MATHEMATICS OF COMPUTATION
VOLUME 50, NUMBER 181
JANUARY 1988, PAGES 261-263

The Twentieth Fermat Number is Composite

By Jeff Young and Duncan A. Buell

Abstract. The twentieth Fermat number, $F_{20} = 2^{2^{20}} + 1$, has been proven composite by machine computation.

The Fermat numbers are the numbers $F_n = 2^{2^n} + 1$, orginally conjectured by Fermat to be prime for all n. In fact, only for n equal to 0 through 4 are they known to be prime, and small factors of $F_9, F_{11}, F_{12}, F_{15}, F_{16}$, have been known for some time. As part of a long-term test of the hardware reliability of the Cray-2 supercomputer at the Supercomputing Research Center, the authors proved that $F_{20} = 2^{2^{20}} + 1$, which had been the smallest Fermat number of unknown character, is composite. The test for compositeness was the standard technique of Pépin [5]: For $n \geq 1, F_n$ is prime if and only if $3^{(F_n-1)/2} \equiv -1 \pmod{F_n}$. This test for compositeness does not, of course, produce factors of the number, but was the test used for proving the compositeness of $F_7, F_8, F_{10}, F_{13}, F_{14}$ [2], [3], [4], [6], [7].

The result of the computation on the Cray-2 has been verified by performing the same computation on a Cray X-MP belonging to Cray Research. The total computation time on the Cray-2 was about 10 CPU days; the time on the Cray X-MP was 82 hours. Both programs ran as single-processor programs on any available CPU of the respective machines; the ability of either computer to run in parallel on multiple CPU's was not used. The time needed to test F_n, for n in the range 10 through 20, is just slightly more than four times the time needed to test F_{n-1}: The number of multiplications doubles in incrementing n, and the time required for each multiplication doubles, being dependent almost entirely on the length of the operands. Our programs would thus determine the character of F_{22}, which is now the smallest Fermat number of unknown character, in a little more than 16 times the time needed for our computation on F_{20}.

The table below summarizes what is now known about the Fermat numbers for n less than or equal to 22. A status list for larger n appears in [1].

This computation, roughly one million squarings modulo a one million bit number, would be impossible to do even on supercomputers without fast Fourier transform techniques for integer multiplication. Since one reason for performing this computation was to verify hardware reliablity and not to minimize the execution time, the program was written entirely in Cray Fortran and called Cray library functions for the FFT's. The program itself was very simple and only about 200 lines long, much of which was used for checkpointing and restarting the program. The program was called into execution every time the Cray-2 was restarted, and so

Received April 6, 1987; revised June 5, 1987.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 11Y11, 11A51, 11-04.

it ran essentially without any operator or programmer intervention over a period of nearly a month. The substantial difference in execution time on the two machines can probably be explained in large part by the fact that the FFT routines used on both machines were identical, but have been optimized by Cray for the X-MP.

Values of n	Character of F_n
$0, 1, 2, 3, 4$	Prime
$5, 6, 7, 8$	Composite and completely factored
$9, 13$	One prime factor known, composite cofactor
$15, 16, 17, 18, 21$	One prime factor known, cofactor of unknown character
$10, 11$	Two prime factors known, composite cofactor
19	Two prime factors known, cofactor of unknown character
12	Five prime factors known, composite cofactor
$14, 20$	Composite, no factor known
22	Character unknown

We estimate that the total number of floating-point multiplications and divisions was about $2.3 \cdot 10^{13}$; this leads to an estimate of about 28 Megaflops sustained by the Cray-2 over the 10 days of computation. While our estimate of the number of operations is admittedly low, it is unlikely to be low by more than a factor of 2. The 28 Megaflops should not, however, be taken as a valid benchmark of Cray-2 performance, since this computation is dominated by the FFT's and, as mentioned, the code used for the FFT's was optimized for a different machine.

The test for compositeness produces the residue R_n of $3^{(F_n-1)/2}$ modulo F_n; F_n is prime if and only if R_n is -1. Some legitimate skepticism must be attached to the conclusion that F_{20} is composite, then, since an error either in the hardware or in the computation is almost certainly going to produce an erroneous residue which would nonetheless lead to the expected conclusion that F_{20} is composite. By producing the same residue R_{20} on two different machines, we feel that the possibility of hardware error has been eliminated. As for the possiblity of the two machines merely duplicating software or computational errors, we point out that although the Cray-2 and the Cray X-MP ran the same Fortran code, their respective Fortran compilers will have produced substantially different executable modules for the somewhat different assembly languages. We feel confident, then, that there were no subtle flaws in the programs or the compilers.

As for the possibility that we have implemented the algorithm incorrectly, Selfridge and Hurwitz [7] published, for purposes of later verification, the values of R_n modulo 2^{36}, $2^{36} - 1$, and $2^{35} - 1$ for $n = 7, 8, 13$, and 14. We checked, with our programs, that we produced the same residues for these Fermat numbers (the computation for F_{14} requires only three minutes), and list the values of R_{20} modulo 2^{36}, $2^{36} - 1$, and $2^{35} - 1$:

$$R_{20} \equiv 175517362761 \pmod{2^{36}},$$
$$R_{20} \equiv 411337412531 \pmod{2^{36} - 1},$$
$$R_{20} \equiv 161572365764 \pmod{2^{35} - 1}.$$

For the above reasons, we feel confident that our conclusion that F_{20} is composite is based on a correct computation of the residue R_{20}.

We remark that this 10-day computation on a supercomputer may well be the largest computation ever performed whose result is a single bit answer. Never have so many circuits labored for so many cycles to produce so few output bits.

Acknowledgments. We are grateful to the anonymous referee for suggestions which have no doubt made this paper much better than it would otherwise have been.

Cray Research, Inc.
1440 Northland Drive
Mendota Heights, Minnesota 55120

Supercomputing Research Center
4380 Forbes Boulevard
Lanham, Maryland 20706

1. WILFRID KELLER, "Factors of Fermat numbers and large primes of the form $k \cdot 2^n + 1$," *Math. Comp.*, v. 41, 1983, pp. 661–673.

2. J. C. MOREHEAD, "Note on Fermat's numbers," *Bull. Amer. Math. Soc.*, v. 11, 1905, pp. 543–545.

3. J. C. MOREHEAD & A. E. WESTERN, "Note on Fermat's numbers," *Bull. Amer. Math. Soc.*, v. 16, 1909, pp. 1–6.

4. G. A. PAXSON, "The compositeness of the thirteenth Fermat number," *Math. Comp.*, v. 15, 1961, p. 420.

5. P. PÉPIN, "Sur la formule $2^{2^n} + 1$," *C. R. Acad. Sci. Paris*, v. 85, 1877, pp. 329–331.

6. R. M. ROBINSON, "Mersenne and Fermat numbers," *Proc. Amer. Math. Soc.*, v. 5, 1954, pp. 842–846.

7. J. L. SELFRIDGE & ALEXANDER HURWITZ, "Fermat numbers and Mersenne numbers," *Math. Comp.*, v. 18, 1964, pp. 146–148.