Computing Fibonacci Numbers using the Chinese Remainder Theorem

Brendan J. Babb

August 11, 1989

The goal of this paper is to calculate Fibonacci numbers using the Chinese Remainder Theorem (CRT for short). The reason for using the CRT is that there are several properties known about the Fibonacci numbers mod a prime[1] . Iwas working with Paul Cull and David Holloway who had just completed a paper on computing Fibonacci Numbers quickly. I asked them if they had looked into the CRT and they said they had glasnced at it, but for some reason, it was too slow. Paul said they hadn't looked into it too deeply, and for me to take a closer look anyway. I will attempt to:

- Explain The Chinese Remainder Theorem.
- Define Fibonacci numbers recurrence.
- Cycle lengths of Fibonacci numbers mod p.
- Minimal cycle lengths.
- Cycles mod a Fibonacci number.
- Closed form for remainder.
- Consecutive Fibonacci numbers are coprime.
- Inverses.
- Formula.
- Cutting of multiplications
- Order of Algorithm.
- General recurrence.
- fastest known ways so far.
- Possibilities.

[1] Boyd's, Wall's, Brother Ted's paper

I assumed the CRT was known, but when talking to people they always said ... now which part do you mulltiply by the remainder ..., so in order to save you the time of looking it up in your old number theory textbook I give the following.

The Chinese Remainder Theorem. Let $m_1, m_2, \ldots, m_t$ be pairwise relatively prime positive integers. Then the system of congruence

$$\begin{aligned} X &\equiv r_1 \quad \text{mod} m_1 \\ &, \\ X &\equiv r_2 \quad \text{mod} m_2 \\ &, \\ &\vdots \\ X &\equiv r_t \quad \text{mod} m_3 \\ &, \end{aligned}$$

has a unique solution modulo $M = m_1 m_2 \cdots m_t$. To find the unique solution let

$$M_s = M/m_k = m_1 m_2 \cdots m_{k-1} m_{k+1} \cdots t \cdot$$

Since $m_1, \cdots m_t$ are pairwise prime then we know $(M_k, m_k) = 1$. From this we can find the inverse i_k of M_k modulo m_k, so that $M_k m_k \equiv 1 \pmod{m_k}$. This gives us:

$$X \equiv r_1 M_1 i_1 + r_2 M_2 i_2 + \cdots + r_t M_t i_t \pmod{M}.$$

Let u_n denote the *nth* Fibonacci number of the sequence $u_0 = 1, u_1 = 1, u_n = u_{n-1} + u_{n-2}$. If you take the sequence $v_i \equiv u_i \pmod{p}$, you eventually a return to your two starting values, thus forming a periodic cycle of period d. Refer to papers by Wall and Brother Alfred's for complete details. This is the motivation for using CRT, since there exist cycles that are less than p, thus forming an incomplete residue class. Having a periodic cycle allows us to look at the index of the Fibonacci number modulo the cycle length d, $u_{\beta d+n} \equiv u_n \pmod{p}$. So if we had a table of the first d remainders in the cycle then for any x we only need to find $i \equiv x \pmod{d}$ and look up the ith remainder. There is a limit though, because if we want to calculate u_X exactly we can only calculate values using CRT that are less then M. This causes a problem since u_n grows exponetially. Rounding Binet's formula gives you

$$u_n = \frac{1}{\sqrt{5}} \lfloor \alpha^n \rfloor$$

$$\alpha = \frac{1 + \sqrt{5}}{2}$$

Therefore I was looking for large p with very small cycles. I wrote a program to calculate cycle length of u_n mod i an integer, not a prime, since it was easier to program a simple for loop then a routine to generate the primes. This actually proved an advantage. As I wrote down the i that had a high magnitude to cycle length value, I noticed the numbers 144, 233, 377. Aha! These were the Fibonacci numbers. It seems very obvious now, that my best bet was to use the Fibonacci numbers. The cycle would hit a zero mod u_n at the n the remainder. If n is even then $d = 2n$, if n is odd then $d = 4n$. It turns out that the cycles generated by Fibonacci numbers moduloa fibonacci number are quite predictable and after a while it was easy to write a somewhat closed form for the remainder. For n even the sequence goes

$$\underbrace{1, 1, u_3, u_4, \cdots, u_{n-2}, u_{n-1}, 0, u_{n-1}, -u_{n-2}, u_{n-3}, un-4, \cdots, -1, 1, 0}_{2n}$$

If n is odd then the sequence goes as follows:

$$1, 1, u_3, u_4 \cdots, u_{n-2}, u_{n-1}, 0, u_{n-1}, -u_{n-2}, u_{n-3}, -u_{n-4}, \cdots, 1, -1, 0, -1, -1, -u_3, -u4, \cdots, -u_{n-2}, u_{n-1}, 0,$$

If you use the fact that $u_{-n} = -1^{n+1}u_n$ it becomes more lucid. The cycle starts with the Fibonacci numbers then goes up to u_{n-2}. The next term is $u_{n-2} - u_{n-1} \equiv u_{n-1}$ then that repeats and we just get the series going back down with alternating sign. When n is even the cycle hits $\cdots, 2, -1, 1, 0, 1, 1, \cdots$ where it repeats again. When n is odd the cycle misses on its way back down the first time hitting $\cdots, -2, 1, -1, 0, -1, -1, \cdots$. This makes the next $2n$ elements equal to the negative of the first $2n$ elements. So on its way back after $3n$ it hits $\cdots, 2, -1, 1, 0, 1, 1, \cdots$. This allows for the closed form that follows.

$$j = x \pmod{n} \tag{1}$$

$$k = \lfloor \frac{x}{n} \rfloor \bmod 2 \tag{2}$$

$$l = (\lfloor \frac{x}{2n} \rfloor \pmod 2)(n \bmod 2) \tag{3}$$

$$F_x \equiv -1^{k(j+1)+l} F_{kn+(-1)^k j} \bmod F_n \tag{4}$$

In the first equation j repesents the position in the n cycle. We actually only need to know the first n Fibonacci numbers because the rest in the cycle are just those numbers, positive or negative. Equation 2 just checks the parity of the cycle. If it is zero then we just have the positive sequence of the Fibonacci numbers. If k is one then we are descending, and alternating signs. In equation 3, l is

the parity for the odd cycle. In the first part of l, if l is 0 then we are just doing a regular $2n$ cycle, but if l is one then we are negative the first $2n$ cyclle. We multiply this by the parity of n, since we have a $2n$ cycle for n even, and therefore don't need to woory about a negative $2n$ cycle. Equation 4 takes care of all these parities and gives the correct remainder. Make sure to take modulo least positive residue in equations 1,2,3. This is because most routines to calculate Fibonacci numbers, don't take into consideration negative indexed Fibonacci numbers. Equations (1-4) also work for the generalized sequence $R_{[k,n]} = kR_{[k,n-1]} + R_{[k,n-2]}$,$R_{[k,0]} = 0, R_{[k,1]} = 1$.

We have taken care of finding the remainders. So now we need to choose coprime Fibonacci numbers. I chose to look at three consecutive Fibonacci numbers, we know $(u_n, u_m) = u_{(n,m)}$, and that the gcd of three consecutive integers is at most 2. So the gcd of three consecutive Fibonacci numbers is either u_1 or $u_2 which both equal 1. So we have three pairwise coprime numbers.$

Now we need to worry about the size of the numbers we can compute, it has to be less than M.

$$M = \prod_{j=n}^{n+2} u_j$$

But now we can use Binet's aprroximation to get a rough estimate of the sizes.

$$M \prod_{j=n}^{n+2} \frac{1}{\sqrt{5}} \alpha^j = \frac{1}{(\sqrt{5})^3} \alpha^{3n+3} \approx \frac{1}{\sqrt{5}} \alpha^{3n} \approx u_{3n}$$

As a result we can calculate Fibonacci numbers in the range 0 to $3n$. Of course it would be ridiculous to calculate Fibonacci numbers in the range 0 to n+2 since we need to know those numbers already to find the remainders.

Now we come to the problem of calculating the inverses i_1, i_2, i_3. We would normally have to solve $m_n x \equiv 1 \bmod u_n, but in the case of three consecutive Fibonacci number$ $-1^n \bmod u_n$
$i_2 \equiv -1^{n+1} \bmod u_{n+1} i_3 \equiv -1^{n+1} \bmod u_{n+2}(5)$This makes finding the inverses a constant time process.

Now without loss of generality, assume n is even then the formula for calculating u_X is:

$u_X \equiv r_1 u_{n+1} u_{n+2} - r_2 u_n u_{n+2} - r_3 u_n un + 1 \bmod u_n u_{n+1} un + 2 = M(6)$We can factor out a $u_{n+1} to get : u_X \equiv un + 1(r_1 u_{n+2} - r_3 un) - r_2 u_n u_{n+2} \bmod M(7)$Now use the fact that $u_{N-1} u_{N+1} = u_N^2 + -1(N+1$ to get:

$$- r_2 u_n u_{n+2} = -r_2 u_{n+1}^2 + r_2 u_X = u_{n+1}(r_1 u_{n+2} - r_2 u_{n+1} - r_3 u_n) + r_2 \quad (8)$$

And finally replace $r_1 u_{n+2} = r_1(u_{n+1} + u_n) = r_1 u_{n+1} + r_1 u_{n+1}$ giving us:

$$u_X \equiv u_{n+1}((r_1 - r_2)u_{n+1} + (r_1 - r_3)u_n) + r_2 \tag{9}$$

The order, $\mathcal{O}$, of the calculation given you have all the values for $u_1, \cdots, u_{frac\{X}3$ is $\mathcal{O}(\frac{\Delta}{3}\gamma^{\epsilon}\backslash^{\epsilon})$ for $\gamma = \log\alpha$. This is the order of the bit operations where γn is the number of bits in u_n. *Thisisbasedon Binet'saprroximation.*

In cocclusion I showed that using the CRT allows you to calculate Fibonacci numbers rather quickly. I also came up with a closed form for the remainder of Fibonacci numbers modulo another Fibonacci number. Further directions would be to write the routine recursively and also to try using four remainders, or maybe five and find out if there our ways to cut multiplicatioins, and find easy inverses.

Bibliography

Alfred, Brother U. "Relation of Zeros to Periods in the Fibonacci Sequence Modulo a Prime." American Mathematical Monthly, vol. 71, 1964, pp. 897-899.

Bong, Nguyen-Huu. "A Class of Numbers Related to Both the Fibonacci and Pell Numbers." Fibonacci Numbers and Their Applications. Eds. Andreas N. Philippou, Gerald E. Bergum and Alwyn F. Horadam. Boston: D. Reidel Publishing Company, 1986, pp. 9-38.

Cull, Paul and James L. Holloway. "Computing Fibonacci Numbers Quickly." Oregon State University, Corvallis, 1989, unpublished.

Rosen, Kenneth H. Elementary Number Theory and It's Application. New York: Addison-Wesley Publishing Company, 1988.

Vorobyov, N.N. The Fibonacci Numbers. Boston: D.C. Heath and Company, 1963.

Wall, D.D. "Fibonacci Series Module M." American Mathematical Monthly. vol. 67, 1960, pp. 525-532.