                a := a - p;
                d := d + 1;
        }
        b := c + c + d;
        while (b >= p) b := b - p;
```

After this pseudocode sequence, the pair a, b has become in-place its correct, modulo-reduced base-p square. Note that there are no explicit divisions and there are six multiplies, though two of the multiplies involve multiplication of the reciprocal r, which is the size of p, by a number the size of p^2. Again, the magnitude of the speed advantage of (1.2) over a straightforward scheme is machine-dependent.

A third enhancement runs as follows. In Theorem 1, the term $(rx) >> s$ can be obtained using a nonstandard multiplication loop. Observe that, though x will be the size of N^2 (the steady-state denominator) and r the size of N, a count of s bits will be lost after the right-shift. This means that in the "grammar-school parallelogram" generated during long multiplication, many of the entries are meaningless, because they will be shifted into oblivion. By intervening in a detailed manner into the usual long multiplication loop, the value $(rx) >> s$ can be obtained in about half the time it takes to multiply two integers the size of N. We achieved in this way a complete base-p square-and-mod sequence of the form (1.2), that requires about five (size of p) multiply times.

The cumulative advantage of all these various enhancements, compared to the straigthtforward scheme and depending, of course, on the computing machinery used, was typically an order of magnitude in speed increase. Some timing details are given in §4.

The nature of the powering ladder arithmetic is not the only issue, because one wishes to test only actual primes p. The tempting Fermat test, that is whether $2^{p-1} \equiv 1 \pmod{p}$, is generally wasteful for a Wieferich search. For one thing, one may as well always do the $(\mathrm{mod}\ p^2)$ arithmetic, which of course contains the Fermat test. We found the most efficient scheme for isolation of testable primes to be an incremental sieve of Eratosthenes. If we are interested in all primes $p < L$, we segment all integers less than L into blocks of common size B, say $B = 10^6$. Then each new block is sieved completely, by all primes $< \sqrt{L}$, so that only primes remain within that block. At any time only one block equivalent of memory is involved, because for the current block we can quickly compute the sieving offsets for each sieving prime, then sieve rapidly. Our overall algorithm for the Wieferich prime search can be described thus:

ALGORITHM FOR WIEFERICH PRIME SEARCH

1) For search limit L, store all primes less than $\sqrt{L}$.
2) For each successive block of length B,
    3) Sieve out, using the stored primes, all composites from the current block,
    4) For each remaining prime p in the block,
        5) Choose $s > 2\log_2 p$, and compute the "reciprocal" $r = [2^s/p]$,
        6) Starting with representation $\{2, 0\}$ in a binary powering ladder, use possible machine-dependent enhancements (such as (1.1), (1.2), Theorem 1) to obtain a final base-p form $2^{(p-1)/2} \pmod{p^2} = \{C, D\}$.
        7) If C is neither 1 nor $p-1$, exit with fatal error. Otherwise, report any desired D-values. A Wieferich prime must have $\{C, D\} = \{1, 0\}$ or $\{p-1, p-1\}$.

In this way we eventually recorded all instances of

$$2^{(p-1)/2} \equiv \pm 1 + Ap \pmod{p^2},$$

TABLE 1. Instances of $2^{(p-1)/2} \equiv \pm 1 + Ap \pmod{p^2}$, with $|A| \leq 100$, for $10^9 < p < 4 \times 10^{12}$. The value $A = 0$ would signify a Wieferich prime

p	$\pm 1 + Ap$	p	$\pm 1 + Ap$
1222336487	$+1 + 60p$	36673326289	$+1 - 45p$
1259662487	$+1 - 71p$	46262476201	$+1 + 5p$
1274153897	$+1 - 86p$	47004625957	$-1 + 1p$
1494408397	$-1 + 52p$	49819566449	$+1 + 27p$
1584392531	$-1 - 24p$	53359191887	$+1 + 50p$
1586651309	$-1 - 24p$	58481216789	$-1 + 5p$
1662410923	$-1 - 70p$	76843523891	$-1 + 1p$
1817972423	$+1 - 56p$	82834772291	$-1 + 82p$
1890830857	$+1 + 69p$	108058158839	$+1 + 58p$
2062661389	$-1 + 55p$	130861186019	$-1 - 97p$
2244893621	$-1 + 47p$	138528575509	$-1 - 23p$
2332252547	$-1 - 33p$	239398882511	$+1 - 72p$
2416644757	$-1 + 67p$	252074060191	$+1 + 61p$
2461090421	$-1 + 47p$	252137567497	$+1 - 31p$
2566816313	$+1 + 52p$	299948374351	$+1 - 19p$
2570948153	$+1 - 41p$	405897532891	$-1 + 61p$
2589186937	$+1 - 85p$	443168739911	$+1 + 64p$
2709711233	$+1 + 50p$	504568016327	$+1 + 55p$
2760945133	$-1 - 77p$	703781283787	$-1 - 49p$
2954547209	$+1 + 32p$	955840782881	$+1 - 84p$
3027263587	$-1 - 95p$	980377925057	$+1 - 90p$
3133652447	$+1 + 87p$	981086885117	$-1 + 85p$
3303616961	$+1 - 20p$	1095406033573	$-1 + 42p$
3520624567	$+1 - 6p$	1104406423781	$-1 + 54p$
3606693551	$+1 + 21p$	1180032105761	$+1 - 6p$
4449676157	$-1 - 15p$	1722721869859	$-1 - 31p$
5045920247	$+1 + 76p$	1730418792409	$+1 - 46p$
5409537149	$-1 + 66p$	1780536689159	$+1 + 84p$
8843450093	$-1 - 20p$	2207775149407	$+1 - 99p$
10048450537	$+1 + 82p$	2424653846701	$-1 - 51p$
10329891503	$+1 + 79p$	2610372685663	$+1 - 28p$
11214704947	$-1 + 56p$	3667691800441	$+1 + 27p$
20051397221	$-1 - 46p$	3713054321579	$-1 - 51p$
20366156849	$+1 - 95p$	3729819224423	$+1 + 77p$

where A is allowed to be bipolar but $|A| \leq 100$, over the primes less than 4×10^{12}. Note that A is essentially the Fermat quotient; or more precisely,

$$q_p(2) \equiv \pm 2A \pmod{p}.$$

Table 1 shows all primes between 10^9 and 4×10^{12} that enjoy the small $|A|$ values. In §3 we discuss some statistical considerations pertinent to these and some analogous data for the Wilson quotients.

The large-integer arithmetic mentioned above is discussed in [9] and [10]. Except for some of our more obscure algorithmic enhancements, relevant code (`giants.`

[ch]) is stored on the disk supplement of [9]; it can also be obained over the World-Wide Web at `ftp://ftp.telospub.com` in the directory

`/pub/ScientificComputing/TopicsAdvSciComp/AppendixCode`.

The book [10] specifically deals with computational issues related to the Wieferich and Wilson searches.

2. Wilson primes

During factorial calculations all arithmetic (mod p^2) is subject to possible refinements of the previous section. In particular, base-p arithmetic is appropriate on most machines. But for the current range of interest, namely p on the order of 10^8, most modern machinery allows low-level arithmetic (mod p) or efficient long-long (double-precision) arithmetic, so the steady-state division of Theorem 1 need not apply (though that theorem becomes useful for much larger p). These arithmetic issues having been settled, we proceeded to analyze various theoretical relations with a view to minimization of computation time.

A useful and interesting identity is due to Granville [14]; for any integer m with $1 < m < p$, we have

$$\prod_{j=1}^{m-1} \binom{p-1}{[\frac{jp}{m}]} \equiv (-1)^{(p-1)(m-1)/2}(m^p - m + 1) \pmod{p^2}. \tag{2.1}$$

This congruence for $m = 2$ was known in the nineteenth century and has been rediscovered a few times since; for $m = 3,\ 4,\ 6$ the congruence follows from (50), (51), (52) in [17]. For example, (50) in [17] is the congruence

$$\binom{p-1}{[p/3]} \equiv (-1)^{[p/3]}(3^p - 1)/2 \pmod{p^2} \tag{2.2}$$

for all primes $p > 3$, which when squared gives (2.1) for $m = 3$.

Granville's identity for $m = 2$ shows right off that one never need evaluate the full factorial of $p - 1$, but may evaluate the factorial of $(p - 1)/2$ instead. A different kind of identity, when combined with (2.1), yields a considerable algorithm enhancement. Consider the representation of certain primes p by quadratic forms:

$$\begin{aligned}
&p \equiv 1 \pmod 4: && p = a^2 + b^2; && a \equiv 1 \pmod 4 \\
&p \equiv 1 \pmod 3: && 4p = c^2 + 27d^2; && c \equiv 1 \pmod 3 \\
&p \equiv 1 \pmod 3: && 4p = u^2 + 3v^2; && u \equiv (-1)^{(p-1)/6} \pmod 3, \\
& && && v \equiv 0 \pmod 6 \text{ if possible}, \\
& && && v \equiv \pm 1 \pmod 6 \text{ if not}.
\end{aligned} \tag{2.3}$$

It is known (see [3], [5], [11] , and also [7], [28]) that for $p \equiv 1 \pmod 4$,

$$\binom{\frac{p-1}{2}}{\frac{p-1}{4}} \equiv (2^{p-1} + 1)\left(a - \frac{p}{4a}\right) \pmod{p^2}, \tag{2.4}$$

and that for $p \equiv 1 \pmod 3$,

$$\binom{\frac{2p-2}{3}}{\frac{p-1}{3}} \equiv -c + \frac{p}{c} \pmod{p^2}, \quad \binom{\frac{p-1}{3}}{\frac{p-1}{6}} \equiv \frac{-u}{3}(2^p + 1) + \frac{p}{u} \pmod{p^2}. \tag{2.5}$$

The results (2.1)-(2.5) can be combined in various ways to yield the following identities, one of which is always applicable to a given odd prime p:

For $p \equiv 1 \pmod 3$, we use (2.2) and (2.5) to get

$$(2.6)\quad (p-1)! \equiv \left(\left(\frac{p-1}{6}\right)!\right)^6 (-u^3(2^p-1)+3pu)\left(-c+\frac{p}{c}\right)\frac{3^p-1}{2} \pmod{p^2}.$$

In (2.6) it is possible to ignore the condition on v in (2.3) since the different values of u produced have $-u^3(2^p-1)+3pu \equiv$ constant $\pmod{p^2}$.

For $p \equiv 5 \pmod{12}$, we use (2.1) with $m=2$ and (2.4) to get

$$(p-1)! \equiv \left(\left(\frac{p-1}{4}\right)!\right)^4 (2^p-1)(2^{p-1}+1)^2\left(a-\frac{p}{4a}\right)^2 \pmod{p^2},$$

so that,

$$(2.7)\qquad (p-1)! \equiv \left(\left(\frac{p-1}{4}\right)!\right)^4 (3\cdot 2^p-4)(2a^2-p) \pmod{p^2}.$$

For $p \equiv 11 \pmod{12}$, we use (2.1) with $m=2$ to get

$$(2.8)\qquad (p-1)! \equiv \left(\left(\frac{p-1}{2}\right)!\right)^2 (1-2^p) \pmod{p^2}.$$

So expensive are the relevant factorial calculations in practice, that one may with impunity ignore fast methods and obtain any of a, b, c, d, u, v by "brute force"; say, by looping through squares a^2, c^2, or u^2. Ignoring the relatively inconsequential operations of powering, inverting, and extraction of a, b, c, d, u, or v, the identities (2.6)-(2.8) involve, for $p \equiv 1 \pmod 3$, $p \equiv 5 \pmod{12}$, $p \equiv 11 \pmod{12}$, respectively, about

$$p/6,\ p/4,\ p/2$$

multiplies $\pmod{p^2}$, amounting to an average reduction, over many primes, of $48/13$ in complexity, as compared to the naive evaluation of $(p-1)!$. As we shall see presently, complexity reduction may be taken further.

An algebraic refinement is to observe that the multiplicands comprising an arbitrary $N!$ generally admit of a certain redundancy. For example, all the even multiplicands can be extracted and written in the form of some power of 2 times a smaller factorial. We start our analysis of what might be called factorial sieving, by defining a generalized factorial:

$$(2.9)\qquad P(q,m) = \prod_{\substack{k=1\\(k,q)=1}}^{m} k = \prod_{\substack{k=1\\(k,q)=1}}^{q-1} \prod_{j\equiv k \pmod q}^{m} j.$$

For example, $P(1,m) = m!$ and for m odd, $P(2,m) = m!!$, the product of the odd numbers up to m. Now our previous statement concerning the even multiplicands of $N!$, can be written, for N even, as

$$(2.10)\qquad N! = 2^{N/2} P(2,N)\left(\frac{N}{2}\right)!$$

From this identity $N!$ can actually be evaluated in about $3N/4$ multiplications. One multiplies together the even integers $\le N/2$, then the odd integers $\le N/2$, squares the latter; then multiplies everything by all odd integers in $(N/2, N]$. But

more can be done along these lines. For example, iterating (2.10) one gets the identity

$$N! = 2^{e_2} \prod_{j \geq 0} P\left(2, \left[\frac{N}{2^j}\right]\right), \tag{2.11}$$

where $2^{e_2} || N!$ and the apparent infinite product may be truncated as soon as $2^j > N/3$. The identity (2.11) allows the computation of $N!$ in $(1/2 + o(1))N$ multiplications.

There are some interesting theoretical branches to take at this point. One approach is to generalize (2.10) and (2.11) by sieving the factorial with, say, all primes $p \leq R$. This idea leads to the following general identity:

$$N! = \prod_{p \leq R} p^{e_p} \prod_{q \text{ is } R-\text{smooth}}^{N} P\left(\pi, \left[\frac{N}{q}\right]\right), \tag{2.12}$$

where $p^{e_p} || N!$, π is the product of all primes not exceeding R, and the q are R-smooth, i.e., not divisible by any prime exceeding R. We have found this formula difficult to use in practice. As just one extra complication, one must find all R-smooth numbers not exceeding N. However, this general factorial-sieve identity may find use in theoretical treatments of factorial complexity. For example, choosing $R = N^{1/\ln\ln N}$, it can be shown that (2.12) allows the computation of $N!$ in $O(N/\ln\ln N)$ multiplications.

A different approach is to partially recurse on some fixed set of sieve identities. By experimentation, the most pragmatic identity which we could find takes $R = 3$ in (2.12), but does not use the full recursion over all 3-smooth numbers. In particular, when $N \equiv 0 \pmod{6912}$ we have

$$N! = 2^{\frac{255N}{256}} 3^{\frac{185N}{432}} \left(\frac{N}{256}\right)! \prod_w P\left(2, \frac{N}{w}\right) \prod_v P\left(6, \frac{N}{v}\right), \tag{2.13}$$

where w runs through the set $\{16,18,24,27,32,36,64,128\}$ and v runs through the set $\{1,2,3,4,6,8,9,12\}$. Analysis of the P products reveals that, on the basis of (2.13), one may evaluate $N!$ by way of an asymptotic count of $(283/768)N$ multiplies.

Regardless of what factorial reduction formulae are in force, there is a powerful, universal enhancement that removes most of the multiplication in favor of addition. It may be observed that any product of interest in our previous identities can be cast as a product of the products of the terms in disjoint arithmetic progressions. For example, $P(q,m)$ by its definition (2.9) is manifestly such a product. Thus, if we can implement an algorithm for rapid evaluation of the product of terms in an arithmetic progression, we can call such a routine as desired. Happily there exists a suitable such algorithm, which will evaluate any product of n terms in arithmetic progression in $O(n^{\phi+\epsilon})$ multiplies, where $\phi = (\sqrt{5}-1)/2$ is the "golden ratio", and $n + O(n^{3-\sqrt{5}+\epsilon})$ adds. For the Wilson search, it is understood that mods must also be taken. However, since the number of adds will generally far exceed the number of multiplies, most of the mod operations involve only "if" statements and their ensuing subtractions.

An algorithm for evaluating a polynomial along arithmetic progression values is given in [15, p. 469], and uses the fact that if enough successive differences are taken, the difference tableau suffices to determine the required values of the

polynomial. We express here, for the purposes of the Wilson search, a particular variant that yields the product of all terms in arithmetic progression:

Algorithm to evaluate the product of terms in arithmetic progression

To evaluate

$$f(x) = x(x+d)(x+2d)\cdots(x+(n-1)d).$$

at a given point x:

1) Choose $G < n$ (an optimal choice of G is discussed later) and set $K := [\frac{n}{G}] - 1$.
2) Create a_0 through a_G:
   For $j = 0$ to G

$$a_j := \prod_{q=0}^{G-1} (x + (q + Gj)d)$$

3) Create difference tableau:

```
For q = 1 to G {
        For j = G down to q
                a_j := a_j - a_{j-1}
}
```

4) Manipulate differences and accumulate:

```
f := a_0
For j = 1 to K {
        a_0 := a_0 + a_1
        f := f a_0
        For q = 1 to G - 1
                a_q := a_q + a_{q+1}
}
```

5) Finish tail end of product:

$$f := f \prod_{j=G(K+1)}^{n-1} (x + jd).$$

For investigations such as the Wilson search, all sums, differences and products are to be reduced (mod p^2); whence the final output f will be the desired product of arithmetic progression terms, that is $f(x) \pmod{p^2}$. If base-p arithmetic is used, all of the arithmetic is to be performed in its natural way amongst representations; thus for example each of the initial a_j will be a representation pair.

Analysis shows that the algorithm requires $O(G^2 + n/G)$ multiplies and $n + O(G^2)$ adds. The optimal G is thus in the neighborhood of $n^{1/3}$, yielding $O(n^{2/3})$ multiplies and $n + O(n^{2/3})$ adds. But the algorithm's step (2) itself involves products of arithmetic progression terms. One may thus recurse on step (2), embedding the algorithm within itself to render ultimately the aforementioned operation count of $O(n^{\phi+\epsilon})$ multiplies and $n + O(n^{3-\sqrt{5}+\epsilon})$ adds. However, we found in practice that not even the first recursive level is really necessary. For primes $p \approx 10^8$, the

basic algorithm without recursion already has a total multiply count more than two orders of magnitude below the add count.

Armed with these various refinements, we arrived at the following practical algorithm:

ALGORITHM FOR WILSON PRIME SEARCH

1) For search limit L, store all primes less than $\sqrt{L}$.
2) For each successive block of length B,
   3) Sieve out, using the stored primes, all composites from the current block,
   4) For each remaining prime p in the block,
      5) As $p \equiv 1 \pmod 3$, $p \equiv 5 \pmod{12}$, $p \equiv 11 \pmod{12}$ adopt one of the identities (2.6)-(2.8). If p is not 11 (mod 12) obtain any required quadratic elements a, b, c, d, u, v.
      6) Use a factorial sieve such as (2.13) together with successive arithmetic progression products for the P product terms. In this way obtain a final base-p representation for $(p-1)! = \{C, D\}$.
      7) If C is not $p-1$, exit with fatal error. Otherwise, report any desired D-values. A Wilson prime must have $\{C, D\} = \{p-1, p-1\}$.

In this way we eventually recorded all instances of

$$(p-1)! \equiv (p-1) + p(p-1-B) \pmod{p^2},$$

where $0 \le B \le 100$, over the primes less than 5×10^8. Note that B is essentially the Wilson quotient; or more precisely,

$$w_p \equiv -B \pmod p.$$

Table 2 shows all primes between 10^7 and 5×10^8 that admit of such small B values.

Incidental to our search effort, we noted some alternative factorial evaluation schemes. Because of their theoretical attraction we shall briefly mention these alternatives, though for the p-ranges of the reported search we were unable to bring any of these alternatives up to the execution speed of the arithmetic progression algorithm written out above.

We implemented a recursive polynomial remaindering scheme for factorials; this method requiring no worse than $O(p^{1/2}(\ln p)^2)$ arithmetic operations to resolve the Wilson quotient. The idea is to evaluate a polynomial of degree m, namely:

$$f(x) = (x+1)(x+2)\cdots(x+m),$$

at the m points $x = 0, m, 2m, \ldots, (m-1)m$ and multiply together all of these evaluations to yield $(m^2)!$. This polynomial evaluation problem is known to require no more than $O(m(\ln m)^2)$ arithmetic operations [4], [20], [23]. For m a power of two, the recursion proceeds as follows. Define polynomials each of degree $m/2$:

$$g_0(x) = x(x-m)(x-2m)\cdots(x-(m/2-1)m),$$

$$g_1(x) = (x-(m/2)m)(x-(m/2+1)m)\cdots(x-(m-1)m).$$

Then it is immediate that $(m^2)!$ is the product of the evaluations of $f(x) \pmod{g_0(x)}$ at the zeros of g_0, times the product of the evaluations of $f(x) \pmod{g_1(x)}$ at the zeros of g_1. But each of the two reduced polynomials $f(x) \pmod{g_i(x)}$ can be evaluated in the same way, modulo appropriate members of a set of four degree-$m/4$ polynomials, and so on recursively. The recursion hits bottom at a chosen level

TABLE 2. Instances of $(p-1)! \equiv -1 - Bp \pmod{p^2}$, with $0 \leq B \leq 100$, for $10^7 < p < 5 \times 10^8$. The value $B = 0$ would signify a Wilson prime

p	$-1 - Bp$
10746881	$-1 - 7p$
11465149	$-1 - 62p$
11512541	$-1 - 26p$
11892977	$-1 - 7p$
12632117	$-1 - 27p$
12893203	$-1 - 53p$
19344553	$-1 - 93p$
21561013	$-1 - 90p$
27783521	$-1 - 51p$
39198017	$-1 - 7p$
45920923	$-1 - 63p$
53188379	$-1 - 54p$
56151923	$-1 - p$
57526411	$-1 - 66p$
72818227	$-1 - 27p$
87467099	$-1 - 2p$
91926437	$-1 - 32p$
93445061	$-1 - 30p$
93559087	$-1 - 3p$
94510219	$-1 - 69p$
101710369	$-1 - 70p$
117385529	$-1 - 43p$
212911781	$-1 - 92p$
216331463	$-1 - 36p$
327357841	$-1 - 62p$
411237857	$-1 - 84p$
479163953	$-1 - 50p$

for which direct evaluation, say by Horner's rule, of the current reduced polynomials proceeds efficiently.

In our implementation, we used polynomials of degree about $(p/k)^{1/2}$, where k is an appropriate integer gleaned from the reduction formulae (2.6)-(2.8). Each coefficient of each polynomial was a base-p representation. The polynomial remaindering used a Newton method for polynomial inversion, with large polynomial multiplication performed by way of a Nussbaumer convolution scheme in which signal array elements are base-p representations. The resulting experiments showed that, indeed, the arithmetic scheme is bested by this remaindering scheme for sufficiently large p. Our implementation of the remaindering scheme begins to be superior in speed for p in the neighborhood of 10^{11}. Though remaindering did not apply over our stated Wilson search region, we were able to use that scheme to resolve isolated, huge factorials. For example, for $p = 1099511628401$ $(= 2^{40} + 5^4)$, we calculated

$$(p-1)! \equiv -1 - 533091778023p \pmod{p^2}.$$

To our knowledge this is the largest prime for which the Wilson quotient is known (mod p); and also the largest integer to have been proven prime, as we have, on the basis of Lagrange's converse of Wilson's classical theorem. We believe that, given the computing machinery of today and the favorable asymptotic complexity of the remaindering algorithm, such calculations for p as high as 10^{20} are not out of the question.

Another scheme for Wilson testing was proposed to us by J. P. Buhler, and is one of two Wilson tests we noted that involve identities merely (mod p). For g a primitive root of an odd prime p, denote by $\{a_k, b_k\}$ the base-p representation of $g^k \pmod{p^2}$. Then

$$(p-1)! = \prod_{k=1}^{p-1} a_k \equiv \prod_{k=1}^{p-1} (g^k - pb_k) \pmod{p^2}$$

$$\equiv g^{\frac{p(p-1)}{2}} \left(1 - p\sum_{k=1}^{p-1} b_k a_{p-1-k}\right) \equiv -1 + p\sum_{k=1}^{p-1} b_k a_{p-1-k} \pmod{p^2},$$

which establishes

Theorem 2. *Let p be an odd prime possessed of a primitive root g. Then the Wilson quotient satisfies*

$$w_p \equiv \sum_{k=1}^{p-1} \left[\frac{g^k}{p}\right] g^{p-1-k} \pmod{p}.$$

It is interesting that the Fermat quotient $q_p(g)$ associated with the Wieferich problem appears in the theorem, in the guise of the final summand. At first glance the convolution in Theorem 2 appears to require at least $O(p)$ multiplies. This is not so in general; in fact we found means by which Theorem 2 may be applied with no multiplies, and $O(p \ln g)$ adds. The main idea is to think of the sum in the theorem as a polynomial in g, and invoke Horner's evaluation rule. We give here the surprisingly simple pseudocode loop in the case that $g = 2$ happens to be a primitive root:

```
(* Assume 2 is a primitive root of the prime p.*)
        a = 1;
        b = s = 0;
        loop{
            a := a + a;
            b := b + b;
            if(a >= p) {
                a := a - p;
                b := b + 1;
            }
            if(b >= p) b := b - p;
            s := s + s + b;
            while(s >= p) s := s - p;
            if(a == 1) break;
        }
(* p is a Wilson prime if and only if s = 0.  *)
```

Note that this pseudocode contains no multiplies and uses negligible memory. When 2 is not a primitive root, one may replace the various doubling steps in the loop with binary addition ladders to effect multiplication by $g > 2$. In spite of its extreme simplicity, this scheme still could not be made competitive with the arithmetic progression scheme. The basic reason for this failure is that the factorial schemes starting from (2.6)-(2.8) involve immediate reduction of the factorial term count, whereas we do not yet know any means by which the number of summands in Theorem 2 can be similarly reduced.

During attempts to reorder factorial products we found a second mere (mod p) algorithm, which we believe to be new, and which is expressed as follows.

Theorem 3. *For an odd prime p and any integer q with $1 < q < p$, denote by $(1/p)_q$ the unique inverse of p (mod q) lying in $[1, q-1]$. Then p is a Wilson prime if and only if*

$$\sum_{q=2}^{p-1} \left(\frac{1}{p}\right)_q \equiv -1 \pmod{p}.$$

This theorem follows immediately upon the observation that the product of terms $(p(1/p)_q - 1)/q$, over all q in the stated range, happens to be $(p-2)!$.

Theorem 3 has so far been difficult to render practical. However, if sufficient memory is available, one might consider storing tables of inverses (mod q) for many small primes q, and attempting to reconstruct the required terms $(1/p)_q$ rapidly. In addition, there is the possibility of parallelism. One could perhaps evaluate quickly sets of inverse terms for many primes p at once.

Finally, we mention the beautiful relation between Bernoulli numbers and the Wilson quotients due to N. G. W. H. Beeger [2]:

$$pB_{p-1} \equiv p - 1 + pw_p \pmod{p^2}.$$

It is interesting that Beeger actually used his congruence and a table of Bernoulli numbers to show that 5 and 13 are the only Wilson primes up to 113.

We have observed that the left-hand side of the Beeger congruence can be evaluated via polynomial algebra (mod p^2). The somewhat intricate polynomial manipulations can be summarized briefly, as follows. First, note that upon formal series expansion of

$$f(x) = \frac{2x \sinh x - x^2}{2(\cosh x - 1)}$$

in even powers of x, the coefficient of x^n is $(n+1)B_n/n!$. Define a coefficient T_{p-1} implicitly by

$$\frac{1 + \sum_{k=1}^{(p-3)/2} 2x^{2k}/(2k+1)!}{1 + \sum_{k=1}^{(p-3)/2} 2x^{2k}/(2k+2)!}$$

$$= 1 + \cdots + T_{p-1}x^{p-1} + O(x^{p+1}),$$

Note that the k-dependent factorials can all be inverted (mod p^2), so that T_{p-1} can be calculated unambiguously via polynomial division (mod p^2). Now, to obtain the desired coefficient $pB_{p-1}/(p-1)!$, which is not quite T_{p-1}, the latter must be corrected on the basis of the missing $(k = (p-1)/2)$ terms, the result being:

$$pB_{p-1} \equiv T_{p-1}(p-1)! + 2 - 2p \pmod{p^2}.$$

In this way Wilson quotients can be obtained via power series manipulation (mod p^2). We are aware that the multiplier $(p-1)!$ in this last relation gives the appearance of recourse, as do the factorials that enter into the evaluation of the hyperbolic polynomials. But we cannot yet rule out the possibility of a polynomial algorithm that does not involve recourse to factorials (for example, there are other, nonhyperbolic expansions involving Bernoulli numbers [10]). Such an algorithm might yield entirely new means for rapid evaluation of Wilson quotients.

In a converse spirit, we note that the Beeger congruence, together with Theorem 2, gives a clear, simple, multiply-free algorithm for evaluation of $pB_{p-1} \pmod{p^2}$. This algorithm involves $O(p \ln g)$ adds and, as we have seen, is especially efficient when $g = 2$ is the primitive root.

3. Statistical considerations

Are there infinitely many Wieferich primes, or *any* larger than 3511? Is 563 the last Wilson prime? Since $A = A(p)$ is an integer in the interval $(-p/2, p/2)$, one might view the "probability" of A assuming any particular value, say the value 0, to be $1/p$. And one might view experiments for different primes p as "independent" events. Thus, heuristically we might argue that the number of Wieferich primes in an interval $[x, y]$ is expected to be

$$\sum_{x \le p \le y} 1/p \approx \ln(\ln y / \ln x).$$

If this is the case, we would only expect to find one Wieferich prime above Lehmer's search bound of 6×10^9 and below 3.8×10^{26}. It is thus not too shocking that no new instances occur below 4×10^{12}. A similar heuristic suggests that there should be just one Wilson prime above the search limit 1.88×10^7 of [13] and below 5.9×10^{19}. Again it is no surprise that we did not find any new Wilson primes. (The "expected" number of Wieferich primes in our Wieferich search interval was about 0.25, and the "expected" number of Wilson primes in our Wilson search interval was about 0.18, so it was not *a priori* completely hopeless to embark upon such a search.)

However, we did find some "near" Wieferich and Wilson primes, where, respectively, the A-values and B-values are small. Tables 1 and 2 show all instances of small-magnitude even Fermat quotients, and small-negative Wilson quotients; more precisely, all occurrences of $0 \le |A|, B \le 100$ over the stated search regions. One might test the heuristic model above with the actual evidence. The "expected" number of primes p in $[10^9, 4 \times 10^{12}]$ with $|A| \le 100$ is $201 \ln(\ln(4 \times 10^{12})/\ln(10^9)) \approx 67.7$. In fact we found exactly 68 such near Wieferich primes p, a very close fit indeed. The "expected" number of primes p in $[10^7, 5 \times 10^8]$ with B-value satisfying $0 \le B \le 100$ is $101 \ln(\ln(5 \times 10^8)/\ln(10^7)) \approx 21.9$. In fact we found exactly 27 near Wilson primes in the interval.

4. Machine considerations

For our Wieferich and Wilson searches, we used various 32-bit processors, such as 68040, 486/586, HP7000, SPARC, and i860. (We inspected the software for the infamous, sometimes faulty "fdiv" instruction on 586 Pentium; and finding one occurrence of that instruction, we double checked all of the relevant primes.) Typically, every individual machine would handle all the primes in some particular region. We allowed each machine to set up its own, equivalent incremental

sieve over that machine's region of responsibility, since this was not too wasteful. Under the algorithm of §1, such 32-bit processors sustained Wieferich test rates of approximately 100 to 1000 primes per second per machine, in the $p \approx 10^{12}$ region. We have noted already that the straightforward scheme (size p^4 products and long divide) to obtain Fermat quotients runs about ten times slower. With the aid of Richard McIntosh we tested the base-p arithmetic on a 64-bit (DEC) machine that allowed low-level divide/mod, finding that such a machine can sustain close to 10,000 primes per second in the stated region. Bailey's computations, as mentioned in §1, were performed on an IBM SP-2 parallel computer at NASA Ames Research Center, utilizing otherwise idle machine cycles. Each of his one hundred individual machines worked with its own set of primes, performing arithmetic in our base-p fashion. For the base-p arithmetic, Bailey used floating-point double precision (64-bit) arithmetic, including clever use of "floating multiply-add" operations, which on the IBM RS6000/590 processors maintain internal 106-bit accuracy. The resulting code achieved an impressive sustained testing rate of over one million primes per second.

The Wilson search proceeded, after all the refinements noted in the arithmetic progression based algorithm, at processor-dependent rates between 1/20 and 1/2 prime per second, in the $p \approx 10^8$ region. That the fastest 32-bit machines would require several seconds to resolve the Wilson quotient for a single p near 100 million is not unimpressive when one considers that the straightforward multiply-accumulate-(mod p^2) method, with multi-precision division also assumed, would require the equivalent of perhaps 10^9 size p multiplies. In fact the straightforward method is roughly 100 times slower than the rate we achieved. The polynomial remaindering scheme, which dominates for $p > 10^{11}$, is roughly ten times faster than our arithmetic progression based scheme for p in the region of 10^{12}. Nevertheless, using polynomial remaindering, it still took a SPARC processor about one day to resolve the Wilson quotient of $p = 1099511628401$ (see §3).

5. Related searches

In the absence of future theoretical results, one is moved to attach the probability $1/p$ also to certain other properties (mod p^2). What might be called Wall-Sun-Sun primes (see [25] and [24]) are those $p > 5$ satisfying

$$F_{p-\left(\frac{p}{5}\right)} \equiv 0 \pmod{p^2}, \tag{4.1}$$

where F_n is the nth Fibonacci number. This congruence is, again, always satisfied merely (mod p). Williams [27] found no Wall-Sun-Sun primes whatsoever, below 10^9, and Montgomery [19] extended this to 2^{32}. On the basis of a search being conducted by R. McIntosh [18], we have learned that there exist no Wall-Sun-Sun primes less than 2×10^{12}. Beyond the $1/p$ statistics, it is of interest that, as Sun and Sun [24] proved, if p is a failing exponent in FLT(I), then p satisfies (4.1).

There are Wieferich composites, namely composite integers N such that

$$2^{\varphi(N)} \equiv 1 \pmod{N^2},$$

where φ is Euler's function, see [1]. (For N odd, the left side is always 1 (mod N).)

For example, 3279 ($= 3 \times 1093$) is such a number N. This congruence is one way to generalize the Wieferich prime definition. Another is the weaker condition that $(N, (2^l - 1)/N) > 1$, where l is the least positive integer with N dividing $2^l - 1$. For this latter generalization it has been shown in [12], perhaps surprisingly, that asymptotically all odd numbers N are Wieferich.

Finally, there are the Wilson composites, being composite N such that

$$P(N,N) = \prod_{\substack{j=1 \\ (j,N)=1}}^{N} j \equiv \pm 1 \pmod{N^2}.$$

This congruence is one way to generalize the Wilson prime definition. Analysis and search is underway for such composites [1]. We were able to offer aid to that project, by using the factorial sieve and arithmetic progression based algorithms of §2 to calculate $P(N,N)$ rapidly, finding the new instances $N = 558771$, 1964215, 8121909 and 12326713. For example, the product of all positive integers less than and coprime to 12326713 is 1 (mod 12326713^2).

Acknowledgments

We wish to thank D. Bailey, J. P. Buhler, T. E. Crandall, R. Evans, A. Granville, and H. W. Lenstra, Jr., for valuable discussions pertaining to this work. Special thanks to Thomas Trappenberg (Dalhousie University and Optimax Software) whose generous contributions of personal time and computing power are responsible for a great deal of the entire Wieferich search; and to Richard McIntosh for his aid at many levels to the search project.

References

1. T. Agoh, K. Dilcher and L. Skula, *Fermat and Wilson quotients for composite moduli*, Preprint (1995).
2. N. G. W. H. Beeger, *Quelques remarques sur les congruences* $r^{p-1} \equiv 1 \pmod{p^2}$ *et* $(p-1)! \equiv -1 \pmod{p^2}$, The Messenger of Mathematics **43** (1913–1914), 72–84.
3. B. Berndt, R. Evans and K. Williams, *Gauss and Jacobi sums*, Wiley-Interscience, to appear.
4. J. M. Borwein and P. B. Borwein, *Pi and the AGM*, John Wiley & Sons, Inc., 1987. MR **89a:**11134
5. S. Chowla, B. Dwork, and R. Evans, *On the mod* p^2 *determination of* $\binom{(p-1)/2}{(p-1)/4}$, J. Number Theory **24** (1986), 188–196. MR **88a:**11130
6. D. Clark, *Private communication.*
7. M. Coster, *Generalisation of a congruence of Gauss*, J. Number Theory **29** (1988), 300– 310. MR **89i:**11005
8. R. Crandall and B. Fagin, *Discrete weighted transforms and large-integer arithmetic*, Math. Comp. **62** (1994), 305–324. MR **94c:**11123
9. R. Crandall, *Projects in Scientific Computation*, TELOS/Springer-Verlag, Santa Clara, CA, 1994. MR **95d:**65001
10. ———, *Topics in Advanced Scientific Computation*, TELOS/Springer-Verlag, Santa Clara, CA, 1995.
11. R. Evans, *Congruences for binomial coefficients*, Unpublished manuscript (1985).
12. Z. Franco and C. Pomerance, *On a conjecture of Crandall concerning the* $qx + 1$ *problem*, Math. Comp. **64** (1995), 1333–1336. MR **95j:**11019
13. R. H. Gonter and E. G. Kundert, *All prime numbers up to 18,876,041 have been tested without finding a new Wilson prime*, Preprint (1994).
14. A. Granville, *Binomial coefficients modulo prime powers*, Preprint.
15. D. E. Knuth, *The art of computer programming, Vol.2*, Addison-Wesley, Reading, Massachusetts, 2nd ed. 1973. MR **44:**3531

16. D. H. Lehmer, *On Fermat's quotient, base two*, Math. Comp. **36** (1981), 289–290. MR **82e:**10004
17. E. Lehmer, *On congruences involving Bernoulli numbers and the quotients of Fermat and Wilson*, Ann. of Math. **39** (1938), 350–360.
18. R. McIntosh, *Private communication.*
19. P. Montgomery, *New solutions of* $a^{p-1} \equiv 1 \pmod{p^2}$, Math. Comp. **61** (1991), 361–363. MR **94d:**11003
20. J. Pollard, *Theorems on factorization and primality testing*, Proc. Cambridge Phil. Soc. **76** (1974), 521–528. MR **50:**6992
21. P. Ribenboim, *13 lectures on Fermat's last theorem*, Springer-Verlag, New York, 1979. MR **81f:**10023
22. ———, *The book of prime number records*, Springer-Verlag, New York, 1988. MR **89e:**11052
23. V. Strassen, *Einige Resultate über Berechnungskomplexität*, Jahresber. Deutsch. Math.-Verein. **78** (1976/77), 1–8. MR **55:**11713
24. Zhi-Hong Sun and Zhi-Wei Sun, *Fibonacci numbers and Fermat's last theorem*, Acta Arith. **60** (1992), 371–388. MR **93e:**11025
25. D. D. Wall, *Fibonacci series modulo m*, Amer. Math. Monthly **67** (1960), 525–532. MR **22:**10945
26. A. Wieferich, *Zum letzten Fermat'schen Theorem*, J. Reine Angew. Math. **136** (1909), 293–302.
27. H. C. Williams, *The influence of computers in the development of number theory*, Comput. Math. Appl. **8** (1982), 75–93. MR **83c:**10002
28. K. M. Yeung, *On congruences for binomial coefficients*, J. Number Theory **33** (1989), 1–17. MR **90i:**11143

Center for Advanced Computation, Reed College, Portland, Oregon 97202
E-mail address: crandall@reed.edu

Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, Nova Scotia, B3H 3J5, Canada
E-mail address: dilcher@cs.dal.ca

Department of Mathematics, University of Georgia, Athens, Georgia 30602
E-mail address: carl@ada.math.uga.edu