The Twin Prime Conjecture

By Robert Jordan

Abstract

The Twin Prime Conjecture (TPC) posits that there are infinitely many pairs of primes *pn*and *pn+1* such that *pn+1* – *pn*= 2. The work of recent mathematicians, notably Yitang Zhang and an internet collaborative project that improved upon his results, has proven that there are infinitely many primes differing by bounded amounts; Zhang’s results originally showed that this bound was less than 70 million, and this was narrowed down to 246 in 2014, and it is estimated that this can be further narrowed down to 12 or 6 given certain assumptions which are as yet unproven. In this paper, I will outline a method that focuses on Twin Primes specifically (i.e., where the bound is exactly 2), and will show that they must occur infinitely often.

Hexadjacents and Their Relation to Primes

For the purposes of this paper, a “hexadjacent”, or “hexa”, is an integer in the form of 6*x* ± 1, where *x* ℕ. An equivalent definition is that a hexa is any integer which has neither 2 nor 3 as a factor. The value of *x* is called the “sextand” of the hexa in question. For example, 11 = 6(2) – 1, therefore 11 is a hexa, and its sextand is 2.

Although this definition is intentionally general enough to include negative integers, the focus of this paper is on positive hexas greater than 1, and so this is the subset that should be assumed when the paper refers to “hexas”.

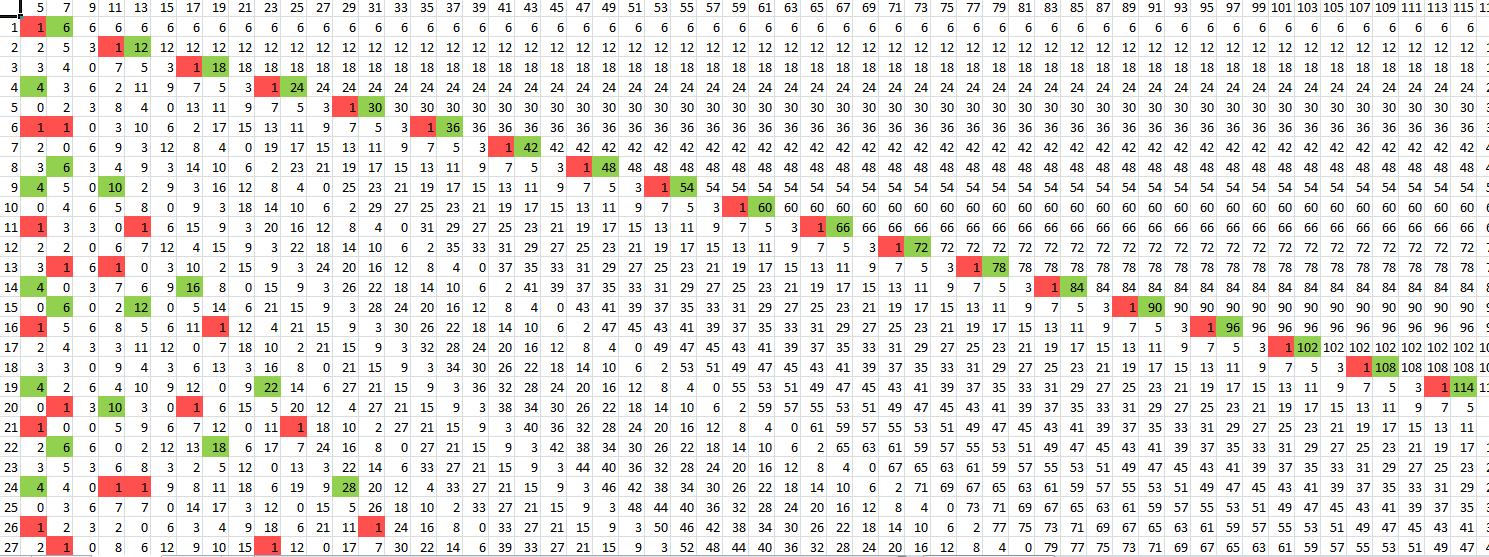
**- Proposition 1: All primes greater than 3 are hexas**

It is simple to prove that all primes greater than 3 are hexas. All integers can only be in one of six forms: 6*x* + 0, 6*x* + 1, 6*x* + 2, 6*x* + 3, 6*x* + 4, 6*x* + 5. The first case, 6*x* + 0, is divisible by 6, and so cannot be prime. The second and fourth cases are both divisible by 2, and also cannot be prime, and nor can the third case, which is divisible by 3, leaving hexas as the only options for primes. All primes except for 2 and 3 then must be hexas. This also gives rise to the second definition of a hexa.

**- Proposition 2: There are no hexas that have 2 or 3 as a factor, and hexas can only have other hexas as a factor**

This is apparent through the fact that hexas are, by definition, 1 greater or less than a multiple of both 2 and 3, and therefore can themselves be a multiple of neither. From this, it follows that hexas can only have other hexas as factors; for if they had some non-hexa *n* as a factor, then *n* must be divisible by 2 or 3, and therefore the end product must also, in which case it is not a hexa.

Since we know that all primes greater than 3 are hexas, by finding patterns in the divisibility of hexas, we can find out more information about primes. It is possible to create a chart of hexas and sextands; on the following chart, the top row contains all odd integers greater than 3; because all hexas are odd, this is an easy way to create a spreadsheet, but the non-hexas can be ignored (since hexas can only have other hexas as factors). On the left-hand side are the positive integers, which indicate a sextand denoting a multiple of 6. Since primes can only occur around multiples of 6, we need only observe sextands. If the top row is the x-axis and the left column the y-axis, then the cells are filled with the remainder when 6y is divided by x; in other words, the cell is 6y mod x. Since hexas can only have other hexas as factors, what we are interested in is whether 6y ± 1 is divisible by a hexa; by looking at the remainders, this becomes easy to see. If 6y – 1 is divisible by a given hexa *h*, then 6y mod x = 1; if 6y + 1 is divisible, then 6y mod *h* = *h* – 1. The boxes are colored according to this; for a given *h*, if the cell contains 1, it is colored red (to show that 6y - 1 divisible), and if it contains *h* – 1, it is colored green (6y + 1 is divisible).

****

From this, certain patterns emerge. There are several striking features present in this graph, possibly worth noting separately, but there are certain patterns that are pertinent to the Twin Prime Conjecture.

**- Proposition 3: On a given row, the absence of either red or green cells indicates the presence of a prime, and a completely uncolored row represents the index of a pair of twin primes**

This is apparent based on the prescribed conditions for coloring the cells. If the cell is uncolored, neither 6y + 1 nor 6y – 1 are divisible by that hexa; if this is true for all hexas up to that point, then it must be a pair of twin primes. If it can be shown that there are infinitely many of these fully uncolored rows, then that is equivalent to showing the infinitude of twin primes. It is worth noting that the topmost line of colors indicates the hexas themselves, and so the presence of colored cells is trivial; no row will be fully uncolored ad infinitum because of this, but this line serves as the boundary for our search; anything above or to the right of that line is greater than the hexa in question, and so is irrelevant.

**- Proposition 4: The modulo values for a given hexa *h* repeat in a cycle of length *h*, and all values between 1 and *h* appear exactly once. There are exactly two colored cells within a given cycle, and they appear at predictable intervals**

If we consider some value *v*, and what happens when *v* mod *h* is taken, we know that adding *h* to *v* will not change the remainder. In other words, *v* mod *h* ≡ (*v* + *h*) mod *h*. This applies to any multiple of *h*; *v* mod *h* ≡ (*v* + *hy*) mod *h* where *y* is an integer. On the chart above, incrementing the index (going down on the y-axis) is equivalent to adding 6. So, if *i* is a given index and *v* ≡ 6*i* mod *h*, then the next iteration of *v* must appear when we add a multiple of both 6 and of *h*. In other words, *r* = 6*i* mod *h* ≡(6*i + 6*h)mod *h* for some index *i* and some hexa *h*. This can be factored into 6(*i* + *h*) mod *h* , making (*i* + *h*) the next index at which the remainder *r* appears. There can be no earlier appearances because we are restricted to adding 6 every time by observing sextands, and the smallest multiple of both 6 and *h* is obviously 6*h*, since *h* cannot itself be divisible by 6 by the definition of a hexa. This shows that if *r* is the remainder at the index *i*, its next appearance is at *i* + *h*. Since this holds for any values, it must be a definite cycle that repeats itself after *h* iterations; The value at *i* = 1 also appears at *i = h + 1*, the value at *i* = 2 also appears at *i* = *h* + 2, etc. Since the cycle is length *h*, this also means that each value between 0 and *h* - 1 must appear; since there are *h* spots to fill, and numbers greater than *h* are unusable because we are working with remainders, then if it were the case that some number in that range went unused, then some other number must appear more than once within a cycle, which is impossible by the previous statement. This further implies that each cycle has exactly two colored cells, since both 1 and *h* – 1 must appear once each in the cycle.

The locations at which the colored cells appear are predictable, too; if we consider the hexa *h* with sextand *s*, then in its very first cycle (i.e. where *i* [0, *h*)), *h \** 1 appears at *i = s* (*i* being the index listed along the y-axis of the above chart), since that is the definition of the hexa. In other words, *h* = 6*s* ± 1, therefore *h* is adjacent to 6*s*, and therefore *r* ≡ ± 1 mod *h* at *i* = s. We also know that a multiple of *h* must appear at *i* = *h – s*. The proof is as follows:

Let *h* = 6*s* + 1. Prove that 6(*h – s*) ≡ -1 mod *h*.

6(*h – s*) = 6*h* – 6*s* by thedistributive property of integers.

6*h* – (*h* – 1), since 6*s* = *h* – 1 by definition of *h*.

6*h* – (*h* – 1) = 5*h* – 1 ≡ -1 mod *h*

This result shows that 6(*h* – *i*) is adjacent to 5*h*; this is unchanged if h = 6*i* – 1, except the result is 5*h* + 1. An important thing we can derive from this is that we no longer need to multiply the index *i* by 6 before we take the remainder mod *h*; though the actual remainders are different, we now have a means of identifying the colored cells solely using the index itself. The only difference is that instead of looking for cells where 6*i* ≡ ±1 mod *h*, we check for cells where *i* ≡ ± *s* mod *h*.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

hn={n is odd: 6((n+1)/2)-1, n is even: 6(n/2)+1

**- For a given hexa *h*  in the form of 6*s*** **± 1, *hy* ± *s* are the sextands of hexas which are a multiple of *h*, where*****y*  ℕ, *i* ≥ 1**

This is less straightforward than the previous properties. In essence, there is a predictable pattern of multiples of a given hexa *h*. Every hexa has an index *i*; if one takes any positive multiple of *h*, call it *hy*, then *hy* + *I* and *hy* – *i* are the indices of hexas which are divisible by the original hexa *h*.

As an example, 5 = 6(1) – 1, and its index is 1. I say that, for any number *y* that satisfies the above conditions, 6(*5y +* 1) and 6(*5y -* 1) are both adjacent to a multiple of 5. If *y* = 2, then 5*y* + 1 = 11 and 5*y* – 1 = 9; 6(9) +1 = 5 \* 11, and 6(11) – 1 = 5 \* 13.

Although the concept itself is not immediately apparent, its proof is simple:

Let *h* be some hexa and *i* be its index. Let *h* = 6*i* + 1, and let *y* be some integer < 0. The goal, then, how that 6(*hy* + *i*) is adjacent (one more or less than) a multiple of h

6(*hy* + *i*) = 6((6*i* + 1)*y* + *i*)

6(6*iy* + y + *i*)

36*iy* + 6*y* + 6*i*

6*i*(6y) + 6y + 6*i*

(h – 1)(6*y*) + 6*y* + (h – 1)

h(6*y*) + h – 1

h(6*y* + 1) – 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The cyclical nature of multiples of hexas can be exploited to create approximations for the number of twin primes. Going forward, for a given hexa *h*,a sextand *y* is considered “valid with respect to *h*” if neither 6*y* ± 1 are divisible by *h* (i.e. the cell (*y*, *h*) is uncolored in the above chart. Because there are exactly 2 invalid values of *y* in a cycle of length *h*, the probability of any random , the probability of any random *y* being valid with respect to *h* can be approximated fairly well by the function