The factorization of the ninth Fermat number

A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard

Abstract. In this paper the full prime factorization of the ninth Fermat number $F_9=2^{512}+1$ is exhibited. It is the product of three prime factors that have 7, 49 and 99 decimal digits. The latter two factors were found by means of the number field sieve, which is a factoring algorithm that depends on arithmetic in an algebraic number field. In the present case, the number field $\mathbf{Q}(\sqrt[5]{2})$ was used. The calculations were done on approximately 700 workstations scattered around the world, and in one of the final stages a supercomputer was used. The entire factorization was accomplished in four months.

Key words: Fermat number, factoring algorithm.

1980 Mathematics subject classification (1985): 11Y05, 11Y40.

1. Introduction.

For a non-negative integer k, the *kth Fermat number* F_k is defined by $F_k = 2^{2^k} + 1$. The ninth Fermat number $F_9 = 2^{512} + 1$ has 155 decimal digits:

$$\begin{aligned} F_9 = {} & 13407\,807929\,942597\,099574\,024998\,205846\,127479\,365820\,592393 \\ & 377723\,561443\,721764\,030073\,546976\,801874\,298166\,903427\,690031 \\ & 858186\,486050\,853753\,882811\,946569\,946433\,649006\,084097. \end{aligned}$$

It is the product of three prime numbers:

$$F_9 = p_7 \cdot p_{49} \cdot p_{99},$$

where p_7, p_{49}, p_{99} have 7, 49, 99 decimal digits, respectively:

$$\begin{aligned} p_7 = {} & 2\,424833, \\ p_{49} = {} & 7\,455602\,825647\,884208\,337395\,736200\,454918\,783366\,342657, \\ p_{99} = {} & 741\,640062\,627530\,801524\,787141\,901937\,474059\,940781\,097519 \\ & 023905\,821316\,144415\,759504\,705008\,092818\,711693\,940737. \end{aligned}$$

In binary, p_7, p_{49}, p_{99} have 22, 163, 329 digits, respectively:

$$
\begin{aligned}
p_7 = {} & 1001\,010000\,000000\,000001,\\
p_{49} = {} & 1\,010001\,100111\,110000\,110010\,110001\,010011\,001111\,001101\\
& 101100\,111111\,001101\,101001\,111101\,000010\,001111\,101010\,110010\\
& 101101\,010111\,100000\,110001\,010011\,001001\,010101\,000010\,100000\\
& 000001,\\
p_{99} = {} & 10101\,101100\,110110\,001111\,010110\,100000\,010011\,100101\,010000\\
& 101110\,011110\,100011\,001010\,111000\,110001\,111001\,100101\,110011\\
& 010011\,000110\,111110\,011000\,100110\,010101\,001011\,000101\,100110\\
& 011110\,000110\,110010\,000110\,111011\,001010\,010110\,001100\,001011\\
& 111111\,111001\,001000\,101010\,101001\,111010\,100011\,001001\,111010\\
& 010100\,000000\,101101\,101010\,111001\,000100\,110001\,101101\,100000\\
& 000001.
\end{aligned}
$$

The binary representation of F_9 itself consists of 511 zeroes surrounded by 2 ones.

In this paper we explain how the above factorization was found, and why.

Fermat numbers were first considered in 1640 by the French mathematician Pierre de Fermat (1601–1665), whose interest in the problem of factoring integers of the form $2^m \pm 1$ arose from their connection with 'perfect', 'amicable' and 'submultiple' numbers [37; 38, Chap. II, § IV]. He remarked that a number of the form $2^m + 1$, $m \in \mathbf{Z}_{>0}$, can only be prime if m is a power of 2, in which case $2^m + 1$ is a Fermat number; such primes are called *Fermat primes*. Fermat repeatedly expressed his strong belief that, conversely, all Fermat numbers are prime. Apparently, this belief was based on his observation that each prime divisor p of F_k must satisfy a strong condition, namely $p \equiv 1 \bmod 2^{k+1}$. In present day language, one would formulate his proof of this by saying that if $2^{2^k} \equiv -1 \bmod p$, then $(2 \bmod p)$ has multiplicative order 2^{k+1}, and so 2^{k+1} divides $p - 1$, by Fermat's own "little" theorem, which also dates from 1640. It is not clear whether Fermat was aware of the stronger condition $p \equiv 1 \bmod 2^{k+2}$ for prime divisors p of F_k, $k \geq 2$. It is most easily proved by replacing, in the above argument, $(2 \bmod p)$ by its square root $(2^{3 \cdot 2^{k-2}} - 2^{2^{k-2}} \bmod p)$ or by $(F_{k-1} \bmod p)$, which have order 2^{k+2}. Incidentally, from the binary representations of the prime factors of F_9 we see that

$$\mathrm{ord}_2(p_7 - 1) = 16, \qquad \mathrm{ord}_2(p_{49} - 1) = 11, \qquad \mathrm{ord}_2(p_{99} - 1) = 11,$$

where ord_2 counts the number of factors 2.

The first five Fermat numbers $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65537$ are indeed prime, but to this day these remain the only known Fermat primes. Nowadays it is considered more likely, on loose probabilistic grounds, that there are only finitely many Fermat primes. It may well be that F_0, F_1, F_2, F_3, F_4 are the only ones. Similarly, one may speculate that all Fermat numbers are squarefree, with perhaps finitely many exceptions.

As for F_5, Fermat knew that any prime divisor of F_5 must be among 193, 449, 577, 641, 769, ..., which is the sequence of primes that are 1 mod 2^6, with $F_3 = 257$ omitted (distinct Fermat numbers are clearly relatively prime). Thus it is difficult to understand how he missed the factor 641, which is only the fourth one to try; among those that are 1 mod 2^7, it is the first! One is led to believe that Fermat did not seriously attempt to verify his conjecture numerically, or that he made a computational error if he did. The factor 641 of F_5 was found by Euler in 1732, thereby refuting Fermat's conjecture [13]. The cofactor $F_5/641 = 6700417$ is also prime.

Gauss showed in 1801 that Fermat primes are of importance in elementary geometry: a regular n-gon can be constructed by ruler and compasses if and only if n is the product of a power of 2 and a set of distinct Fermat primes [14].

Since the second half of the nineteenth century, many mathematicians have been intrigued by the problem of finding prime factors of Fermat numbers and, more generally, numbers of the form $2^m \pm 1$. Somewhat later, this interest was extended to the larger class of "Cunningham numbers" $b^m \pm 1$ (with b small and m large) [11; 7]. The best factoring algorithms that were available were usually applied to these numbers, so that the progress that was made in the general area of factoring large integers was reflected in the factorization of Fermat and Cunningham numbers. It must be kept in mind that each Fermat number has only half as many digits as the next one (rounded upwards), so that the effort required for the complete prime factorization of a Fermat number may be expected to be substantially larger than for the preceding one; in several cases it could only be accomplished by means of a newly invented method. In 1880, Landry and Le Lasseur found the prime factorization of F_6, using a never-published method [19; 12, Chap. XV, p. 377; 15]. In 1970, Morrison and Brillhart factored F_7 with the continued fraction method [28]. Brent and the fourth author factored F_8 in 1980 by means of a modified version of Pollard's rho method [6]. In 1988, Brent used the elliptic curve method to factor F_{11} (see [4; 5]). Finally, F_9 was factored in 1990 by means of the *number field sieve*.

Unlike previous methods, the number field sieve is far more effective on Fermat and Cunningham numbers than on general numbers. Factoring arbitrary numbers of the order of magnitude of F_9 with the number field sieve, or with any other known method, currently requires substantially more time and financial resources than were spent on F_9; factoring arbitrary numbers of the order of magnitude of $10^{15}F_9$ is not yet practically feasible. Due to this development, Fermat and Cunningham numbers are losing their value as a yardstick to measure progress in factoring. One wonders which class of numbers will take their place.

It is hard to choose good numbers to test factoring algorithms. They should be defined *a priori*, to avoid the impression that the factored numbers were generated by multiplying known factors. They should be easy to compute. They should not have known arithmetic properties that might be exploited by a special factorization algorithm. For any size range, there should be enough test numbers so that one does not quickly run out, but few enough to spark competition for them. They should have some mathematical significance, so that factoring them is a respectable activity. The last condition is perhaps a controversial one; but do we want to factor numbers that are obtained from a pseudo-random number generator, or from the digits of π (see [2; 35])? The values of the partition function [1] meet the above conditions reasonably well. We offer them to future factorers as test numbers. Nonetheless, factoring Fermat numbers remains difficult and challenging, and it is likely to exercise a special fascination for a long time to come.

In addition to the more-or-less general methods mentioned above, a very special method has been used to search for factors of Fermat numbers. It proceeds not by fixing k and searching for numbers p dividing F_k, but by fixing p and searching for numbers k with $F_k \equiv 0 \bmod p$. To do this, one first chooses a number $p = u \cdot 2^l + 1$, with u

odd and l relatively large, that is free of small prime factors; this can be done by fixing one of u, l and sieving over the other. Next one determines, by repeated squarings (mod p), the residue classes $2^{2^k} \bmod p$, $k = 2, 3, \ldots$. If no value $k \leq l-2$ is found with $2^{2^k} \equiv -1 \bmod p$, then p does not divide any F_k, $k \geq 2$, and p is discarded. If a value of k is found with $2^{2^k} \equiv -1 \bmod p$—which one expects, loosely, to happen with probability $1/u$, if p is prime—then p is a factor of F_k. The primality of p is then usually automatic from knowledge that one may have about smaller prime factors of F_k or, if p is sufficiently small, from the fact that all its divisors must be $1 \bmod 2^{k+2}$. Many factors of Fermat numbers have been found this way. In 1903, A.E. Western [10] found the prime factor $p_7 = 2424833 = 37 \cdot 2^{16} + 1$ of F_9. In 1984, Keller found the prime factor $5 \cdot 2^{23473} + 1$ of F_{23471}; the latter number is the largest Fermat number known to be composite.

If no factor of F_k can be found, one can apply a primality test that is essentially due to Pepin [29]: for $k \geq 1$, the number F_k is prime if and only if $3^{(F_k-1)/2} \equiv -1 \bmod F_k$. This congruence can be checked in time $O((\log F_k)^3)$. One should not view Pepin's test as a polynomial time algorithm, however; since k is the input, the time that it takes is *doubly exponential*, and indeed the test has only been applied for a very limited collection of values of k. If F_k has known factors, one can investigate the factors using primality tests for general numbers. Thus, Brillhart [16, p. 110] found in 1967 that the number $F_9/2424833$, which has 148 decimal digits, is composite. In 1988 Brent and Morain found that F_{11} divided by the product of four relatively small prime factors is a prime number of 564 decimal digits, thereby completing the prime factorization of F_{11}.

The many results that have been obtained by the above methods, as well as bibliographic information, can be found in [12, Chap. XV; 11; 7; 33; 17]. For up-to-date information one should consult the current issues of *Mathematics of Computation*, as well as the updates to [7] that are regularly published by S.S. Wagstaff, Jr. (Department of Computer Science, Purdue University). We give a brief summary of the present state of affairs.

The complete prime factorization of F_k is known for $k \leq 9$, for $k = 11$, and for no other k. One or more prime factors of F_k are known for all $k \leq 32$ except $k = 14$, 20, 22, 24, 28, 31, as well as for 76 larger values of k, the largest being $k = 23471$. For $k = 10$, 12, 13, 15, 16, 17, 18 the cofactor is known to be composite. No non-trivial factor is known of F_{14} or F_{20}, but it is known that these numbers are composite. For $k = 22, 24, 28, 31$ and all except 76 values of $k > 32$ it is unknown whether F_k is prime or composite.

The smallest Fermat number that has not been completely factored is F_{10}. Its known prime factors are

$$11131 \cdot 2^{12} + 1 = 45\,592577,$$

$$395937 \cdot 2^{14} + 1 = 6487\,031809.$$

The cofactor has 291 decimal digits. Unless it has a relatively small factor, it is not likely to be factored soon.

The factorization of Fermat numbers is of possible interest in the theory of finite fields. Let m be a non-negative integer, and let the field K be obtained by m successive quadratic extensions of the field $\mathbf{F}_2 = \mathbf{Z}/2\mathbf{Z}$, so that $\#K = 2^{2^m}$; an elegant explicit description of K was given by Conway [9, Chap. 6] and another by Wiedemann [39]. It is not difficult to prove that there is an isomorphism

$$K^* \cong \bigoplus_{k=0}^{m-1} \mathbf{Z}/F_k\mathbf{Z}$$

of abelian groups, where K^* denotes the multiplicative group of K. Therefore knowledge of the prime factors of Fermat numbers is useful if one wishes to determine the multiplicative order of a given element of K^* or if one searches for a primitive root of K.

Some notation: $\mathbf{Q}$ is the field of rational numbers; $\mathbf{F}_p$, for a prime number p, is the field $\mathbf{Z}/p\mathbf{Z}$; and R^* is the group of units of a ring R with 1.

2. Factoring integers.

In this section n is an odd integer greater than 1. It should be thought of as an integer that we want to factor into primes.

Many factoring algorithms depend on the elementary fact that the subgroup

$$\{x \in \mathbf{Z}/n\mathbf{Z} : x^2 = 1\}$$

of $(\mathbf{Z}/n\mathbf{Z})^*$ has, as a vector space over $\mathbf{F}_2$, dimension equal to the number of distinct prime factors of n. Hence, if n is not a power of a prime number, then there is an element $x \in \mathbf{Z}/n\mathbf{Z}$, $x \neq \pm 1$, such that $x^2 = 1$. Moreover, explicit knowledge of such an element x, say $x = (y \bmod n)$, leads to a non-trivial factorization of n. Namely, from $y^2 \equiv 1 \bmod n$, $y \not\equiv \pm 1 \bmod n$, it follows that n divides the product of $y - 1$ and $y + 1$ without dividing the factors, so that $\gcd(y - 1, n)$ and $\gcd(y + 1, n)$ are non-trivial divisors of n. They are in fact complementary divisors, so that only one of the gcd's needs to be calculated; this can be done with Euclid's algorithm. We conclude that, to factor n, it suffices to find $x \in \mathbf{Z}/n\mathbf{Z}$ with $x^2 = 1$, $x \neq \pm 1$.

This procedure will fail if n is a prime power, so it is wise to first make sure that this is not the case. One can do this by first subjecting n to a primality test, as in [21, section 5]. If n is prime, the factorization is finished. Suppose that n is not prime. One still needs to check that n is not a prime *power*. This check is often omitted, since in many cases it is considered highly unlikely that n is a prime power if it isn't prime; it may even be considered highly likely that n is squarefree, i. e., not divisible by the square of a prime number. For example, suppose that n is the unfactored portion of some randomly drawn integer, and one is certain that it has no prime factors below a certain bound B. Then the probability for n not to be squarefree is $O(1/(B \log B))$, in a sense that can be made precise, and the probability that n is a proper power of a prime number is even smaller. A similar statement may be true if n is the unfactored portion of a Cunningham number, since, to our knowledge, no such number has been found to be divisible by the square of a prime factor that was difficult to find. Whether other classes of test numbers that one may propose behave similarly remains to be seen; if the number n to be factored is provided by a 'friend', or by a colleague who does not yet have sufficient understanding of the arithmetical properties of the numbers that his computations produce, one can not assume that it is likely that n is squarefree or not a prime power.

At the moment, there are no squarefreeness tests that are essentially faster than factoring. This is often contrasted with the case of polynomials in one variable over a field K, in which case it suffices to take the gcd with the derivative. This illustrates that for many algorithmic questions the well-known analogy between $\mathbf{Z}$ and $K[X]$ appears to break down. Note also that for many fields K, including finite fields and algebraic number fields, there exist excellent practical factoring algorithms for $K[X]$ (see [20]), which have no known analogue in $\mathbf{Z}$.

There do exist factoring methods that become a little faster if one only wishes to test squarefreeness; for example, if n is not a square—which can easily be tested—then to determine whether or not n is squarefree it suffices to do a trial division up to $n^{1/3}$ instead of $n^{1/2}$. There is also a factoring method that has great difficulties with numbers n that are not squarefree. Suppose, for example, that p is a large prime for which $p-1$ and $p+1$ both have a large prime factor, and that n has exactly two factors of p. The factoring method described in [34], which depends on the use of "random class groups", and which is also known as the Shanks-Pollard-Atkin-Rickert algorithm, does not have a reasonable chance of finding *any* non-trivial factor of n, at least not within the time that is conjectured in [34], see [25].

Fortunately, the situation is much better if one only wants to know that n is not a proper prime power. One way to proceed is by testing that n is not a proper power. Namely, if $n = m^l$, where m, l are integers and $l > 1$, then $m \geq 3$, $2 \leq l \leq [(\log n)/\log 3]$, and one may assume that l is prime. Hence the number of values to be considered for l is quite small, and this number can be further reduced given a better lower bound for m, such as a number B as above. For each value of l, calculate an integer m' for which $|m' - n^{1/l}| < 1$, using Newton's method. Next one tests whether $n = m'^l$; this is the case if and only if n is an lth power. One can often save time by doing the calculation of m' only if n satisfies the condition

$$n^{l-1} \equiv 1 \bmod l^2 \qquad (\bmod 8 \quad \text{if } l = 2),$$

$$n^{(q-1)/l} \equiv 1 \bmod q$$

for several small primes q with $q \equiv 1 \bmod l$, which is a necessary condition for a number n that is free of small prime factors to be an lth power, if l is prime.

There is a second, less well-known way to proceed, which only tests that n is not a *prime* power. It assumes that one has already proved that n is composite. Often, one has employed Fermat's theorem for this purpose, which states that $a^n \equiv a \bmod n$ for every integer a, if n is prime. Hence, if an integer a has been found for which $a^n \not\equiv a \bmod n$, then one is sure that n is not a prime number. To be sure that n is not a prime *power*, it is enough to show that the element $(a^n - a \bmod n) \in \mathbf{Z}/n\mathbf{Z}$, which one has already calculated, and which is not zero, is actually a *unit*; for this it suffices to calculate a single gcd, namely that of $a^n - a$ (taken modulo n) with n. If the gcd is 1, then n is not a prime power, and if the gcd is not 1 then it is a non-trivial factor of n, which is usually more valuable than the information that n is or is not a prime power. The correctness of this test follows immediately from the fact that $a^{p^k} \equiv a \bmod p$ for any integer a, any integer $k \geq 0$, and any prime number p; this is proved by induction on k with Fermat's theorem.

Nowadays one often uses a variant of Fermat's theorem, which depends on the splitting

$$a^n - a = a \cdot (a^u - 1) \cdot \prod_{i=0}^{t-1} (a^{u \cdot 2^i} + 1),$$

where $n - 1 = u \cdot 2^t$, with u odd and $t = \mathrm{ord}_2(n-1)$. Hence, if n is prime, then for any integer a one of the $t + 2$ factors on the right is divisible by n. This variant has the advantage that the converse is true, in a strong sense: if n is not prime, then *most* integers a have the property that *none* of the factors on the right is $0 \bmod n$ (see [32] for a precise statement and proof); such integers a are called *witnesses* to the compositeness of n. Currently, if one is sure that the number n to be factored is composite, it is usually because one found such a witness. A witness a can, just as above, be used to check that n is in fact not a prime power: calculate $a^n - a$ (mod n), which is most easily done by first squaring the number $a^{u \cdot 2^{t-1}}$ (mod n) that was last calculated; if it is non-zero, one verifies as before that $\gcd(a^n - a, n) = 1$, and if it is zero then one of the $t + 2$ factors on the right has a non-trivial factor in common with n, which can readily be found. (It may be noted that in the latter case n is not a prime power, since the odd parts of the $t + 2$ factors are pairwise relatively prime.)

As we mentioned in the introduction, the number $F_9/2424833$ was proved to be composite by Brillhart in 1967. We do not know whether he or anybody else proved that it is not a prime power until this fact became plain from its prime factorization. We didn't, not because we thought it wasn't worth our time, but simply because we didn't think of it. If it *had* been a prime power, our method would have gotten nowhere, and we would have felt greatly embarrassed towards the many people who helped us in this project. One may believe that the risk that we were unconsciously taking was extremely small, but until the number was factored this was indeed nothing more than a belief. In any case, it would be wise to include, in the witness test described above, the few extra lines that prove that a number is not a prime power, and to explicitly publish this information about a number rather than just saying that it is composite.

For the rest of this section we assume that n, besides being odd and > 1, is not a prime power. We wish to factor n into primes. As we have seen, each $x \in \mathbf{Z}/n\mathbf{Z}$ with $x^2 = 1$, $x \neq \pm 1$ gives rise to a non-trivial factor of n. In fact, it is not difficult to see that the full factorization of n into powers of distinct prime numbers can be obtained from a set of generators of the $\mathbf{F}_2$-vector space $\{x \in \mathbf{Z}/n\mathbf{Z} : x^2 = 1\}$. (If we make this vector space into a *Boolean ring* with $x * y = (1 + x + y - xy)/2$ as multiplication, then a set of ring generators also suffices.) The question is how to determine such a set of generators. Several algorithms have been proposed to do this, most of them following some refinement of the following scheme.

Step 1. *Selecting the factor base.* Select a collection of non-zero elements $a_i \in \mathbf{Z}/n\mathbf{Z}$, with i ranging over some finite index set I. *How* this selection takes place depends on the particular algorithm; it is usually not done randomly, but in such a way that Step 2 below can be performed in an efficient manner. The collection $(a_i)_{i \in I}$ is called the *factor base*. We shall assume that all a_i are *units* of $\mathbf{Z}/n\mathbf{Z}$. In practice, this is likely to be true, since if n is difficult to factor one doesn't expect one of its prime factors to show up in one of the a_i's; one can verify the assumption, or find a non-trivial factor of n, by means of a few gcd calculations. Denote by $\mathbf{Z}^I$ the free abelian group on generators e_i, $i \in I$, and let $f\colon \mathbf{Z}^I \to (\mathbf{Z}/n\mathbf{Z})^*$ be the group homomorphism (from an additively to a multiplicatively written group) that sends e_i to a_i, for $i \in I$. This map is surjective if and only if the elements a_i generate $(\mathbf{Z}/n\mathbf{Z})^*$. For the choices of a_i that are made in practice that is usually the case, although we are currently unable to prove this. (In general, hardly anything has been rigorously proved about practical factoring algorithms.)

Step 2. *Collecting relations.* Each element $v = (v_i)_{i\in I}$ of the kernel $\ker f$ of f is a *relation* between the a_i, in the sense that

$$\prod_{i\in I} a_i^{v_i} = 1.$$

In the second step, one looks for such relations by a method that depends on the algorithm. One stops as soon as the collection V of relations that have been found has slightly more than $\#I$ elements. One hopes that V generates $\ker f$, although this is again typically beyond proof. Note that $\ker f$ is of finite index in $\mathbf{Z}^I$, so that by a well-known theorem from algebra it is freely generated by $\#I$ elements; therefore the hope is not entirely unreasonable.

Step 3. *Finding dependencies.* For each $v \in V$, let $\bar{v} \in (\mathbf{Z}/2\mathbf{Z})^I = \mathbf{F}_2^I$ be the vector that is obtained from v by reducing its coordinates modulo 2. Since $\#V > \#I$, the vectors $\bar{v}$ are linearly dependent over $\mathbf{F}_2$. In Step 3, one finds explicit dependence relations. Although the matrices that one encounters in practice are *huge* and *sparse*, for which special methods exist [18], one usually employs ordinary Gaussian elimination. The size of the matrices may make it desirable to modify Gaussian elimination somewhat; see section 4. Each dependence relation that is found can be written in the form $\sum_{v\in W} \bar{v} = 0$ for some subset $W \subset V$, and each such subset gives rise to a vector $w = (\sum_{v\in W} v)/2 \in \mathbf{Z}^I$ for which $2 \cdot w \in \ker f$. Each such w, in turn, gives rise to an element $x = f(w) \in (\mathbf{Z}/n\mathbf{Z})^*$ satisfying $x^2 = f(2 \cdot w) = 1$, and therefore possibly to a decomposition of n into two non-trivial factors. If the factorization is trivial (because $x = \pm 1$), or more generally if the factors that are found fail to pass a prime power test as described earlier in this section, then one repeats the same procedure starting from a different dependence relation between the vectors $\bar{v}$. Note that it is useless to use a dependence relation that is a linear combination of dependence relations that have been used earlier. Also, if several factorizations of n into two factors are obtained, they should be combined into one factorization of n into several factors by a few gcd calculations. One stops when all factors are prime powers; if indeed

f is surjective and V generates $\ker f$, this is guaranteed to happen before all relations between the $\bar{v}$ are exhausted.

Example. A typical way to select the factor base is to choose

$$I = \{p : p \text{ is prime}, p \le B\},$$
$$a_p = (p \bmod n) \qquad (p \in I),$$

where B is a suitably chosen bound. Collecting relations between the a_p can be done as follows. Using a sieve, one searches for positive integers b with the property that both b and $n+b$ are "B-smooth", i. e., have all their prime factors $\le B$. Replacing, in the congruence $b \equiv n + b \bmod n$, both sides by their prime factorizations, we see that each such b gives rise to a multiplicative relation between the a_p. The main merit of the resulting factoring algorithm—which is, essentially, the number field sieve, with the number field chosen to be the field $\mathbf{Q}$ of rational numbers—is that it illustrates the above scheme concisely. It is not recommended for practical use, not because it is inefficient in itself, but because other methods are much faster.

The choice of the "smoothness bound" B is very important: if B, and hence $\#I$, is chosen too large, one needs to generate *many* relations, and one may end up with a matrix that is larger than one can handle in Step 3. On the other hand, if B is chosen too small, then not enough integers b will be found for which both b and $n+b$ are B-smooth. The same remarks apply to the other algorithms that satisfy our schematic description.

In practice, the optimal value for B is determined empirically. In theory, one makes use of results that have been proved about the function ψ defined by

$$\psi(x, y) = \#\{m \in \mathbf{Z} : 0 < m \le x,\ m \text{ is } y\text{-smooth}\};$$

so $\psi(x, y)/[x]$ is equal to the probability that a random positive integer $\le x$ has all its prime factors $\le y$. Brief summaries of these results, which are adequate for the purposes of factoring, can be found in [30, section 2; 21, section 2.A and (3.16)].

It will be found, not surprisingly, that both from a practical and a theoretical point of view the optimal choice of the smoothness bound and the performance of the factoring algorithm mainly depend on the size of the numbers that one wishes to be smooth.

The smaller these numbers are, the more likely they are to be smooth, the smaller the smoothness bound can be taken, and the faster the algorithm is.

In the above algorithm, one wishes the numbers $b(n+b)$ to be smooth, and since b is small these numbers may be expected to be $n^{1+o(1)}$ (for $n \to \infty$). The theory of the ψ-function then suggests that the optimal choice for B is

$$B = \exp\bigl((\tfrac{1}{2}\sqrt{2} + o(1))(\log n)^{1/2}(\log\log n)^{1/2}\bigr) \qquad (n \to \infty),$$

and that the running time of the entire algorithm is

$$\exp\bigl((\sqrt{2} + o(1))(\log n)^{1/2}(\log\log n)^{1/2}\bigr) \qquad (n \to \infty).$$

(This assumes that the dependencies in Step 3 are found by a method that is faster than Gaussian elimination.)

A big improvement is brought about by the *quadratic sieve* algorithm [30; 36], which belongs to the same family. In this algorithm the numbers that one wishes to be smooth are only $n^{1/2+o(1)}$. This leads to the conjectured running time

$$\exp\bigl((1 + o(1))(\log n)^{1/2}(\log\log n)^{1/2}\bigr) \qquad (n \to \infty),$$

the smoothness bound being approximately the square root of this. Although the quadratic sieve never had the honor of factoring a Fermat number, it is still considered to be the best practical algorithm for factoring numbers without small prime factors.

In the *number field sieve* [22], the numbers that one wishes to be smooth are $n^{o(1)}$, or more precisely

$$\exp\bigl(O((\log n)^{2/3}(\log\log n)^{1/3})\bigr),$$

and both the smoothness bound and the running time are conjecturally of the form

$$\exp\bigl(O((\log n)^{1/3}(\log\log n)^{2/3})\bigr).$$

This leads one to expect that the number field sieve is asymptotically the fastest factoring algorithm that is known. It remains to be tested whether for numbers in realistic ranges the number field sieve beats the quadratic sieve, if one does not restrict to special classes of numbers like Fermat numbers and Cunningham numbers.

3. The ninth Fermat number.

Let n be the number $F_9/2424833$, which has 148 decimal digits:

$$\begin{aligned} n = \ & 5529\,373746\,539492\,451469\,451709\,955220\,061537\,996975\,706118 \\ & 061624\,681552\,800446\,063738\,635599\,565773\,930892\,108210\,210778 \\ & 168305\,399196\,915314\,944498\,011438\,291393\,118209. \end{aligned}$$

In this section we discuss how the prime factorization of n was found.

There exist factoring methods—*not* satisfying the description given in section 2—that are especially good at finding *small* prime factors of a number, and most of these had been applied to n. Richard Brent tried Pollard's $p \pm 1$ method and a modified version of Pollard's rho method (see [21]), both without success. He estimates that if there had been a prime factor $< 10^{20}$, it would probably have been found by the rho method. The $p \pm 1$ method would have been successful if at least one of the four numbers $p_{49} \pm 1$, $p_{99} \pm 1$ had been built from small prime factors. The failure of this method is explained by the factorizations

$$\begin{aligned} p_{49} - 1 &= 2^{11} \cdot 19 \cdot 47 \cdot 82\,488781 \cdot 1143\,290228\,161321 \cdot 43\,226490\,359557\,706629, \\ p_{49} + 1 &= 2 \cdot 3 \cdot 167\,982422\,287027 \cdot 7397\,205338\,652138\,126604\,651761\,133609, \\ p_{99} - 1 &= 2^{11} \cdot 1129 \cdot 26813 \cdot 40\,644377 \cdot 17\,338437\,577121 \cdot \\ & \quad 16\,975143\,302271\,505426\,897585\,653131\,126520\,182328\,037821 \\ & \quad 729720\,833840\,187223, \\ p_{99} + 1 &= 2 \cdot 3^2 \cdot 83 \cdot 496412\,357849\,752879\,199991\,393508\,659621\,191392\,758432 \\ & \quad 074313\,189974\,107191\,710682\,399400\,942498\,539967\,666627. \end{aligned}$$

These factorizations were found by Richard Crandall with the $p-1$ method and the elliptic curve method, which is another factoring algorithm that is good at finding small prime factors [24; 3; 27]. He used a special second phase that he developed in collaboration with Joe Buhler, that is similar to the second phase given in [3].

Both Richard Brent and we attempted to factor n using the elliptic curve method, supplemented with a second phase. Brent tried 5000 elliptic curves, his first-phase bound

(i. e., the bound B_1 from [27]) ranging from 240000 to 400000. This took 200 hours on a Fujitsu VP 100. We tried approximately 2000 elliptic curves, with first-phase bounds ranging from 300000 to 1000000, during a one-week run on a network of approximately 75 Firefly workstations at Digital Equipment Corporation Systems Research Center (DEC SRC). The elliptic curve method did not succeed in finding a factor. Our experience indicates that *if* there had been a prime factor $< 10^{30}$, it would almost certainly have been found. If there had been a factor $< 10^{40}$ we should probably have continued with the elliptic curve method; our decision to stop was justified by the final factorization, which the elliptic curve method did not have a reasonable chance of finding without major technological or algorithmic improvements.

We do not know what is the best lower bound for the prime factors of n that had been rigorously established before n was factored completely. In [10] one finds the lower bound $2.5 \cdot 10^6$.

If we had been certain—which we weren't—that n has no prime factor $< 10^{30}$, then we would have known that n is a product of either two, three or four prime factors. Among all composite numbers of 148 digits that have no prime factor $< 10^{30}$, about 15.8% are products of three primes, about 0.5% are products of four primes, and the others are products of two primes. We expected—rightly, as it turned out—to find two prime factors, but we would have been more excited with three large ones.

The number n was factored by means of the *number field sieve*. For a description of this method, as it applies to Cunningham numbers, see [22]. In the present paper we describe the method only as it applied to n.

The number field that we used is the field $\mathbf{Q}(\sqrt[5]{2})$. The elements of this field can be written uniquely as $\sum_{i=0}^{4} r_i \sqrt[5]{2}^i$, with $r_i \in \mathbf{Q}$. To perform arithmetic operations on these expressions one uses the rule $\sqrt[5]{2}^5 = 2$. The elements for which all r_i belong to $\mathbf{Z}$ form the subring $\mathbf{Z}[\sqrt[5]{2}]$ of $\mathbf{Q}(\sqrt[5]{2})$. By means of standard techniques from algebraic number theory one proves, first, that $\mathbf{Z}[\sqrt[5]{2}]$ is a principal ideal domain, and, secondly, that the unit group $\mathbf{Z}[\sqrt[5]{2}]^*$ is isomorphic to $(\mathbf{Z}/2\mathbf{Z}) \oplus \mathbf{Z} \oplus \mathbf{Z}$, with $\epsilon_0 = -1$, $\epsilon_1 = -1 + \sqrt[5]{2}$ and $\epsilon_2 = -1 + \sqrt[5]{2}^2 - \sqrt[5]{2}^3 + \sqrt[5]{2}^4$ as a set of generators.

The *norm* $\mathbf{Na}$ of a non-zero ideal $\mathbf{a}$ of $\mathbf{Z}[\sqrt[5]{2}]$ is defined by $\mathbf{Na} = \#(\mathbf{Z}[\sqrt[5]{2}]/\mathbf{a})$, which is a positive integer. A *first degree prime ideal* of $\mathbf{Z}[\sqrt[5]{2}]$ is a non-zero ideal $\mathbf{p}$ of $\mathbf{Z}[\sqrt[5]{2}]$ for which $\mathbf{Np}$ is a prime number p; then $\mathbf{Z}[\sqrt[5]{2}]/\mathbf{p} \cong \mathbf{F}_p$, which is a field, so that $\mathbf{p}$ is indeed a prime ideal (and "first degree" indicates that the extension degree of $\mathbf{Z}[\sqrt[5]{2}]/\mathbf{p}$ over its prime field equals 1). The set of first degree prime ideals $\mathbf{p}$ is in bijective correspondence with the set of pairs $(p, c \bmod p)$, where p is a prime number and $c \in \mathbf{Z}$ is such that $c^5 \equiv 2 \bmod p$; if $\mathbf{p}$ corresponds to $(p, c \bmod p)$, then $\mathbf{Np} = p$, the map $\mathbf{Z}[\sqrt[5]{2}] \to \mathbf{Z}[\sqrt[5]{2}]/\mathbf{p} \cong \mathbf{F}_p$ maps $\sqrt[5]{2}$ to $(c \bmod p)$, and $\mathbf{p}$ is generated, as an ideal, by p and $c - \sqrt[5]{2}$. Since $\mathbf{Z}[\sqrt[5]{2}]$ is a principal ideal domain, each $\mathbf{p}$ can also be generated by a single element, which is only unique up to multiplication by units; let one such element be called $\pi_{\mathbf{p}}$. The description of the map $\mathbf{Z}[\sqrt[5]{2}] \to \mathbf{Z}[\sqrt[5]{2}]/\mathbf{p}$ just given provides us with an easy test for divisibility by $\pi_{\mathbf{p}}$: an element $\sum_i r_i \sqrt[5]{2}^i$ of $\mathbf{Z}[\sqrt[5]{2}]$, with $r_i \in \mathbf{Z}$, is divisible by $\pi_{\mathbf{p}}$ if and only if $\sum_i r_i c^i \equiv 0 \bmod p$, with p, c as above.

We next discuss the property that makes $\mathbf{Z}[\sqrt[5]{2}]$ useful in factoring n. By definition of n, we have $2^{512} \equiv -1 \bmod n$, so $(2^{205})^5 = 2^{1025} = 2 \cdot (2^{512})^2 \equiv 2 \bmod n$. In other words, 2^{205} is a fifth root of 2 (modulo n), so there is a ring homomorphism $\varphi\colon \mathbf{Z}[\sqrt[5]{2}] \to \mathbf{Z}/n\mathbf{Z}$ sending $\sqrt[5]{2}$ to $(2^{205} \bmod n)$, and of course $\sum_i r_i \sqrt[5]{2}^i$ to $\left(\sum_i r_i 2^{205i} \bmod n\right)$. An important role will be played by the element $\alpha = -\sqrt[5]{2}^3 \in \mathbf{Z}[\sqrt[5]{2}]$, which has the property that $\varphi(\alpha) = (2^{103} \bmod n)$; what is important about this is that 2^{103} is very small with respect to n—it is not much bigger than $\sqrt[5]{n}$.

We are now in a position to describe the selection of the factor base. Let the set I consist of: (i) the first 99700 prime numbers p; these are all primes $\leq B_1 = 1295377$; (ii) the three generating units ϵ_0, ϵ_1 and ϵ_2; (iii) the elements $\pi_{\mathbf{p}}$ for all 99500 first degree prime ideals $\mathbf{p}$ of $\mathbf{Z}[\sqrt[5]{2}]$ with $\mathbf{Np} \leq B_2 = 1294973$. For each $i \in I$, let $a_i = \varphi(i) \in \mathbf{Z}/n\mathbf{Z}$. These form the factor base.

The elements $\pi_{\mathbf{p}}$ were found by computing the norms of the 1092846526 expressions $\sum_{i=0}^4 h_i \alpha^i \in \mathbf{Z}[\alpha]$ for which the h_i have no common factor, $h_i \geq 0$ if h_{i+1} through h_4 are 0, and $\sum_{i=0}^4 h_i^2 2^{6i/5} \leq 15000$. In this way we determined 49726 of the 99500 generators. (This suggests that the Picard group—invertible ideals modulo principal ideals—of the order $\mathbf{Z}[\alpha]$ is of order 2, which can indeed be verified directly.) For the other 49774 prime ideals $\mathbf{p}$ the same search produced generators for the ideals $\alpha\mathbf{p}$ of norm $8 \cdot \mathbf{Np}$, so that the proper generators could be determined by dividing out α. It would have been better to search among the expressions $\sum_{i=0}^{4} h_i \sqrt[5]{2}^i$, but we found it less troublesome to use a program written for a previous occasion than to change it; the whole calculation took only a few hours on a single workstation.

Relations are found in several ways. In the first place, there are relations that are already valid in $\mathbf{Z}[\sqrt[5]{2}]$, before applying φ. Apart from the relation $\epsilon_0^2 = 1$, these come from prime numbers that, as an ideal, split into five (not necessarily distinct) first degree prime ideals. For example, the prime ideal decomposition of the prime number 2 is $(2) = \mathbf{p}^5$, where $\mathbf{p}$ corresponds to the pair $(2, 0 \bmod 2)$; we can choose $\pi_{\mathbf{p}} = \sqrt[5]{2}$, giving rise to the relation $2 = \sqrt[5]{2}^5$. A less trivial example is given by the prime number 5. Its prime ideal factorization is $(5) = \mathbf{p}^5$, with $\mathbf{p}$ corresponding to $(5, 2 \bmod 5)$; we can choose $\pi_{\mathbf{p}} = 1 + \sqrt[5]{2}^2$, and the relation we obtain now has a unit factor: $5 = \epsilon_1^2 \epsilon_2^{-2} \cdot (1 + \sqrt[5]{2}^2)^5$. Apart from 2 and 5, there are the primes p with the property that $(2 \bmod p)$ has five distinct fifth roots in $\mathbf{F}_p$. Each such prime is $1 \bmod 5$, and one out of every five primes that is $1 \bmod 5$ has this property. Below $\min\{B_1, B_2\} = 1294973$ there are 4944 such primes, the first being 151 and the last 1294471. This gives 2.5% of the ~ 200000 relations that are needed.

The remaining ~ 195000 relations between the a_i are found by searching for pairs of integers a, b, with $b > 0$, satisfying the following conditions:

(i) $\gcd(a, b) = 1$;

(ii) $|a + 2^{103} b|$ is built up from primes $\leq B_1$ and at most one larger prime p_1, which should satisfy $B_1 < p_1 < 10^8$;

(iii) $a + b\alpha$ is built up from prime ideals of norm $\leq B_2$ and at most one additional prime ideal, of which the norm p_2 should satisfy $B_2 < p_2 < 10^8$.

If the large prime p_1 in (ii) does not occur, then we write $p_1 = 1$, and likewise for p_2 in (iii). Pairs a, b for which $p_1 = p_2 = 1$ will be called *full* relations, and the other pairs *partial* relations.

Before we describe how the search for such pairs is performed, let us see how they give rise to relations between the a_i.

It is a charming exercise in algebraic number theory to prove that the prime ideal factorization of an element of the form $a + b\alpha$, with a, $b \in \mathbf{Z}$, $\gcd(a,b) = 1$, and $\alpha = -\sqrt[5]{2}^3$ as above, has the following properties: first, all prime ideals occurring in it are of the first degree; secondly, at most one prime ideal of a given norm p occurs in it; and finally, the product of the norms of the prime ideals occurring in it, with the proper multiplicities, equals $|a^5 - 8b^5|$. Hence, if (i) holds, then (iii) is equivalent to the condition that $|a^5 - 8b^5|$ be B_2-smooth, except possibly for a single prime divisor $p_2 > B_2$; and if (i) and (iii) hold, then the prime ideal factorization of $a + b\alpha$ is easy to write down.

Let us first consider the case that a, b is a full relation. Then, in the prime ideal factorization of $a + b\alpha$, we can replace each $\mathbf{p}$ by $\pi_{\mathbf{p}}$. This yields a product representation for $a + b\alpha$, except that the unit contribution has to be figured out. This is done by means of a technique described in [22, section 2], and it leads to an identity of the form

$$a + b\alpha = \prod_{i=0}^{2} \epsilon_i^{v(i)} \cdot \prod_{\mathbf{p}} \pi_{\mathbf{p}}^{u(\mathbf{p})},$$

with $\mathbf{p}$ ranging over the first degree prime ideals of norm $\leq B_2$, and with $v(i) \in \mathbf{Z}$, $u(\mathbf{p}) \in \mathbf{Z}_{\geq 0}$. Factoring $a + 2^{103}b$, we obtain an identity of the form

$$a + 2^{103}b = \prod_{p} p^{w(p)},$$

with p ranging over the prime numbers $\leq B_1$ and $w(p) \in \mathbf{Z}_{\geq 0}$ (if $a + 2^{103}b < 0$, replace a, b by $-a$, $-b$). Also, from $\varphi(\alpha) = (2^{103} \bmod n)$ we see that $a + b\alpha$ and $a + 2^{103}b$ have the same image under φ. Hence, if (i), (ii), (iii) are all satisfied, then we have

$$\prod_{i=0}^{2} \varphi(\epsilon_i)^{v(i)} \cdot \prod_{\mathbf{p}} \varphi(\pi_{\mathbf{p}})^{u(\mathbf{p})} = \prod_{p} \varphi(p)^{w(p)}.$$

Thus we see that each full relation a, b gives rise to a relation between the a_i.

With partial relations the situation is a bit more complicated. They only give rise to relations between the a_i if they are combined into *cycles*, as described in [22]. In each cycle, one takes an alternating product of relations $\varphi(a + b\alpha) = \varphi(a + 2^{103}b)$, in such a way that the large prime ideals and prime numbers cancel. This leads, by a procedure that is completely similar to the above one, to a relation between the a_i. It is not necessary to know generators $\pi_{\mathbf{p}}$ for the large prime ideals, since these are divided out.

If, in (iii), we have $p_2 > 1$, then the additional prime ideal corresponds to the pair $(p_2, c \bmod p_2)$, where $c = a^2/(2b^2)$; this is uniquely determined by p_2 unless $p_2 \equiv 1 \bmod 5$.

The search for pairs a, b as above was performed by means of the sieving technique described in [22]. We used 2.2 million values of b, all satisfying $0 < b < 2.5 \cdot 10^6$. For each b, we sieved $|a + 2^{103}b|$ with the primes $\leq B_1$, and $|a^5 - 8b^5|$ with the primes $\leq B_2$, each over 10^8 consecutive a-values centered roughly at $8^{1/5} \cdot b$.

The best values for a are those that are close to $8^{1/5} \cdot b$. If we take for instance $b = 10^6$, then for such a's we are asking for simultaneous smoothness of two numbers close to 10^{37} and $8 \cdot 10^{30}$; for $b = 10^7$ this becomes 10^{38} and $8 \cdot 10^{35}$. The quadratic sieve algorithm when applied to n would depend on the smoothness of numbers close to $\sqrt{n}$ times the sieve length, which amounts to at least 10^{80}. This is the main reason why the number field sieve performs better for this value of n than the quadratic sieve. The comparison is still very favorable when a is further removed from the center of its interval, although the numbers become larger. The tails of the interval are less important, so the fact that centering it at 0 would have been better did not bother us.

Smaller b-values are more likely to produce good pairs a, b than larger ones. The best approach is therefore to process the b-values consecutively starting at 1, until the total number of full relations plus the number of independent cycles among the partial relations that have been found equals ~ 195000. One can only hope that this happens before b assumes prohibitively large values. Of course, B_1 and B_2 must have been selected in such a way that one is reasonably confident that this approach will succeed. This is discussed below.

We started sieving in mid-February 1990 on approximately 35 workstations at Bellcore. On the workstations we were using (DEC3100's and SPARC's) each b took approximately eight minutes to process. We had to split up the a-intervals of length 10^8 into 200 intervals of length $5 \cdot 10^5$, in order to avoid undue interference with other programs. After a month of mostly night-time use of these workstations, the first range of 10^5 b's was covered. Mid-March, the network of Firefly workstations at DEC SRC was also put to work. This approximately tripled our computing power. With these forces we could have finished the sieving task within another seven months. At the time, we did not know this, since we did not know how far we had to go with b.

Near the end of March it was rumored that we had a competitor. After attempts to join forces had failed, we decided to accelerate a little by following the strategy described in [23]. We posted messages on various electronic bulletin boards, such as sci.crypt and sci.math, soliciting help. A sieving program, plus auxiliary driver programs to run it, were made available at a central machine at DEC SRC in Palo Alto to anyone who expressed an interest in helping us. After contacting one of us personally, either by electronic mail or by telephone, a possible contributor was also provided with a unique range of consecutive b-values. The size of the range assigned to a particular contributor depended on the amount of free computing time the contributor expected to be able to donate. Each range was sized to last for about one week, after which a new range was assigned. This allowed us to distribute the available b's reasonably evenly over the contributors, so that the b's were processed more or less consecutively.

It is difficult to estimate precisely how many workstations were enlisted in this way. Given that we had processed 2.2 million b's by May 9, and assuming that we mostly got night-time cycles, we must have used the equivalent of approximately 700 DEC3100 workstations. We thus achieved a sustained performance of more than 3000 mips for a period of five weeks, at no cost. (Mips is a unit of speed of computing, 1 mips being one million instructions per second.) The total computational effort amounted to about 340 mips-years (1 mips-year is about $3.15 \cdot 10^{13}$ instructions). We refer to section 5 for the names of many of the people and institutions who responded to our request and donated computing time.

Each copy of the sieving program communicated the pairs a, b that it found by electronic mail to DEC SRC, along with the corresponding pair p_1, p_2 and, in the case $p_2 > 1$, $p_2 \equiv 1 \bmod 5$, the residue class $(a/b \bmod p_2)$. In order not to overload the mail system at DEC SRC, the pairs were sent at regular intervals. At DEC SRC, these data were stored on disk. Notice that the corresponding two factorizations were *not* sent, due to storage limitations. These were later recomputed at DEC SRC, but only for the relations which turned out to be useful in producing cycles. The residue class $(a/b \bmod p_2)$ could also have been recomputed, but since it simplified the cycle counting we found it more convenient to send it along. Notice that $(a/b \bmod p_2)$ distinguishes between the five prime ideals of norm p_2.

When we ran the quadratic sieve factoring algorithm in a similar manner [23], we could be wasteful with inputs: we made sure that different inputs were distributed to our contributors, but not that they were actually processed. We could afford this approach because we had millions of inputs, each of which was in principle capable of producing thousands of relations. For the number field sieve the situation is different: each b produces only a small number of relations, if any, and the average yield decreases as b increases. In order not to lose our rather scarce and valuable 'good' inputs (i. e., the small b-values), we wanted to be able to monitor what happened to them after they were given out. For this reason, each copy of the sieving program also reported through electronic mail which b's from its assigned range it had completed. This allowed us to check them off from the list of b's we had distributed. Values that were not checked off within approximately ten days were redistributed. Occasionally this led to duplications, but these could easily be sorted out.

By May 7 we had used approximately 2.1 million b's less than 2.5 million, and we had collected 44106 full relations and 2999903 partial relations. The latter gave rise to a total of 158105 cycles. Since $44106 + 158105$ is well over 195000, this was already more than we needed. Nevertheless, to facilitate finding the dependencies, we went on for two more days. By May 9, after approximately 2.2 million b's, we had 45719 full relations and 176025 cycles among 3114327 partial relations. Only about one fifth of these 3114327 relations turned out to be useful, i. e., actually appeared in one of the 176025 cycles. It took a few hours on a single workstation to find the cycles in terms of the a, b, p_1, and p_2 involved, using an algorithm explained in [22]. The number of cycles of each length is given in the table below.

This is what we hoped and more or less expected to happen, but there was no guarantee that our approach would work. For any choice of B_1 and B_2 (and size of a-interval) we could quite accurately predict how many full and partial relations we would find by processing all b's up to a certain realistic limit. This made it immediately clear that values B_1 and B_2 for which full relations alone would suffice would be prohibitively large.

cycle length	number of cycles
2	48289
3	43434
4	32827
5	22160
6	13444
7	7690
8	4192
9	2035
10	1055

cycle length	number of cycles
11	473
12	243
13	100
14	55
15	14
16	8
17	2
19	2
20	2

Thus we were faced with the problem of choosing B_1 and B_2 in such a way that the full relations plus the cycles among the partials would be likely to provide us with sufficiently many relations between the a_i. It is, however, hard to predict how many partials are needed to produce a given number of cycles. For instance, the average number of cycles of length 2 resulting from a given number of partials can be estimated quite accurately, but the variance is so large that for each particular collection of partials this estimate may turn out to be far too optimistic or pessimistic. An estimate that is too low is harmless, but an estimate that is too high can be fatal: once b is sufficiently large, hardly any new fulls or partials will be found, and the only alternative is to start all over again with larger B_1 and B_2. As a consequence, we selected the values for B_1 and B_2 carefully and conservatively, we made sure that we did not skip many b-values, and we milked each b for all it was worth by using an excessively long a-interval.

We decided to set the size of the factor base approximately equal to $2 \cdot 10^5$ only after experiments had ruled out $1.2 \cdot 10^5$, $1.4 \cdot 10^5$, and $1.6 \cdot 10^5$ as probably too small, and $1.8 \cdot 10^5$ as too risky. For $2 \cdot 10^5$ we predicted ~ 50000 full and at least 3 million partial relations after the first 2.5 million b's. This prediction was based on the first figure below, where the results of some preliminary runs of the sieving program are presented. For i ranging from 1 to 40 the total number of relations (fulls plus partials) found for the 300 consecutive b's starting at $i \cdot 10^5$ is given as a function of i. The upper curve gives the yield for an a-interval of length 10^8, the lower curve for length $2 \cdot 10^7$.

Our experience with other number field sieve factorizations made us hope that 3 million partials would produce 150000 cycles, which indeed turned out to be the case. But even if 3 million partials had not been enough, we knew that the b's between 2.5 and 4 million would lead to at least another million partials, and a good chance to find enough cycles. In the second figure we give the number of cycles, the number of full relations, and their sum, that were obtained after we had found a given number of partial relations. This does not include the initial 4944 relations.

Now that we have seen how everything worked out in this particular case, we know that with the same B_1 and B_2 and a much *smaller* a-interval we could have spent a lot less time to produce 3 million partials after using *more* b's. For example, halving the length of the a-interval would reduce the average yield per b-value by only 15%. It would probably have been optimal to use about $1.5 \cdot 10^7$ values of a per b, with b ranging up to about 5.5 million; this would have taken about 40% of the time that we actually spent. Still, we cannot be certain that this would have given rise to the same number of cycles.

We might have profited a little from the known factor 2424833 of F_9 by putting it in the factor base, along with the prime ideal corresponding to $(2424833, 2^{205} \bmod 2424833)$, since the prime appears on the right if and only if the prime ideal appears on the left. We only realized that we could have done this after the third author had found seven 'awfully suspicious' pairs a, b, namely pairs with $p_1 = p_2 = 2424833$, while generating the cycles.

To conclude the second step, the full relations and the cycles had to be transformed into relations between the a_i. To this end we recomputed the $2 \cdot 722241$ factorizations corresponding to the 722241 (not all distinct) pairs a, b involved, and determined the unit contributions. This work was divided over fifteen workstations at DEC SRC, and it took about sixteen hours.

This finishes the description of the first two steps of the number field sieve as applied to n. The final step is described in the next section.

4. Finding dependencies.

As a result of the computations described in the previous section, we had $4944 + 45719 + 176025 = 226688$ relations between $3 + 99700 + 99500 = 199203$ different a_i's. (We did not include the two relations based on the decompositions of 2 and 5 that were given in section 3.) To finish the factorization of n, we had to determine a few dependencies between the 226688 rows of the 199203 column matrix over $\mathbf{F}_2$ that is obtained by taking the relations (i. e., the exponents of the a_i) modulo 2. A dense representation of this matrix would require more than 5 Gigabytes ($= 5 \cdot 2^{30}$ bytes) of storage, where one byte represents 8 bits. Fortunately, the matrix is sparse, because relatively few primes and prime ideals appear in the factorizations leading to the relations; this situation is slightly worsened by the fact that many relations were obtained by combining partial relations. In any case, there were only 11264596 non-zero entries in the matrix, for an average of of 49.7 non-zero entries per row. Thus, the entire matrix could easily be stored.

Finding dependencies was still a challenging task. The sieving step had posed no problems that had not already been solved for other numbers, except that an unusually large amount of computing time had to be arranged. The matrix step, however, presented a difficulty that had not been encountered in previous factorizations. Actually, the only reason that we had not embarked upon the factorization of F_9 earlier is that we did not know how to handle the matrix.

The largest matrices that we had ever dealt with in previous factorizations contained approximately 80000 columns, and a few more rows. Dependencies modulo 2 among the rows were found in an entirely straightforward fashion using ordinary Gaussian elimination, with pivot-search from the sparse side. In this way some profit could be gained from the sparseness, but not much: usually, the storage that one ultimately needs is about two thirds of what it would have been in the dense case. This fits in only 0.5 Gigabytes for an 80000 matrix, so that the elimination task for such a matrix is more or less trivial for someone with access to a large supercomputer. At DEC SRC, where the computations were carried out, the only machine with enough disk space that could entirely be devoted to the elimination task was a four processor Firefly workstation. On this workstation, elimination of a sparse 80000 matrix takes approximately six weeks. Here we should note that for two of the three 80000 matrices we processed in this way, the resulting dependencies turned out to be faulty. In both instances a re-run (with another six week wait!) was successful. We suspect that in both first runs an irreproducible cache read or write error had occurred. Clearly, a single bit error can render the entire computation worthless.

Extrapolation of these figures to a 200000 matrix did not look promising. Even if our workstation had enough disk space, $6 \cdot (2.5)^3 \approx 90$ weeks is unacceptably long, and the probability of a bit error occurring would be unacceptably large. On a supercomputer the figures still looked unattractive. Therefore we investigated whether there is a better way to profit from the sparseness of the matrix.

Among the several existing techniques for dealing with sparse matrices, we decided to attempt *structured Gaussian elimination* [18; 31]. In structured Gaussian elimination the columns of the matrix are partitioned into *heavy* and *sparse* columns. Initially all columns are considered sparse. Roughly speaking, one does eliminations with pivots in sparse columns that only cause fill-in in the heavy columns of the matrix, thereby removing the pivot rows and columns from the matrix. When this is impossible, one either moves some of the columns from the sparse to the heavy part, or one removes some excess rows, if there are any. Next one tries again. This is repeated until no sparse columns are left. For reasons that are not yet understood it seems to be beneficial to have many excess rows initially.

During this process one does not keep track of what happens in the heavy columns, but one only remembers which eliminations have been carried out. This information can then be used to build the smaller but much denser matrix corresponding to the heavy columns, and to convert dependencies among its rows into dependencies among the rows of the original matrix. Dependencies in the smaller matrix can for instance be found using ordinary Gaussian elimination.

It took us a few hours on a single workstation to reduce our 226688 row and 199203 column matrix to a 72413 row and 72213 column matrix. We kept 200 excess rows, to have a reasonable guarantee that one of the dependencies would be useful. It took slightly more than one day to actually build the small matrix, and to verify that all entries in the sparse and eliminated part were indeed zero. The small matrix turned out to be entirely dense. In the small matrix we included at regular intervals rows that consisted of the sum (modulo 2) of all previous rows, thus creating several spurious but predictable dependencies.

We immediately set out to reduce this 'small' matrix, using ordinary Gaussian elimination and our familiar set-up at DEC SRC. This time, however, we had some protection against bit errors: if one of the spurious dependencies failed to show up, something must have gone wrong recently. Then we could back up a few hundred rows, and restart the elimination from a point where we were confident that everything was still correct. We estimate that the entire elimination on this single workstation would have taken less than seven weeks.

While this process was making its slow progress, the third author, tired of keeping it alive and not too confident of its outcome, contacted Roger Frye and Mike McKenna at Thinking Machines, and explained the problem to them. After a short while they had written a Gaussian elimination program for a Connection Machine. They estimated that their program, when executed on a 65536-processor Connection Machine, could handle our 72000 matrix within three hours. Jim Hudgens and George Marsaglia at the Supercomputer Computation Research Institute at Florida State University arranged the computer time we needed. We sent, by Federal Express, a box with ten tapes containing the data for the matrix to Florida. Jim Hudgens consolidated these ten tapes into one "exotape". In the night of June 14 he mounted the exotape, so that Roger Frye and Mike McKenna, remotely logged in from Thinking Machines in Cambridge, MA, could read the data as one large sequential file, and execute the program. It solved the system in three hours, but then a crash occurred, due to a mistake in the output routine. The second run, which again took three hours, produced a few hundred dependencies among the rows of the dense 72000 matrix.

In the early morning of June 15, 1990, the dependencies were sent, electronically, to DEC SRC, were they were converted into dependencies of the original sparse 200000 matrix. At least, that is what we hoped that they would turn out to be. At 9:15 PDT we started our final program, the attempt to factor n by processing the dependencies sequentially until the factorization was found. This led to the most exciting moment of the entire factorization of F_9: at 9:45 PDT the program concluded that the first alleged dependency among the rows of the sparse 200000 matrix was a true one. This moment of great relief could not be spoilt by the sobering message, displayed at 10:15 PDT, that the first dependency had just found a trivial factorization of n. An hour later, at 11:15 PDT (18:15 GMT), the second dependency proved to be luckier by finding a 49-digit factor. Both this factor and the 99-digit cofactor were announced prime, which means that no witnesses to their compositeness could be found among five randomly chosen integers (see section 2).

Five minutes later the backup Gaussian elimination process, still crunching along on a single workstation, was terminated, five days short of its goal. Still on June 15, Andrew Odlyzko used the first author's Cray X-MP implementation of the Jacobi sum primality test [8] to prove that both factors are indeed prime.

5. Acknowledgments.

We would like to thank everyone who helped with collecting relations: Kannan Alagappan, Steve Albrecht, Sharon Allan, Dean Alvis, Jon M. Arnold, Thomas G. Arnold, Amy Asbury, Elena Aschkenasi, Bob Ayers, Pete Balkus, Joel Bartlett, David M. Bernardo, Doug Bonin, Anita Borg, Wieb Bosma, Patrick Boyle, Richard P. Brent, John Brownie, Mark Buda, Joe Buhler, Bruce Bullis, Toni Burks, Al Butler, Neil J. Calkin, Don Carignan, Jerome Chailloux, George Chaltas, Chran-Ham Chang, Bruce Chase, Scott Chase, Carol Cignotti, Ian Clements, Donn Coe, Henri Cohen, Leisa Condie, Tom Coppeto, Jeff Cormier, Ken Cowan, Richard Critz, Richard L. Curtis, Stu Davidson, Chris DeLise, Bhupendra Desai, Srinivas Desirazu, Bob Devine, Jeff Diewald, Jeff Dike, John Dillon, Jeremy Dion, Paul DiTommaso, Dan Donahue, Jeff Donovan, Mike Doody, David W. Dunkle, Charles B. Dunn, Gene Durso, John Dustin, Charles Dyer, Alan Eustace, Marc Evans, Mike Ferrara, Ed Flecchia, John Forecast, Joel Foss, Chris Franklin, Sandy Fraser, Hania Gajewska, Sachin Galgalikar, Edward A. Gardner, Eran Gartner, Morrie Gasser, Eric Gauthier, Lance J. Gay, Carl Gerstle, David Gibson, Stephen Gildea, Andy Goldstein, Mike Greenfield, Tim Greenwood, Liz Guth, Ramsey Haddad, Kenneth J. Haduch, Lori Hagen, John C. Hallyburton, Jr., Gary Harding, Bob Harris, Charles Haynes, B. J. Herbison, Ted Hess, Max Hillson, Buren Hoffman, Jim Horning, Felix S. Hsu, Han Hsu, Scott Huddleston, Gary Huff, Peter Ilieve, Frank Infante, Philippe Jacquet, Jeff Janock, Jeff Jenkins, Mike Johnson, Kevin Jones, Dave Juitt, Bill Kalsow, Irving Kaplansky, Harsh Kapoor, Philip L. Karlton, Bill Katz, Christopher A. Kent, Jeff Kenyon, Alan Kirby, Manfred Koethe, John T. Kohl, David M. Kuchta, Rick Landau, Sundaram Laxman, Ted Lemon, Bill Licea-Kane, John Linn, Walter Lioen, Todd Little, Jim Lo, Mark Longo, Kevin Lynch, Pat Madden, Joseph A. Martin, Robert N. Mayo, Murray S. Mazer, James T. McCartney, Joel McCormack, Ellen McDermott, Randall V. Meyers, Michael J. Miano, Steven Miller, Thomas Mitchell, Jeffrey Mogul, Bruce Moore, François Morain, A. E. Mossberg, David Mostardi, Victoria Murphy, Gopal Nagarajan, Jeff E. Nelson, Chuck Newman, Marc Nozell, Vinay Nulkar, Paul E. Oppenheimer, Lisa Palermo, Bill Parke, Tom Patterson, Eileen Perez, Don Pettini, Nigel Poole, Eric D. Postpischil, Edward G. Prentice, Dennis Racca, Ramgopal Ramgiri, Ram Rao, Jon Reeves, Brian Reid, August G. Reinig, Herman te Riele, John Riley, Buzzy Ritter, John Robinson, D. C. Rocks, David M. Rosenberg, Eduardo Santiago, Sanjay Saxena, Richard Schedler, Jeffrey I. Schiller, Michael Sclafani, Jeff Sebring, Mike Sekurski, Shekhar Sengupta, Mark Shand, Robert Shen, Bob Silverman, John Simakauskas, Al Simons, Michael Soha, Kiran Somalwar, Bill Sommerfeld, Bob Souza, Vivian Sovinsky, Jerry Stange, Alan Stanier, John T. Stapleton, Jorge Stolfi, Geof Stone, Steve Strange, Richard Swan, Ed Taranto, Pahtiban Thilagar, Benjamin J. Thomas III, Bob Thomas, Gary Thomas, Mathews Thomas, Dennis Ting, Ward Travis, Win Treese, Tom Truscott, Jim Turner, Percy Tzelnic, Bob Unnold, Srinivasa Uppugunduri, Mohan Vaghul, Virendra Verma, Brick Verser, Paul Vixie, David Wall, David F. Wall, Chuck Wan, Bradley M. Waters, Dave Weiss, Dan White, Bill Whitney, Dick Wilkins, Dik Winter, Ted Wobber, Frank M. Woodall, Jr., Tom Woodburn, John Wray, Frank Zereski, Paul Zimmermann, John Zornig, plus many people at Bellcore and MSRI, at DEC's research labs CRL, PRL, SRC, and WRL, and at WSE and WSL. This list is incomplete, since some of our contributors were known to us only as electronic addresses that were no longer in service when we tried to get their names. We apologize for any omissions and misspellings.

We are grateful to the people who helped with Gaussian elimination: Andrew Odlyzko and Carl Pomerance for theoretical assistance, Jim Hudgens, George Marsaglia and the Supercomputer Computation Research Institute at Florida State University for computing time on the Connection Machine, and Roger Frye and Mike McKenna at Thinking Machines for writing and running their Gaussian elimination program on the Connection Machine.

The Mathematical Sciences Research Institute in Berkeley is gratefully acknowledged for giving us the opportunity to meet with various people involved in this project. We also thank our competitor—whose name, surprisingly, occurs in the above list—for spurring us on, and Richard Brent, Richard Crandall, Wilfrid Keller and Sam Wagstaff for providing information concerning previous F_9 factoring attempts and factors of Fermat numbers in general.

The first author would like to thank DEC SRC and Bob Taylor for their hospitality and support in the Summer of 1989 when most of the programming for the implementation of the number field sieve was done. The second author would like to thank the Institute for Advanced Study in Princeton for their hospitality and support while this paper was being written. The second author was supported by NSF under Grant No. DMS 90-02939.

References.

1. G. E. Andrews, *The theory of partitions*, Addison-Wesley, Reading, Mass., 1976.
2. A. O. L. Atkin, F. Morain, *Elliptic curves and primality proving*, to appear.
3. R. P. Brent, *Some integer factorization algorithms using elliptic curves*, Austral. Comp. Sci. Comm. **8** (1986), 149–163.
4. R. P. Brent, *Factorization of the eleventh Fermat number (preliminary report)*, Abstracts Amer. Math. Soc. **10** (1989), 89T-11-73.
5. R. P. Brent, *Parallel algorithms for integer factorisation*, pp. 26–37 in: J. H. Loxton (ed.), *Number theory and cryptography*, London Math. Soc. Lecture Note Series **154**, Cambridge University Press, Cambridge, 1990.
6. R. P. Brent, J. M. Pollard, *Factorization of the eighth Fermat number*, Math. Comp. **36** (1981), 627–630.
7. J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman, S. S. Wagstaff, Jr., *Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers*, second edition, Contemporary Mathematics **22**, Amer. Math. Soc., Providence, 1988.
8. H. Cohen, H. W. Lenstra, Jr., *Primality testing and Jacobi sums*, Math. Comp. **42**

(1984), 297–330.

9. J.H. Conway, *On numbers and games*, Academic Press, London, 1976.

10. A. Cunningham, A.E. Western, *On Fermat's numbers*, Proc. London Math. Soc. (2) **1** (1904), 175.

11. A.J.C. Cunningham, H.J. Woodall, *Factorisation of* $(y^n \mp 1)$, $y = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers* (n), Hodgson, London, 1925.

12. L.E. Dickson, *History of the theory of numbers*, vol. I, Carnegie Inst. of Washington, Washington, 1919.

13. L. Euler, *Observationes de theoremate quodam Fermatiano aliisque ad numeros primos spectantibus*, Comm. acad. sci. Petropol. **6** (1732/1733), 103–107; pp. 1–5 in: *Leonhardi Euleri opera omnia*, Ser. I, vol. II, Teubner, Leipzig, 1915.

14. C.F. Gauss, *Disquisitiones arithmeticae*, Fleischer, Leipzig, 1801.

15. A. Gérardin, *Méthodes de Landry*, L'intermédiaire des mathématiciens **16** (1909), 199–201.

16. J.C. Hallyburton, Jr., J. Brillhart, *Two new factors of Fermat numbers*, Math. Comp. **29** (1975), 109–112; *corrigendum*, ibid. **30** (1976), 198.

17. W. Keller, *Factors of Fermat numbers and large primes of the form* $k \cdot 2^n + 1$. *II*, in preparation.

18. B.A. LaMacchia, A.M. Odlyzko, *Solving large sparse systems over finite fields*, Proc. Crypto '90, to appear.

19. F. Landry, *Note sur la décomposition du nombre* $2^{64} + 1$ *(Extrait)*, C. R. Acad. Sci. Paris **91** (1880), 138.

20. A.K. Lenstra, *Factorization of polynomials*, pp. 169–198 in [26].

21. A.K. Lenstra, H.W. Lenstra, Jr., *Algorithms in number theory*, Chapter 12 in: J. van Leeuwen (ed.), *Handbook of theoretical computer science*, Volume A, *Algorithms and complexity*, Elsevier, Amsterdam, 1990.

22. A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, J.M. Pollard, *The number field sieve*, in preparation. Extended abstract: Proc. 22nd Annual ACM Symp. on Theory of Computing (STOC), Baltimore, May 14–16, 1990, 564–572.

23. A.K. Lenstra, M.S. Manasse, *Factoring by electronic mail*, Advances in Cryptology,

Eurocrypt '89, Lecture Notes in Comput. Sci. **434** (1990), 355–371.

24. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. **126** (1987), 649–673.
25. H. W. Lenstra, Jr., C. Pomerance, *A rigorous time bound for factoring integers*, in preparation.
26. H. W. Lenstra, Jr., R. Tijdeman (eds), *Computational methods in number theory*, Math. Centre Tracts **154/155**, Mathematisch Centrum, Amsterdam, 1983.
27. P. L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Math. Comp. **48** (1987), 243–264.
28. M. A. Morrison, J. Brillhart, *A method of factoring and the factorization of* F_7, Math. Comp. **29** (1975), 183–205.
29. T. Pepin, *Sur la formule* $2^{2^n}+1$, C. R. Acad. Sci. Paris **85** (1877), 329–331.
30. C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, pp. 89–139 in [26].
31. C. Pomerance, J. W. Smith, *Reduction of large, sparse matrices over a finite field via created catastrophes*, manuscript in preparation.
32. M. O. Rabin, *Probabilistic algorithm for testing primality*, J. Number Theory **12** (1980), 128–138.
33. H. Riesel, *Prime numbers and computer methods for factorization*, Birkhäuser, Boston, 1985.
34. C. P. Schnorr, H. W. Lenstra, Jr., *A Monte Carlo factoring algorithm with linear storage*, Math. Comp. **43** (1984), 289–311.
35. D. Shanks, J. W. Wrench, Jr., *Calculation of* π *to* 100,000 *decimals*, Math. Comp. **16** (1962), 76–99.
36. R. D. Silverman, *The multiple polynomial quadratic sieve*, Math. Comp. **48** (1987), 329–339.
37. P. Tannery, C. Henry (eds), *Oeuvres de Fermat*, vol. II, *Correspondance*, Gauthier-Villars, Paris, 1894.
38. A. Weil, *Number theory: an approach through history*, Birkhäuser, Boston, 1983.
39. D. Wiedemann, *An iterated quadratic extension of* GF(2), Fibonacci Q. **26** (1988),

290–295.

Bellcore, 445 South Street, Morristown, NJ 07960, U. S. A.

Department of Mathematics, University of California, Berkeley, CA 94720, U. S. A.

DEC SRC, 130 Lytton Avenue, Palo Alto, CA 94301, U. S. A.

Tidmarsh Cottage, Manor Farm Lane, Tidmarsh, Reading, Berkshire, RG8 8EX, United Kingdom.