A Better Chebyshev Algorithm

William Boyles

July 22, 2021

1 Definitions

Definition 1. *$U(n,x)$ is the degree n Chebyshev polynomial of the second kind and is defined by the following recurrence relation:*

$$U(n,x) = \begin{cases} 1 & n = 0 \\ 2x & n = 1 \\ 2xU(n-1,x) - U(n-2,x) & n \geq 2 \end{cases}.$$

Definition 2. *$f(n,x)$ is $U(x/2)$ over the field $GF(2)$. So, it is defined by the following recurrence relation where all operations happen mod 2:*

$$f(n,x) = \begin{cases} 1 & n = 0 \\ x & n = 1 \\ xf(n-1,x) - f(n-2,x) & n \geq 2 \end{cases}.$$

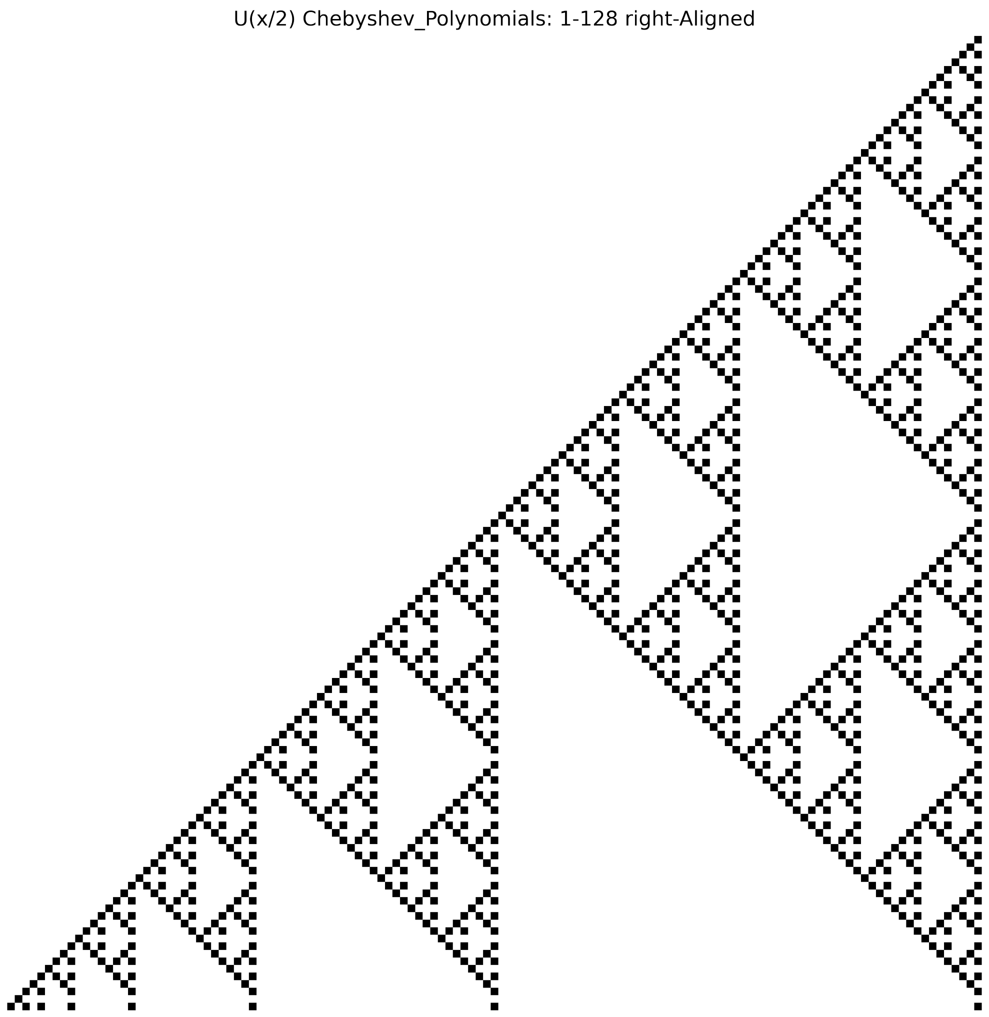
It's known that the nullity, $d(n)$ of an $n \times n$ *Lights Out* board is the degree of $\gcd(f(n,x), f(n,1+x))$. So, one needs to be able to calculate $f(n,x)$ efficiently to calculate $d(n)$ efficiently. Ideally, the algorithm to calculate $f(n,x)$ would not require also calculating $f(0,x) \ldots f(n-1,x)$. We present such an algorithm which takes $\mathcal{O}(n)$ time and space.

2 Investigating Patterns

Since $f(n,x)$ is a polynomial over $GF(2)$, all of its coefficients are either 1 or 0. So, a sensible way to visualize $f(n,x)$ would be to color squares black or white to represent the coefficients. For example,

$$f(5,x) = 1x^5 + 0x^4 + 0x^3 + 0x^2 + 1x + 0 = ■□□□□■□.$$

If we plot out $f(0,n)$ through $f(128,n)$ in the same way, aligning coefficients of the same degree, we get the following image.



This looks exactly like the right half of a Sierpinski Triangle rotated a counter-clockwise quarter-turn. However, the the length of each row in the Sierpinski Triangle grows by 1, and in our image, lengths grow by 2. So, we might suspect we are only seeing the even indexed rows of the triangle. It's well-known that the odd values of Pascal's Triangle form Sierpinski's Triangle. So, we might suspect that we can calculate coefficients of $f(n,x)$ by looking at the parity of binomial coefficients.

Conjecture 1. *Let $f(n,x)_i$ be the ith entry in the coefficient list of $f(n,x)$ so $f(n,x)_0$ is the highest degree term and $f(n,x)_n$ is the units term. Let K be the integer such that $2^k-1 \geq n$ and $S = 2(n-K)$. Then*

$$f(n,x)_i = \begin{cases} 1 & \binom{2i+S}{i+S} \text{ is odd} \\ 0 & \text{otherwise} \end{cases}.$$

3 Parity of Binomial Coefficients

Here's a neat fact about the divisibility of binomial coefficients.

Theorem 1 (Kummer)**.** *Let p be a prime, n and m positive integers. Then the highest power of p that divides $\binom{n+m}{m}$ is the number of carry operations when adding n and m in base p.*

Since we're concerned with base 2, we can give a more specific version of this result.

Corollary 1. $\binom{n+m}{m}$ *is odd if and only if the bitwise AND of n and m, is 0.*

This is an $\mathcal{O}(1)$ divisibility test, so we have an $\mathcal{O}(1)$ to calculate $f(n,x)_i$.

4 The Algorithm

Putting all the pieces together, here is the algorithm to calculate the coefficient list of $f(n,x)$.

```
from math import ceil, log2

def binomial_parity(n: int, m: int) -> int:
    return 0 if ((n-m) & m) else 1

def f(n: int) -> list[int]:
    K = (1 << ceil(log2(n + 1))) - 1
    S = 2*(K - n)

    return [binomial_parity(2*i + S, i + S) for i in range(n+1)]
```