Collatz Conjecture

William Manning

December 12, 2014

1 Introduction

The Collatz conjecture states that if you take any natural number n and follow two simple rules you will always eventually reach 1. The two rules are as follows:

1. If n is even, divide it by 2 to get n / 2.
2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1.

Or perhaps more clearly:

$$f(n) = \begin{cases} n/2, & \text{if } n \text{ even} \\ 3n+1, & \text{if } n \text{ odd} \end{cases}$$

This process is repeated indefinitely until 1 is reached.

1.1 Proof?

As of yet there is no proof to the conjecture, however there is a $500 prize offered by Paul Erdős and a £1000 prize offered by Sir Bryan Thwaites for anyone able to prove (or disprove) it.
Although there is no definitive proof for all natural numbers, it has been proven that all numbers $n \leq 19 \times 2^{58} \approx 5.48 \times 10^{18}$ will eventually reach 1 [1].

2 Exploring the Conjecture Using Sage

2.1 Simple Test

The following function [2] takes any positive integer and goes through each step, eventually ending at 1:

```
def Test(n):
    i = n
    while i != 1:
        if i % 2 == 0:
            i = i/2
        else:
            i = 3*i +1
    print "Got to 1, test is positive for %s as a starting point." % n
```

Using this test we find that any number that we can put in always gets to 1 eventually. However some numbers take many steps.

2.2 Number of Steps

Next I decided to find how many steps it takes to get to 1. I then plotted the results using the following function [3].

```
t = [0]
for n in range (1, 200):
    i , m = n, 0
    while i != 1:
        m += 1
        if i % 2 == 0:
            i = i/2
        else:
            i = 3*i +1
    t += [m]
bar_chart(t, width=0.1, axes_labels=['Sarting Integer','Number of Steps'])
```

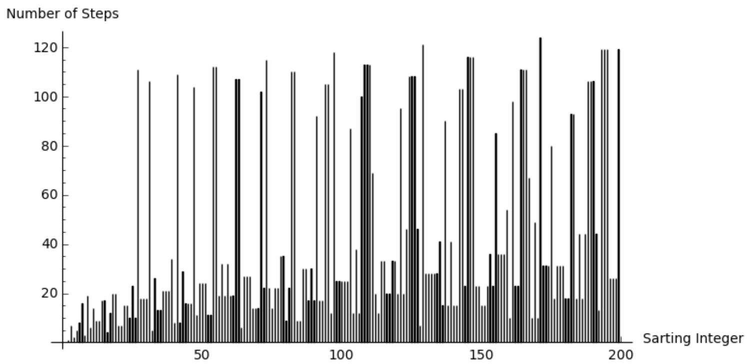

Figure 1: The number of steps it takes for the first 200 integers.

From the graph we can see that there is generally an increase in the number of steps required to get to 1. However, there are many tall peaks which make it difficult to tell what the general change in steps is. So I rewrote the code to ignore any integer where the number of steps needed to get to 1 is greater than 50. This code [4] is shown here:

```
t = [0]
for n in range (1, 200):
    i, m = n, 0
    while i != 1:
        m += 1
        if i % 2 == 0:
            i = i/2
        else:
            i = 3*i +1
    if m >= 50:
        m = 0
    t += [m]
bar_chart(t, width=0.1, axes_labels=['Starting Integer','Number of Steps'])
```

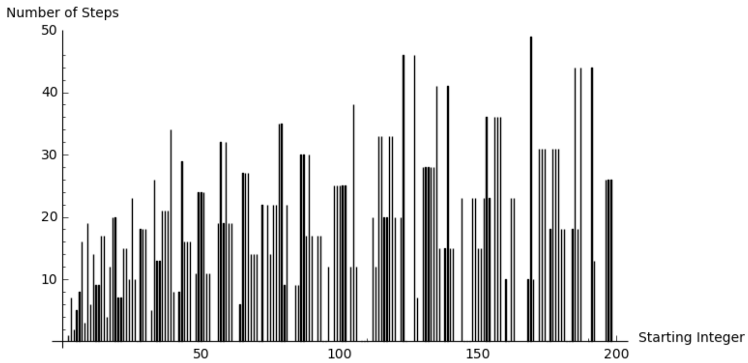

Figure 2: The number of steps it takes for the first 200 integers, without any integers that take over 50 steps to get to 1.

2.3 Average Number of Steps

To make the general increase in number of steps required easier to see I have written a function which takes values in a range and calculates the average number of steps required to get to 1. The code [5] is as follows:

```
def Average(R):
    t, s = 0, 0
    for n in range (1, R):
        i, m, s = n, 0, s + 1
        while i != 1:
            m += 1
            if i % 2 == 0:
                i = i/2
            else:
                i = 3*i +1
        t += m
    Average = float(t/s)
    print Average
```

It shows that as the range increases the average number of steps required also increases.

3 Conclusion

In this report I have explored the Collatz Conjecture showing that the conjecture is true for any number that we can choose and the number of steps involved varies but generally increases.

References

[1] *http://www.wolframalpha.com/input/?i=collatz+conjecture.*

[2] *https://cloud.sagemath.com/projects/9a5c4d48-b96d-468e-b561-ac7489b56b07/files/SimpleTest.sagews.*

[3] *https://cloud.sagemath.com/projects/9a5c4d48-b96d-468e-b561-ac7489b56b07/files/PlotofSteps.sagews.*

[4] *https://cloud.sagemath.com/projects/9a5c4d48-b96d-468e-b561-ac7489b56b07/files/PlotofSteps2.sagews.*

[5] *https://cloud.sagemath.com/projects/9a5c4d48-b96d-468e-b561-ac7489b56b07/files/AverageNumberofSteps.sagews.*