The prime counting function and related subjects

Version 3.1415

Tuesday, April 05, 2005

Patrick DEMICHEL

Abstract:

In this paper I will show the heuristic that I use to demonstrate that the smallest value for x that satisfies the equation $Li(x) < \pi(x)$ is the value 1.397162914e316

I will demonstrate that the fundamental element for finding the exact location of the first crossover and to guaranty that the crossover exists, depends mainly from the number of complex zeros of the zeta function used in the computations

I will show some special regions where $Li(x)$ is quite far from $\pi(x)$ after a crossover

Finally I will propose some conjectures or problems that some of you can try to resolve

Introduction:

The prime counting function $\pi(x)$ is the function giving the number of primes <= (x)

$$\pi(x) = \sum_{p \leq x}$$

There are many ways to compute $\pi(x)$

Exact computations:

**Sieve of Eratosthenes : simple but requires a huge memory and the compute time increases as sqrt(x).
Then this is not usable beyond 1e16 : current record for the Brun constant computation that uses a very similar sieve method. I did the computation; I know how it was hard ☺**

**Meissel method, refined by D. H. Lehmer, Lagarias, Miller, Odlyzko and finally by Xavier Gourdon: it is quite complex, but the method offers a significantly better efficiency over the Eratosthenes Sieve approach.
The current world record for 2003 is only in the range of 1e22 and it cost ~10 days of compute time if run on the best processor of that year. For 1e23 this would require 4 times more CPU and so on assuming computing in extended precision is free**

http://www.utm.edu/research/primes/howmany.shtml

http://numbers.computation.free.fr/Constants/Primes/countingPrimes.html

Approximations:

In 1798, Legendre in "*Essai sur la Théorie des Nombres* " establishes

$$\pi(x) \approx \frac{x}{log(x) - 1.08366}$$

In 1791 Gauss independently was first to suggest that:

$$\pi(x) \approx \frac{x}{log(x)}$$

In a letter to Encke in 1849 Gauss communicated that Li(x) is a much better approximation for π(x) , this is known as the "Prime Number Theorem" or PNT

$$\pi(x) \approx Li(x)$$

with:

$$Li(x) = \int_2^{\inf} \frac{dx}{log(x)}$$

This was independently proved by Hadamard and de la Vallee Poussin in the year 1896

Latter Riemann found a much better approximation for π(x)

$$\pi(x) \approx Li(x) - Li(x^{1/2})/2$$

Riemann, in searching to reduce the error term, established a formula that gives an exact value for π(x), but only if we take into account an infinite number of zeros of the complex zeta function
This result still depends on the proof that all complex zeros of the zeta function lay on the critical line "Riemann hypothesis"

With ρ representing all the complex zeros of the zeta function, and μ(n) representing the Möbius function

$$\pi(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} S(x^{1/n})$$

$$S(x) = Li(x) - \sum_{\rho} Li(x^{\rho}) - ln(2) + \int_{x}^{\inf} \frac{dt}{t(t^2-1)ln(t)}$$

Another simpler and frequently used approximation is R(x) :

$$R(x) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} Li(x^{1/n})$$

For large values of x most terms of S(x) become very rapidly negligible, look at the following slide, a better approximation is :

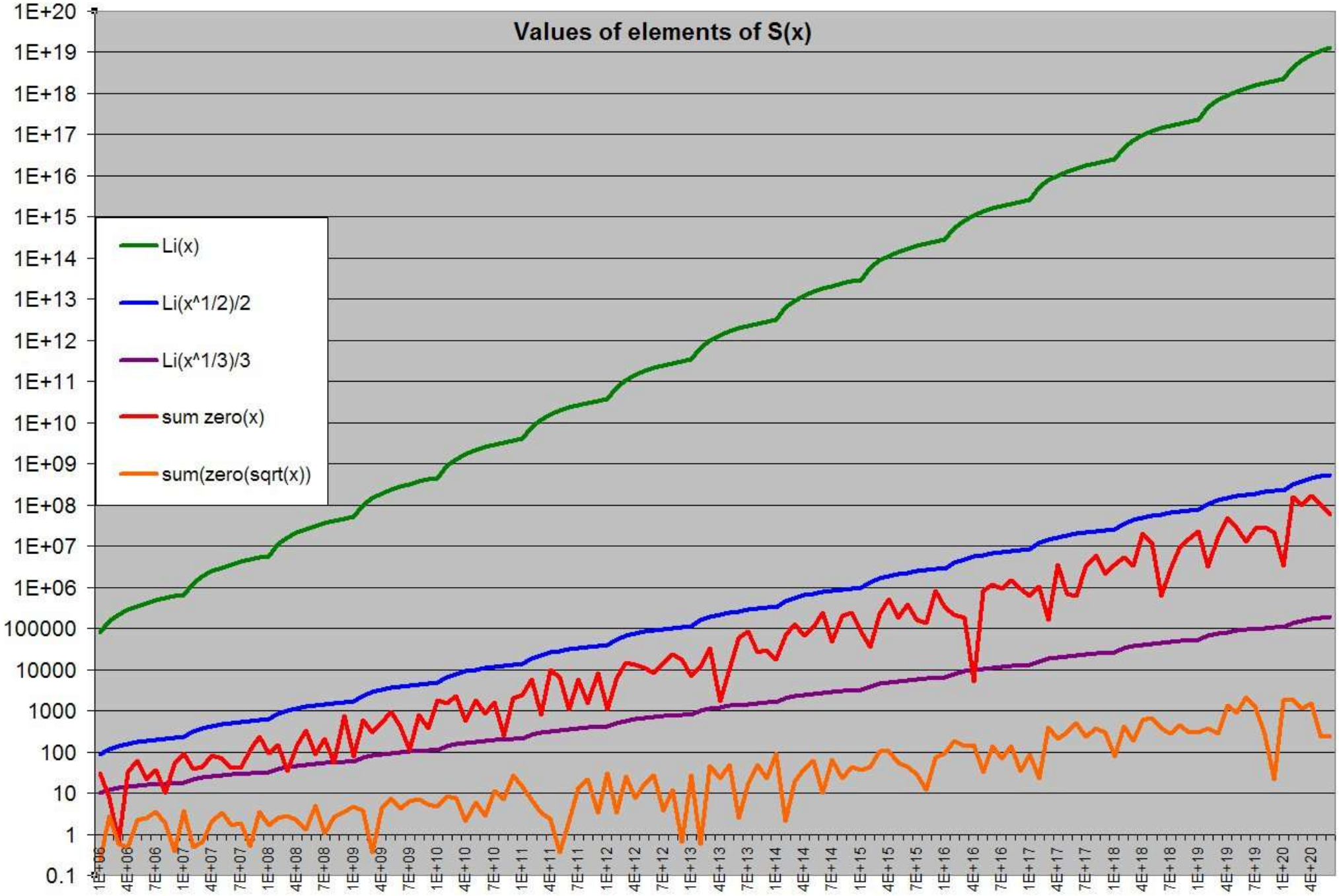
$$\pi(x) \approx Li(x) - Li(x^{1/2})/2 - \sum_{\rho} Li(x^{\rho})$$

This equation is more precise that R(x) but requires the computation of lot of complex Li functions with many zeros of the Riemann zeta function. I will show that the precision of this function improves when we increase the number of zeros. The problem is that the precision increases extremely slowly

The difference between Li(x) and π(x) is:

$$Li(x) - \pi(x) \approx Li(x^{1/2})/2 + \sum_{\rho} Li(x^{\rho})$$

What is the size of the absolute value for the various elements of S(x) when x increase?



The Li functions are always positive. The sums of zeros are either positive or negative, see some slides further

We can observe that $Li(x^{1/3})/3$ and the $\Sigma((x^{\rho})^{1/2})$ become extremely rapidly negligible relatively to $Li(x)$, $Li(x^{1/2})/2$ and $\Sigma(x^{\rho})$

This is even worse for all others terms in the S(x) function

Then it is valid to simply ignore those terms even if we compute with high precision

I should precise that the oscillations on the curves for Li are caused by using a constant step and not a logarithmic step to define x inside a power of 10. In other words, between 10^x and 10^(x+1), the difference between 2 samples is a constant value. If I had used a logarithmic step for x, we could have observed that all Li curves would appear linear, but computing the exact values of pi(x) with integer step is more interesting

Observe also that $Li(x^{1/2})/2$ is always superior to $\Sigma(x^{\rho})$

We could believe that to be true forever. In fact it was believed until 1914 that $Li(x) > \pi(x)$ for all x, until J.E. Littlewood in 1914 proved this was false

In 1933, Skewes gave an upper bound for the first exception: 10^10^10^34
In 1966 Lehman found a first region where we could find a crossover: 1.5926e1165
Te Riele made an improvement in discovering the region 6.658e370
In 2000, Te Riele and Hudson found another crossover in the region 1.39822e316. They suspected that the first crossover would probably be found in the vicinity of this number, or could occur probably only in a limited number of potential regions that they listed

I will confirm their hypothesis by giving the value where the first crossover occurs and showing that the potential regions have in fact no chance to contain a crossover. I say no chance, because I will demonstrate that there exist an infinitively small probability that a crossover occurs there or before 1e316, but only if there would be something extraordinary special in the way that the zeros are organized
The risk seems at first approach close to zero and beyond the compute capacity with current methods, but all people having worked on the zeta function knows how it is surprising, then we need to stay prudent

Lets define :

$$T(x) = Li(x^{1/2})/2$$

$$U(x) = \sum_{\rho} Li(x^{\rho})$$

and

$$V(x) = \frac{U(x)}{T(x)}$$

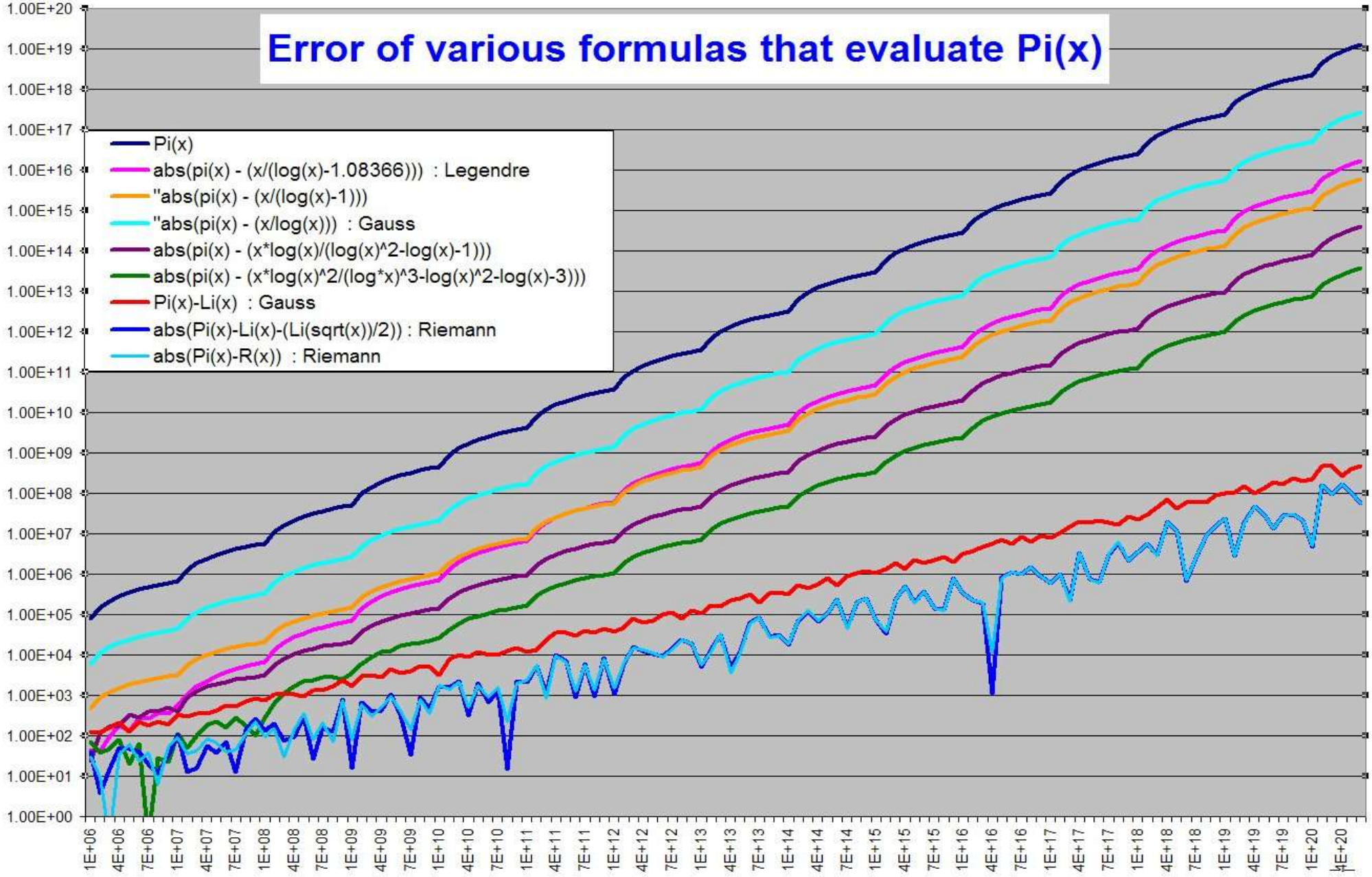
T(x) is always > 0
U(x) is either positive or negative

Then we have 3 possibilities:

U(x) >= 0	**then V(x) >= 0**	**and Li(x) > π(x)**
U(x) < 0 and abs(U(x)) < T(x)	**then 0 > V(x) > -1**	**and Li(x) > π(x)**
U(x) < 0 and abs(U(x)) > T(x)	**then V(x) < -1**	**and Li(x) < π(x)**

In other terms: if we search the crossovers of Li(x) it is sufficient to search for the values of x where V(x) < -1
Remember: in all following slides the crossover will be all the regions where V(x) < -1

Comparison of various functions without using the complex zeros



The formulas of Gauss and Riemann are definitively much better than the family of formulas similar to the formula of Legendre, they give roughly half correct leading digits

Observe that the formula of Legendre is becoming worse than the simpler formula in orange around 1e11

Remember the errors are exact values because I use a table of exact values of $\pi(x)$

With the zeros

Let's look at what happend when we take into account the sum with the complex zeros: U(x)

The first step before computing this sum is naturally to find a list of zeros or to compute this list

Some questions appear immediately:

- **How many zeros do we need?**
- **What is the precision needed for the zeros?**
- **What is the precision needed for the computations?**

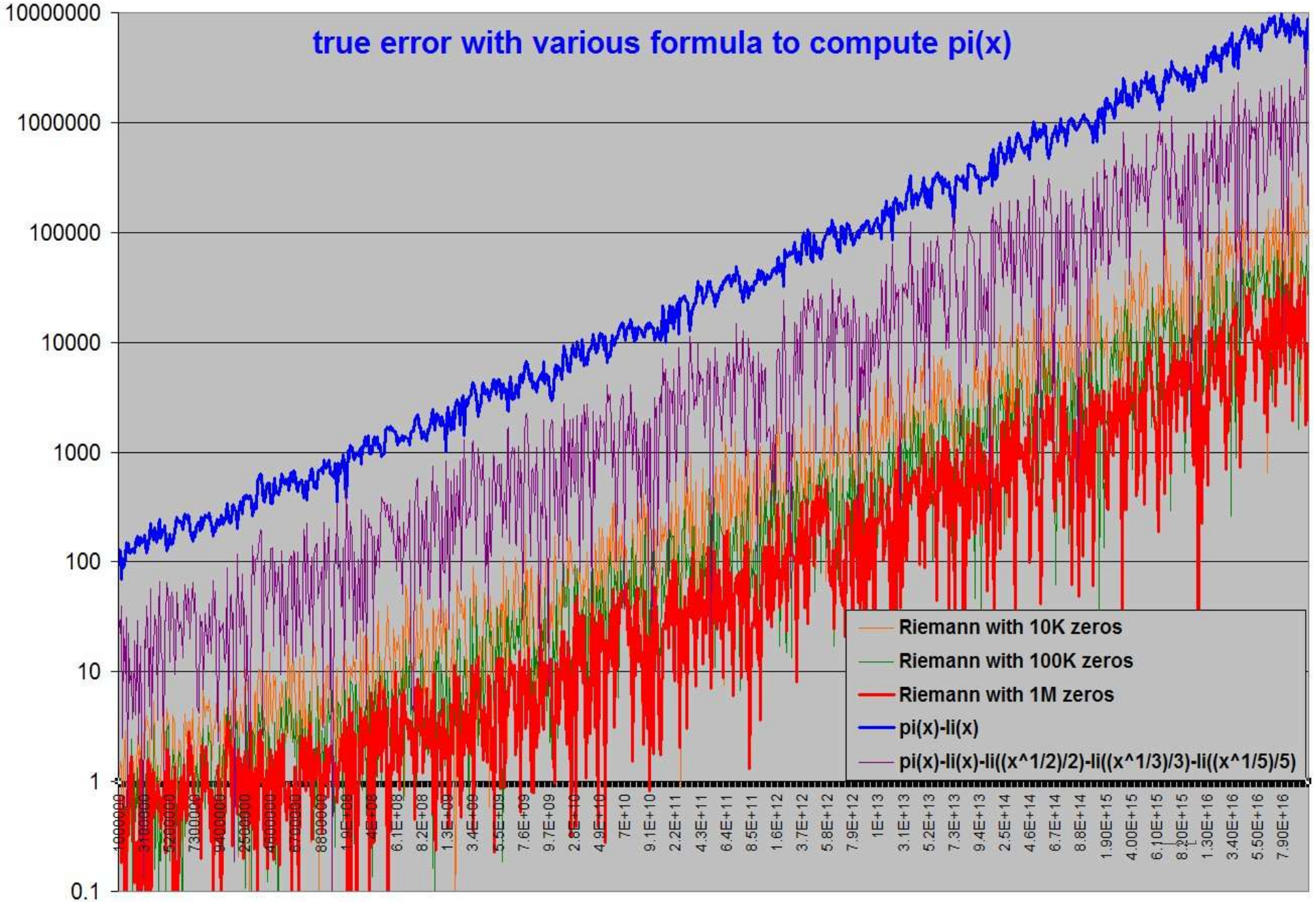
There is one list of zeros available http://www.dtc.umn.edu/~odlyzko/zeta_tables/zeros1, but this list contains only 100K zeros, is it sufficient?

Since I wanted much more zeros to evaluate the benefice of using a very large number of zeros, I used a program developed by Xavier Gourdon and computed 10G zeros with 7+ digits of precision:
Compute time = 3 days, file size = 200Gbytes, this can be reduced to 50Gbytes by various compression tricks

To measure the effect of the precision of the zeros , I created multiple lists of zeros, with the zeros rounded to lower different precisions, this will permit to determine what is the minimal and sufficient precision needed for the computations
It is very important to understand what is the minimal and sufficient precision needed in the computations, because it is very hard to compute the complex zeros in high precision. If we need a huge number of zeros in high precision, it will be impossible to determine with great confidence, where is the first crossover

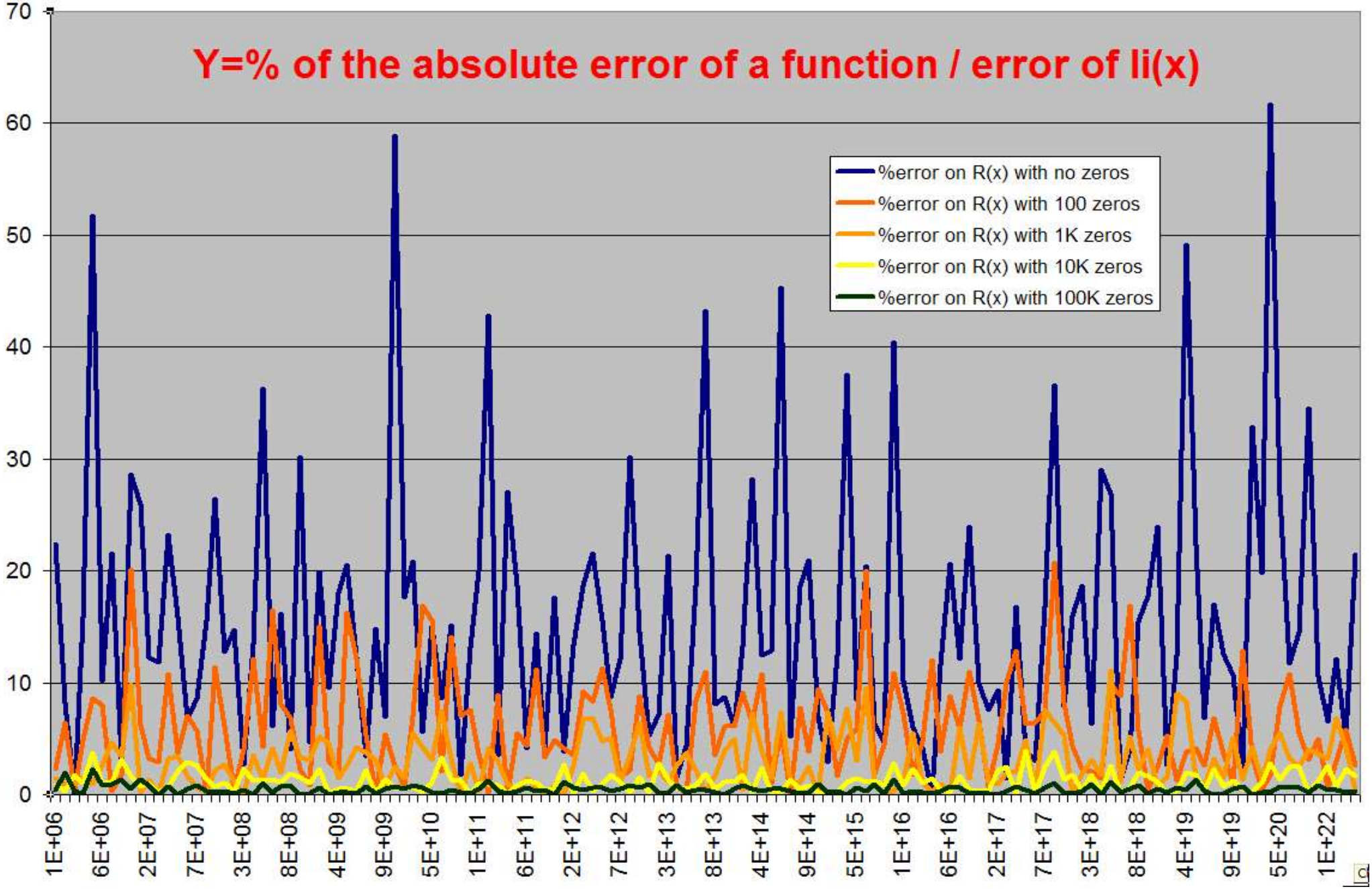
Then the first step is to determine how many zeros are needed to reach a specific level of precision, and if it is possible to predict, what is the precision accordingly to the various factors involved in the computations

To do so, I need to compute many values of Pi(x) in the range 1e6 to 4e20. Since I know the exact value of $\pi(x)$
I can measure the exact error whatever the formula used or the number of zeros taken into account
I should note that this program is another contribution of my friend Xavier Gourdon



We can observe that taking into account the complex zeros improves the precision of π(x)
We can also observe that the more complex zeros we use, the better the precision
Latter in this paper I will show that each time I increase by 10 the number of zeros, I roughly increase the precision by ~2.5 . In others terms the absolute worst error is divided by 2.5 but only in average

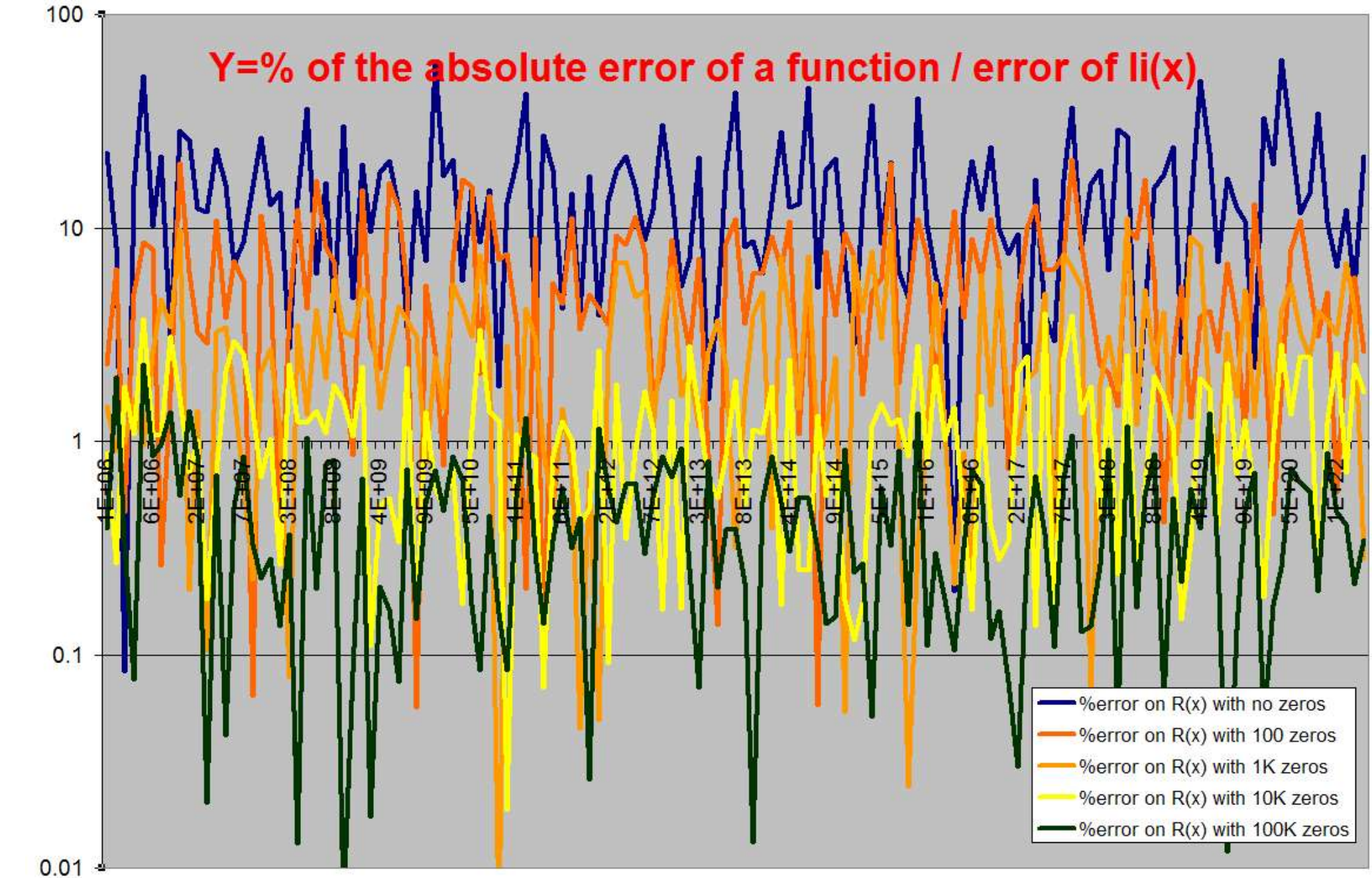
What is the absolute true error in percentage for R(x) with and without the zeros ?



We can observe that R(x) is a neat improvement over li(x), but in many cases the improvement is only a reduction of 50% of the large error $\pi(x)$-li(x)

With the zeros the error is significantly reduced, with 10K zeros the error is far smaller than 1%

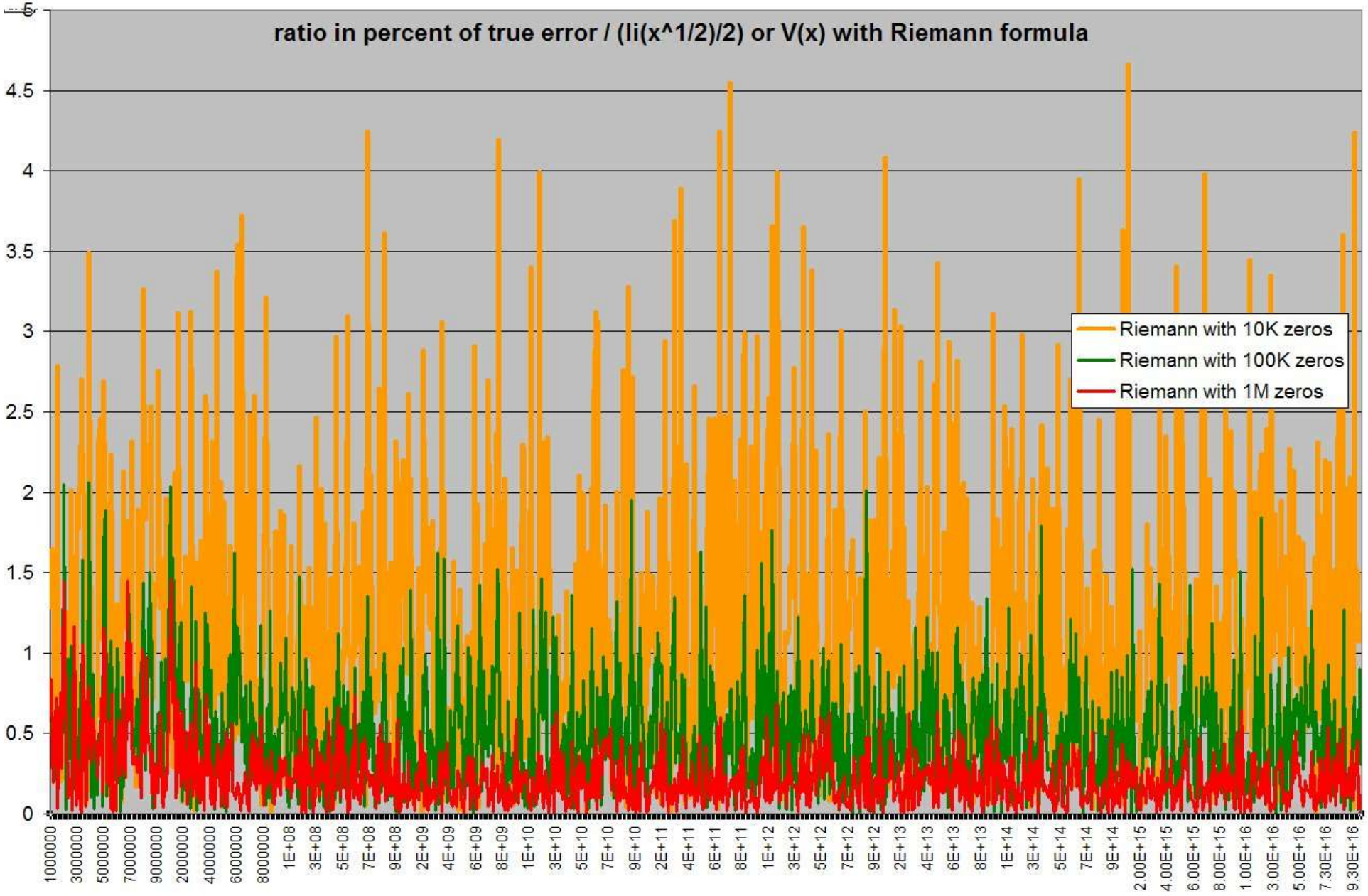
We need a logarithmic Y scale to better see the behavior with many more zeros



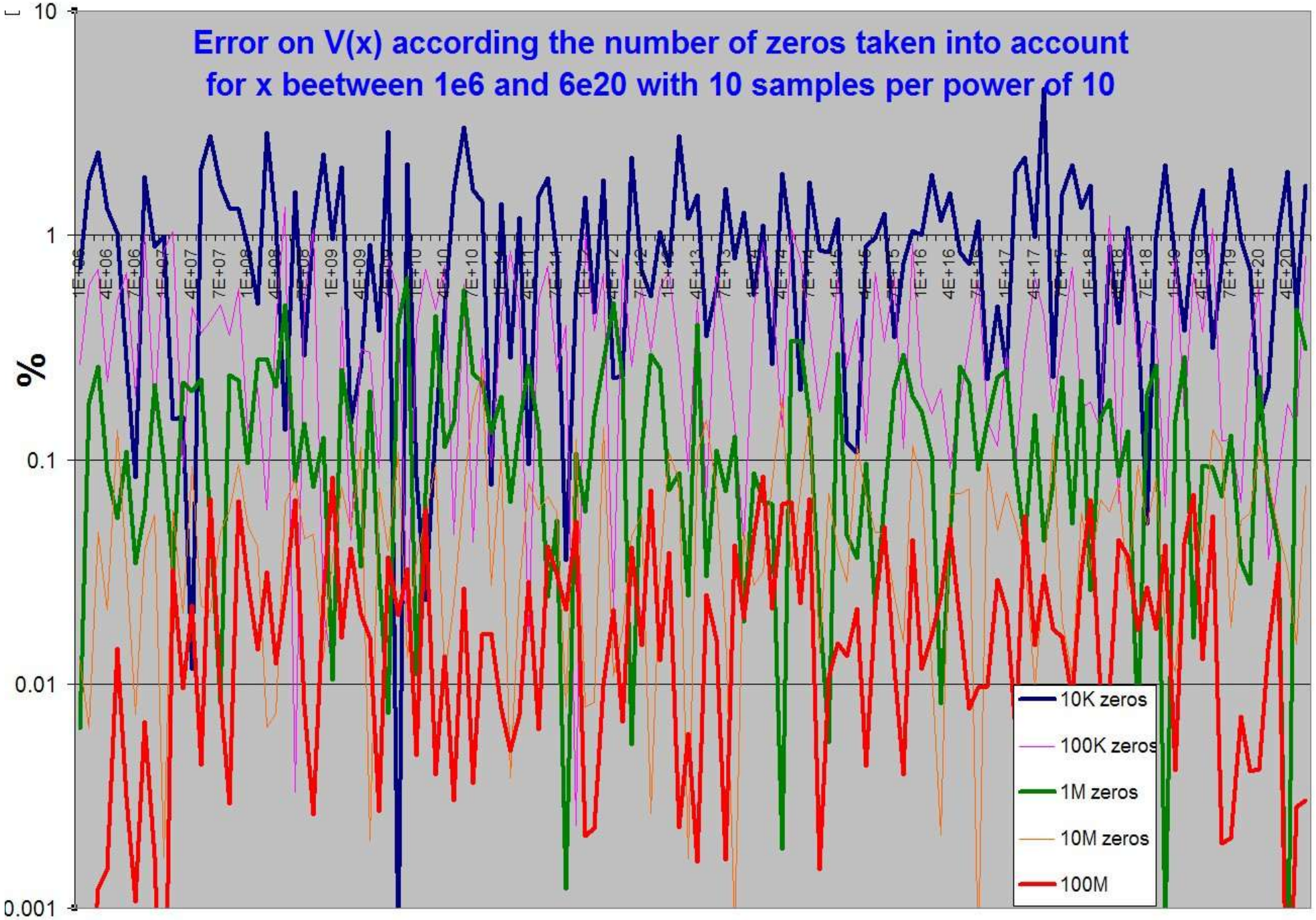
With a logarithmic scale we observe clearly how the precision improve

Remark that more zeros do not give always a better precision. You need far more zeros to be sure that the function is converging to the exact solution. We can observe that we need roughly 100 times more zero to reduce the error by 10

Same analysis but with even more samples and more zeros

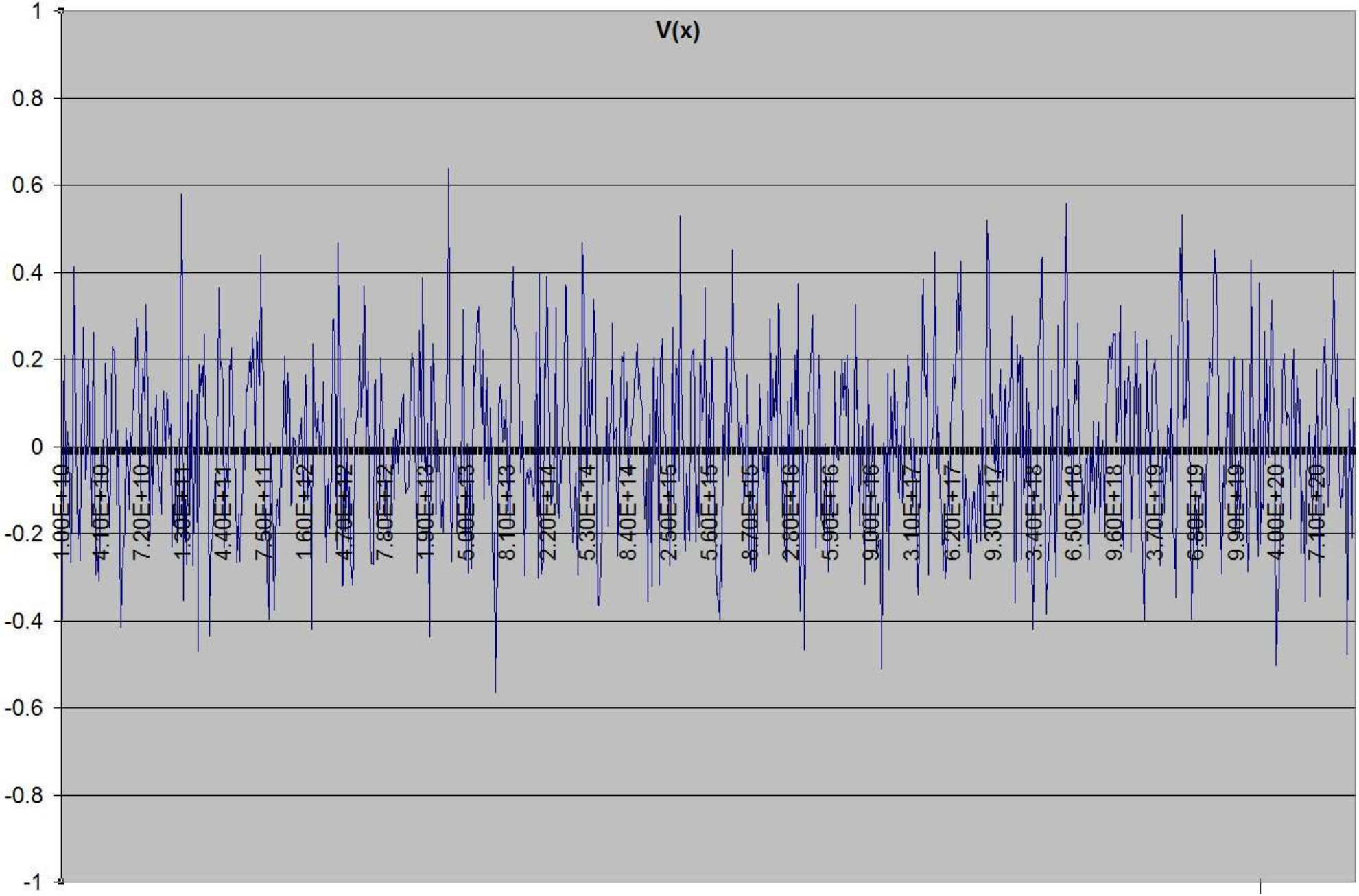


We can observe that the real error is greater with x below 1e8 and 1M zeros, this is because the deleted terms of S(x) still have some minor impact that is observable with lot of zeros. Beyond 1e8 this becomes negligible.
Here also we can observe, that each time I take 10 times more zeros, the error rate is roughly divided by 2.5
In following slide we can observe a similar behavior, but with a logarithmic scale we can see more levels.



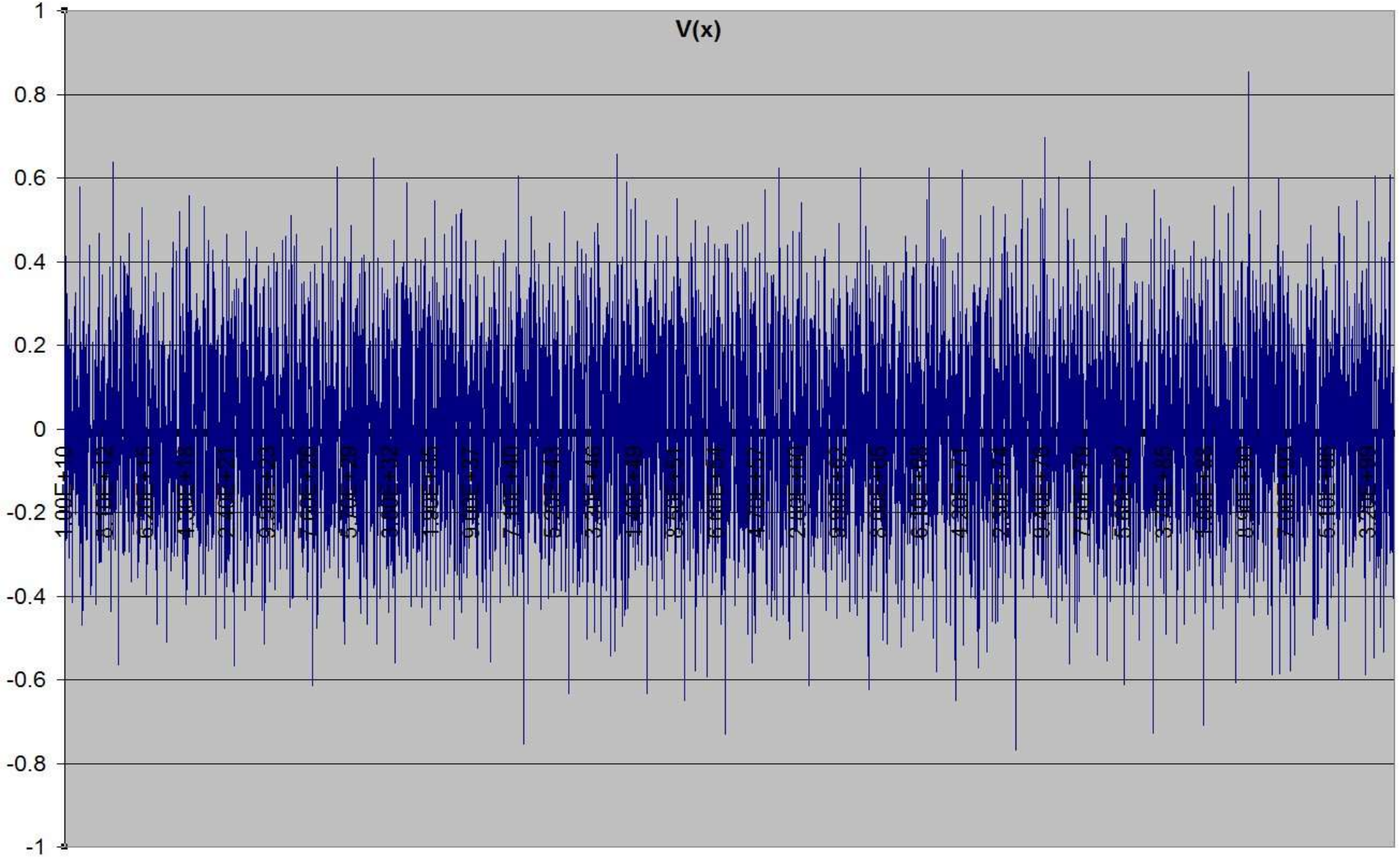
The precision of V(x) seems to be 6% or .06 with 10K zeros , 2.5% or .025 with 100K zeros , 1% or .01 with 1M zeros, 0.4% or 0.004 with 10M zeros, 0.16% or 0.0016 with 100M zeros
We can observe that the precision increases roughly by 2.5 every time I use 10 times more zeros in U(x)

V(x) behavior between 1e10 and 1e21



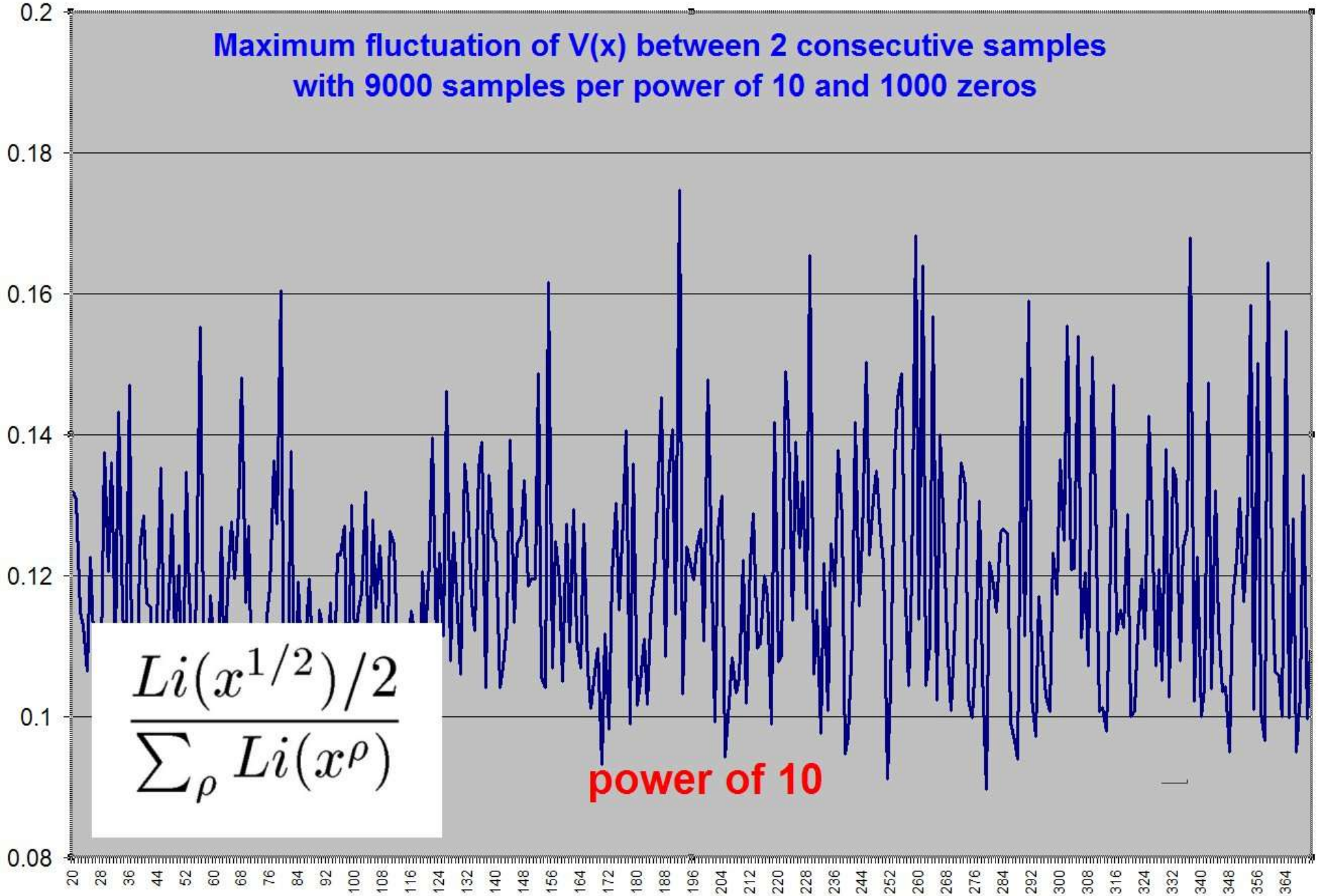
We can observe that V(x) is equally distributed around the x axis but nowhere V(x) is inferior to -0.6
This explain that Li(x) is always superior to Pi(x) up to 10^20, nowhere we approach the critical value for V(x) of -1.0 . In fact we stay quite far, maybe we can expect this behavior to be true up to ∞

Let’s look how V(x) behave up to 10^100



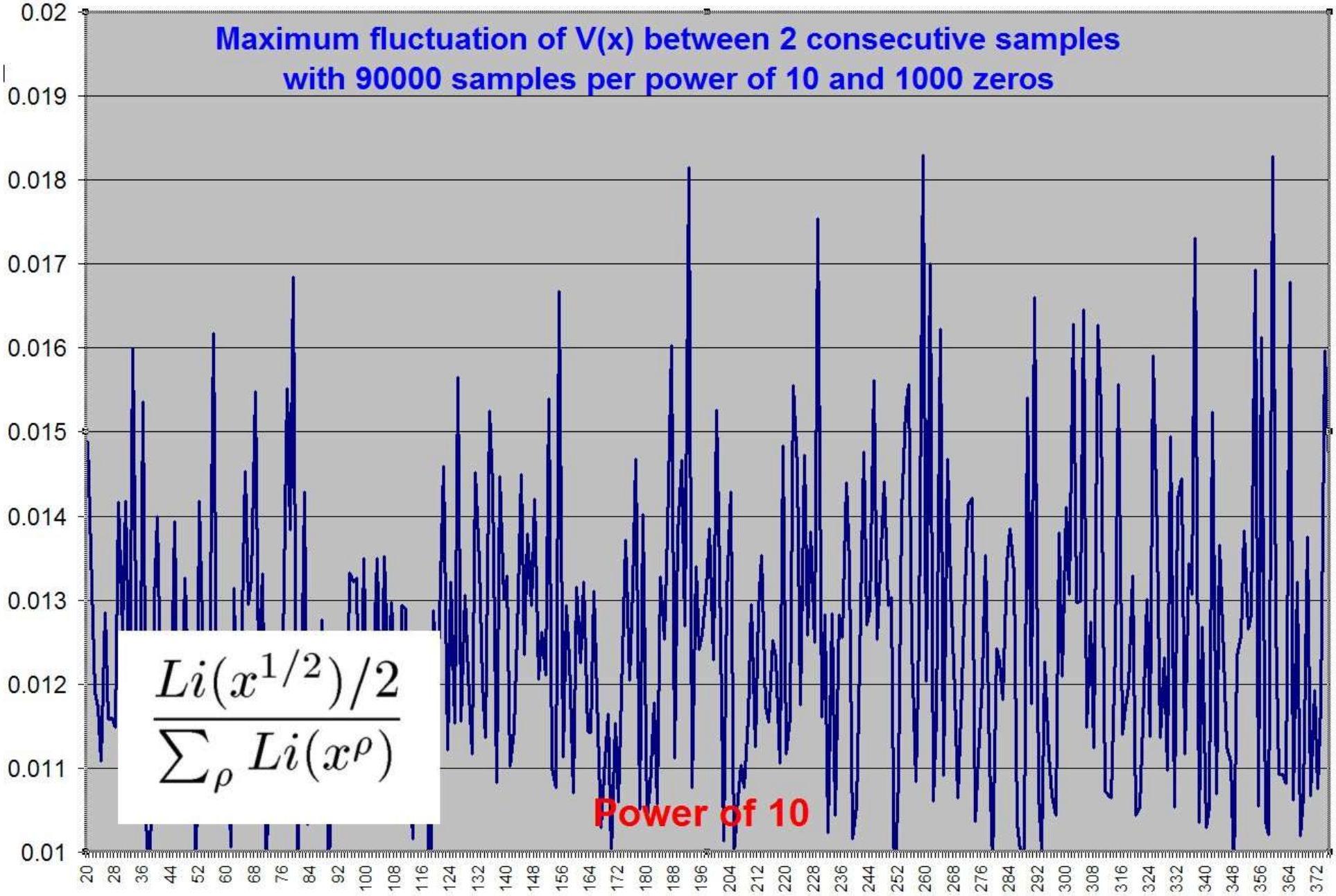
This seems similar up to 10^100, V(x) never goes below -0.8. There seems to be no crossover before 1e100 otherwise V(x) would reach the value -1, but since some values approach -0.8, we need to suspect that it can be possible that V(x) cross the critical line y=-1
The problem is that I have no idea on the precision of the computations
I have also no idea if I have missed an interesting spot by only using 90 samples per power of 10

What I need to determine now, is the largest fluctuation of V(x) in percentage for a specific number of samples per power of 10. This will help to determine the maximum error that I can tolerate



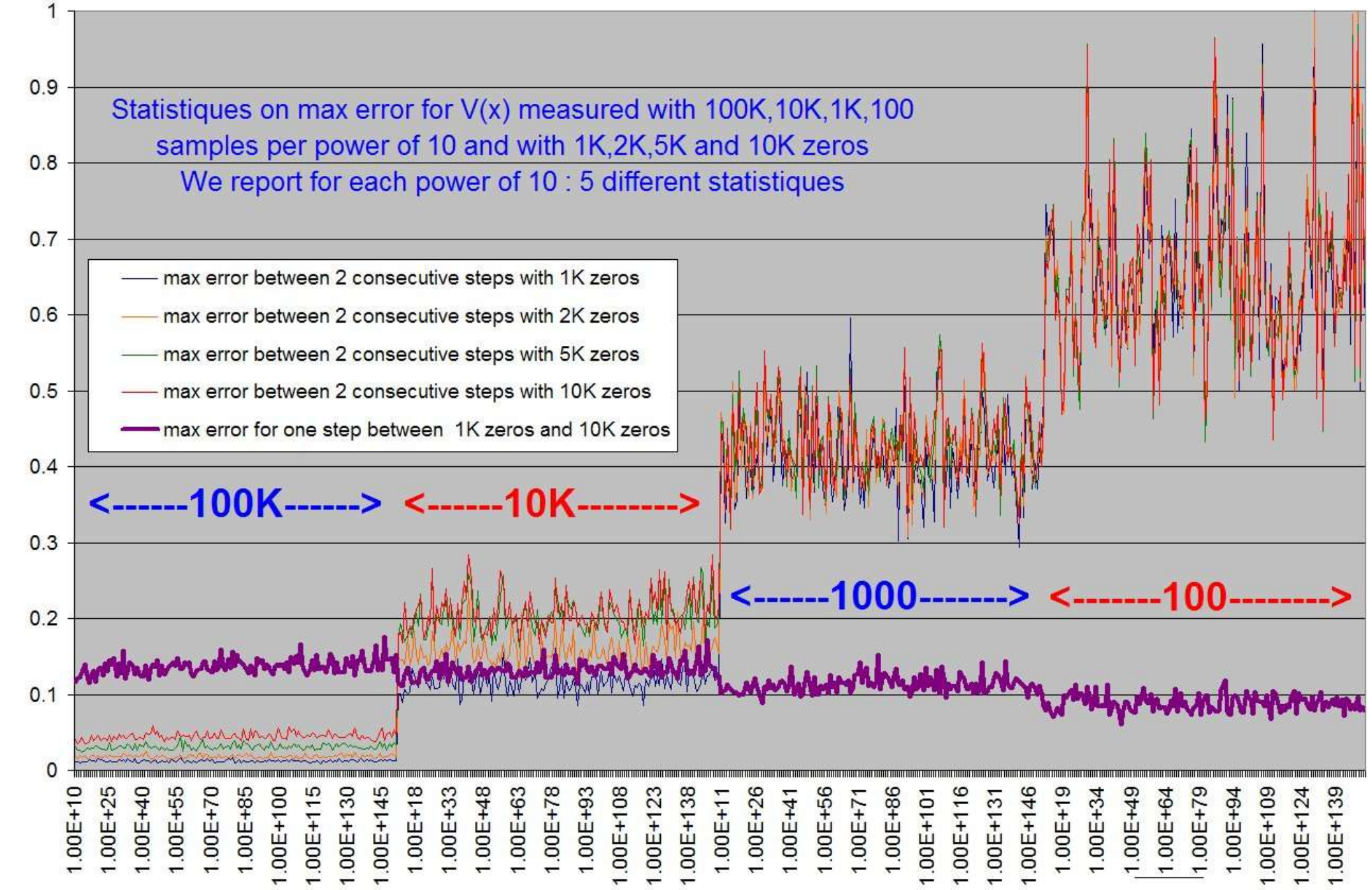
We can observe that the error rate on V(x) seems independent of the value of x between 10^20 and 10^380
We can see that the worst error is below 0.18 or 18% for 10K samples per power of 10 at 1K zeros

And now with 90K samples per power of 10



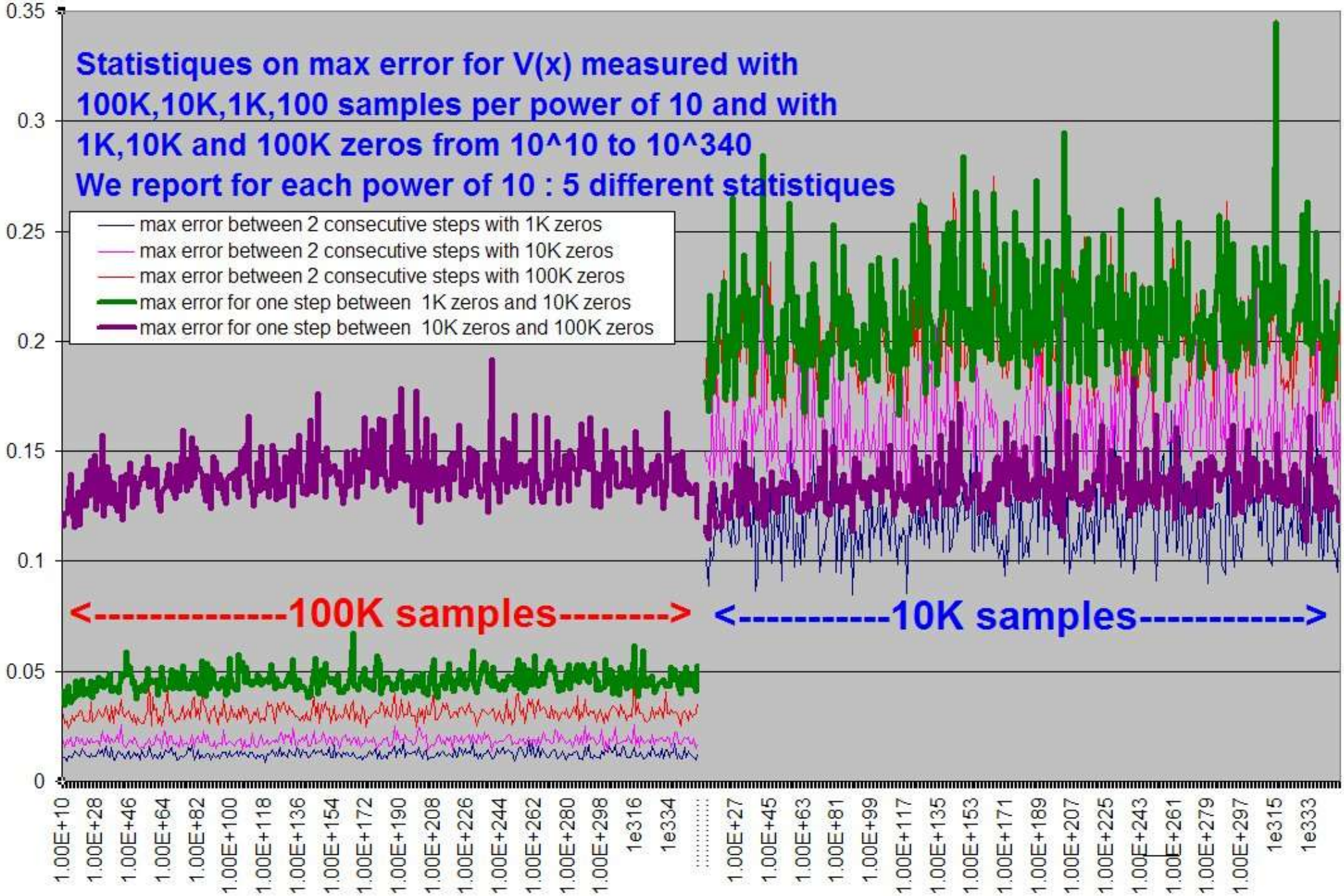
With 9000 samples the largest fluctuations where ~.18 or ~ 20%
With 90000 samples the biggest fluctuations are ~.018 or ~2%
Each time we increase the sampling rate by 10, the largest fluctuations decrease by a factor of 10
This is easy to understand if we observe that the largest fluctuations occur on vertical fronts of V(x)

What is the error rate in changing the sampling rate and the number of zeros together ?



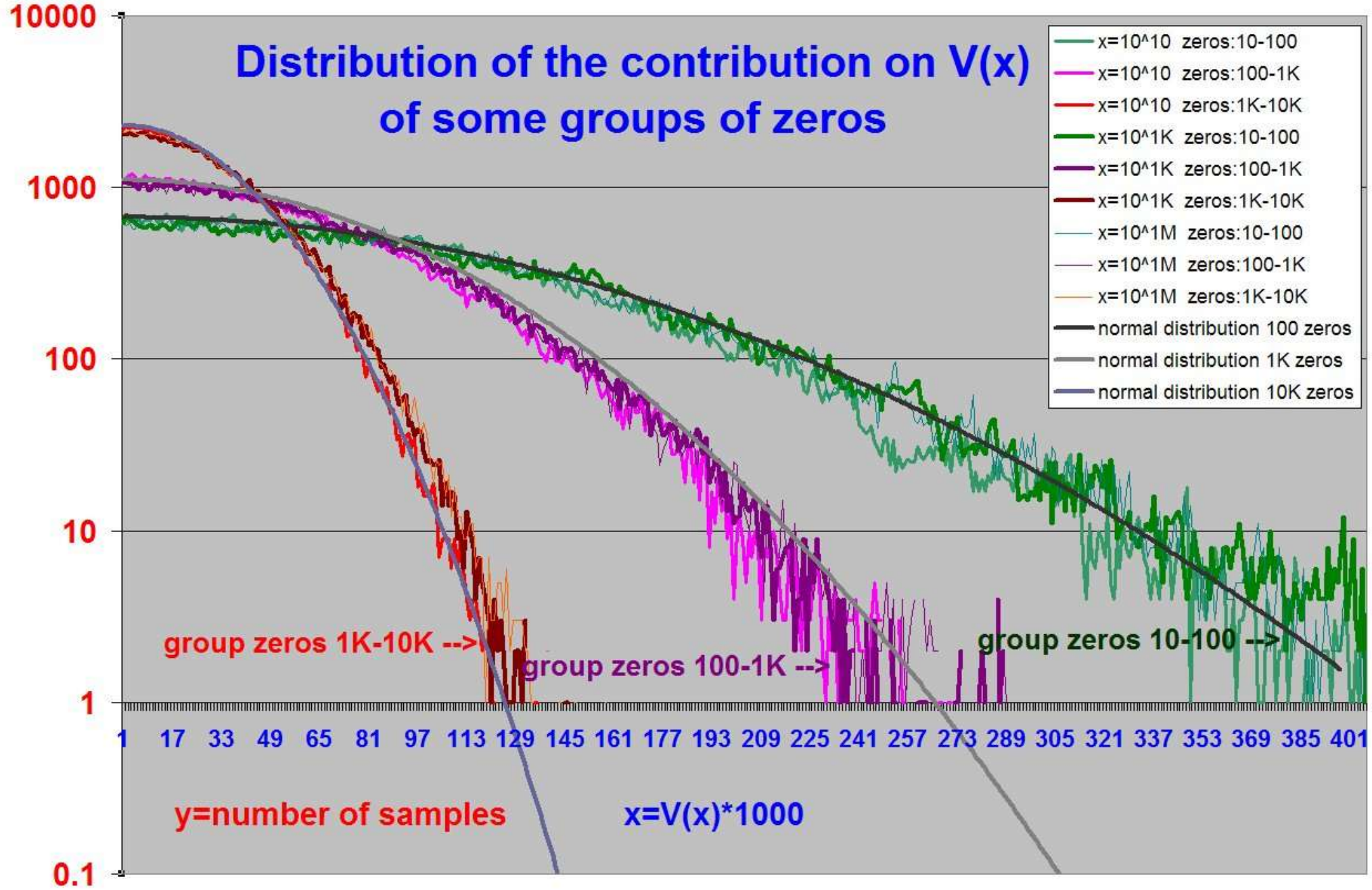
We can observe that the worst error on V(x) seems independent of the size of x at least up to 1e150
The error rate between 2 consecutive steps increases a little with more zeros, mostly because as you will see in the further slides ,when we use more zeros we have more important fluctuations then better resolution

Same with 100K samples with a wider range of x



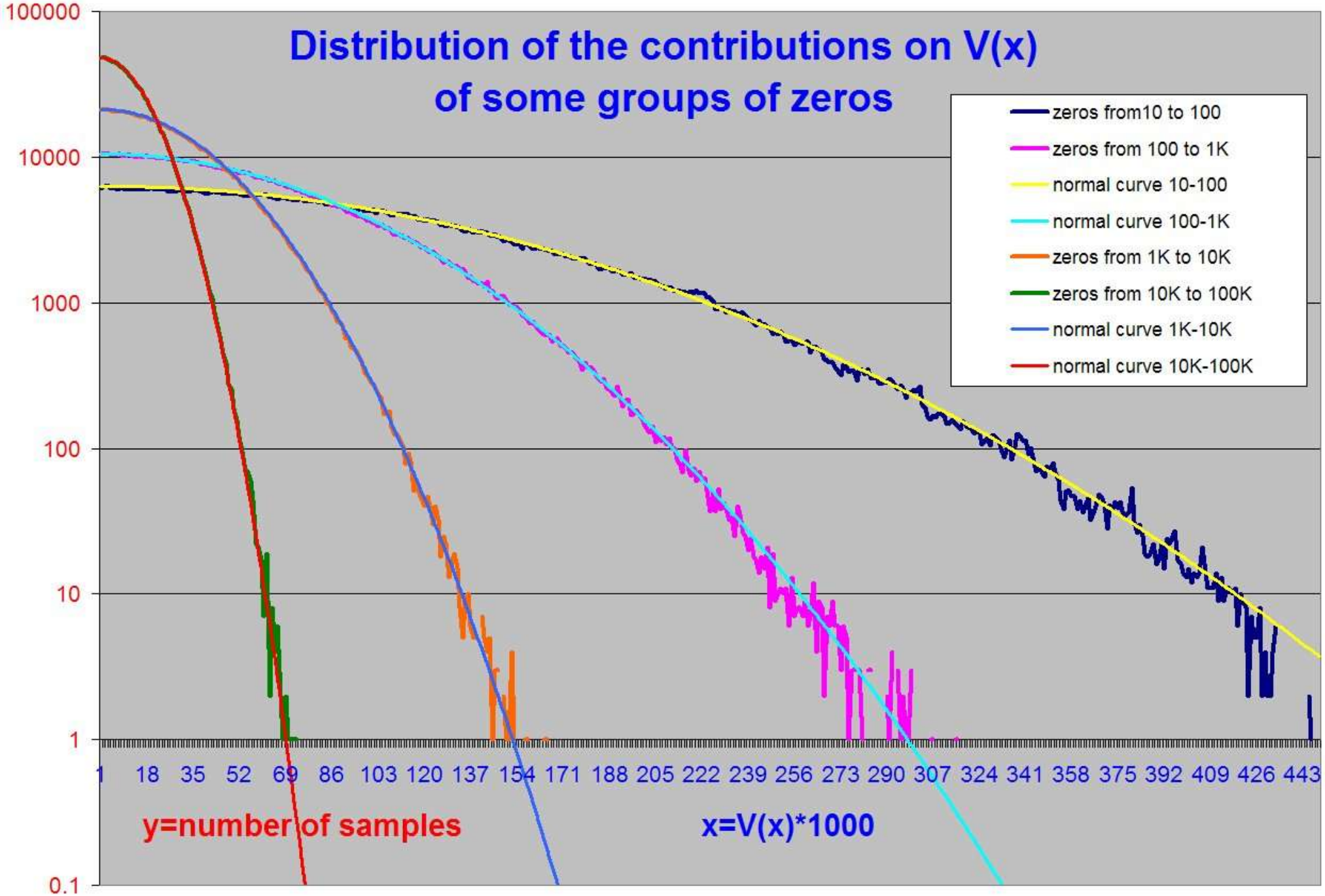
We can see that the accumulated error for 10K samples and 10K zeros is 0.3+0.07 = ~0.4 of 40%
For 100K samples and 100K zeros the maximum cumulated error is 0.05 + 0.03 = ~0.1 or 10%

Another way to look at the precision is to look at the contribution on V(x) of some groups of zeros



We can observe that the contribution of larger groups of higher zeros diminishes very rapidly
We can observe also that the value of x has only a very small effect on the distribution
The distributions seem perfectly normal then we can expect extremely rare large values

Another view on the distribution for x around 10^100 with a larger set of data



This slide is very important; it gives a precise idea of the error if we limit the number of zeros.
It also gives a precise idea on the probability to find an error of a specific size

The cost of computing exactly in simple precision "64 bits" one V(x) value on a good processor of year 2003 is:

with 1K zeros : 1ms
with 10K zeros : 10ms
with 100K zeros : 100ms
with 1M zeros : 1s
with 10M zeros : 10s
with 100M zeros : 100s
with 1G zeros : 2Ks : not stored in memory
with 10G zeros : 20Ks : not stored in memory
with 100G zeros : 3Ms : not stored in file

Precision of V(x) with N zeros

with 100 zeros : 40% or 0.4
with 1K zeros : 16% or 0.16
with 10K zeros : 6% or 0.06
with 100K zeros : 2.5% or 0.025
with 1M zeros : 1% or 0.01
with 10M zeros : 0.4% or 0.004
with 100M zeros : 0.16% or 0.0016
with 1G zeros : 0.07% or 0.0007
with 10G zeros : 0.03% or 0.0003
with 100G zeros : 0.01% or 0.0001

Remember this is the worst observed error and this worst error is pretty rare and observed only when we have a huge number of computations
The worst theoretical error is much greater, but will occur only if a large number of the millions of sinusoids would coincide in one place. Otherwise they have a natural tendency to cancel each other. That's why the worst observed error, is orders of magnitude smaller than the worst theoretical maximum. Then without any special property of the distribution of the zeros, we can consider the values as random. And as it seems to be the case, the errors distribution follows perfectly the normal distribution

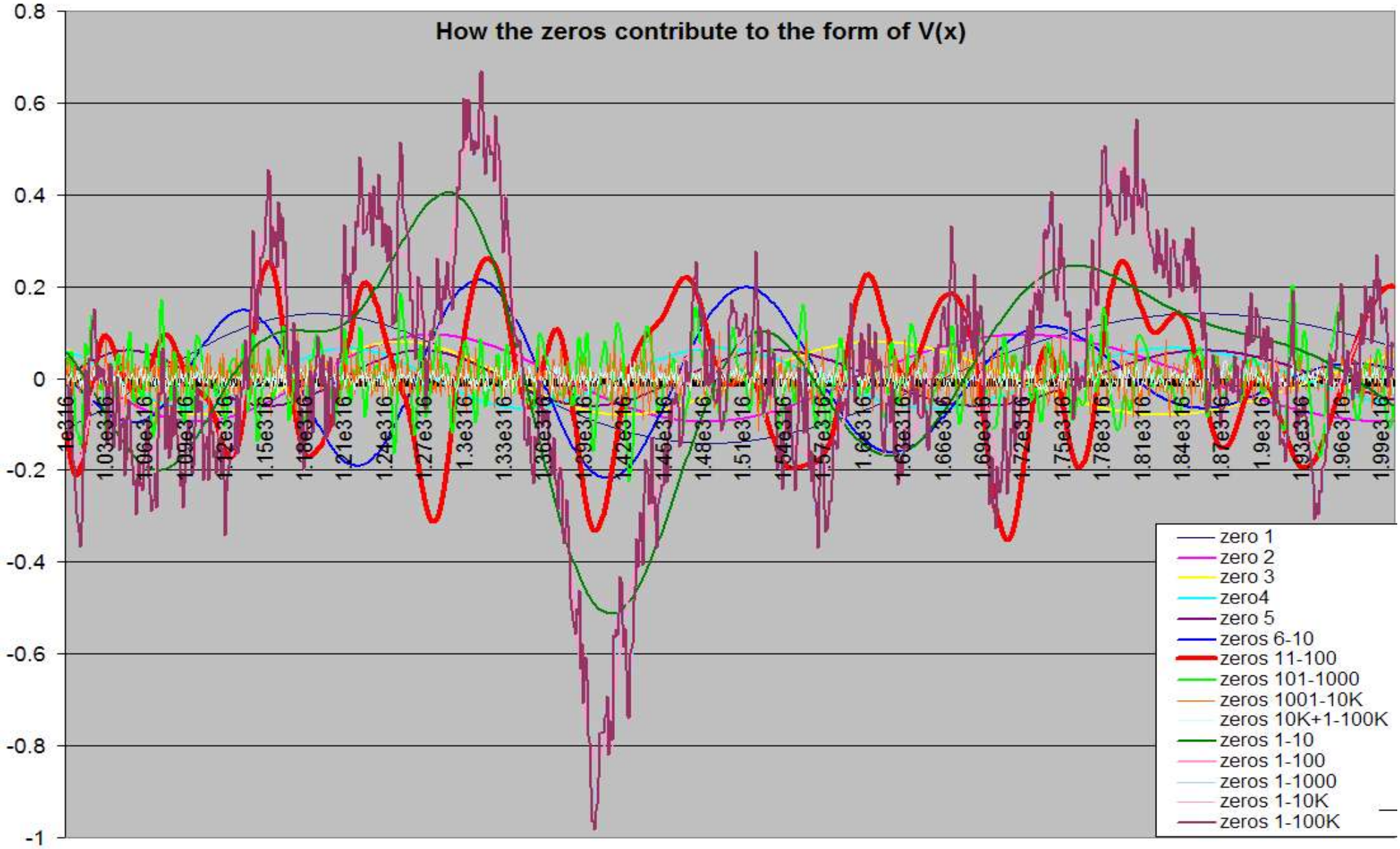
What is the greatest theoretical value that the zeros would contribute in V(x) ?

Measured at x=10^10M	**biggest amplitude of 1 sinusoid**
10 zeros : 0.67299	**0.141407**
100 zeros : 2.06216	**0.0089**
1K zeros : 4.64193	**0.00141**
10K zeros : 8.58697	**0.000202**
100K zeros : 13.98401	
1M zeros : 20.88519	

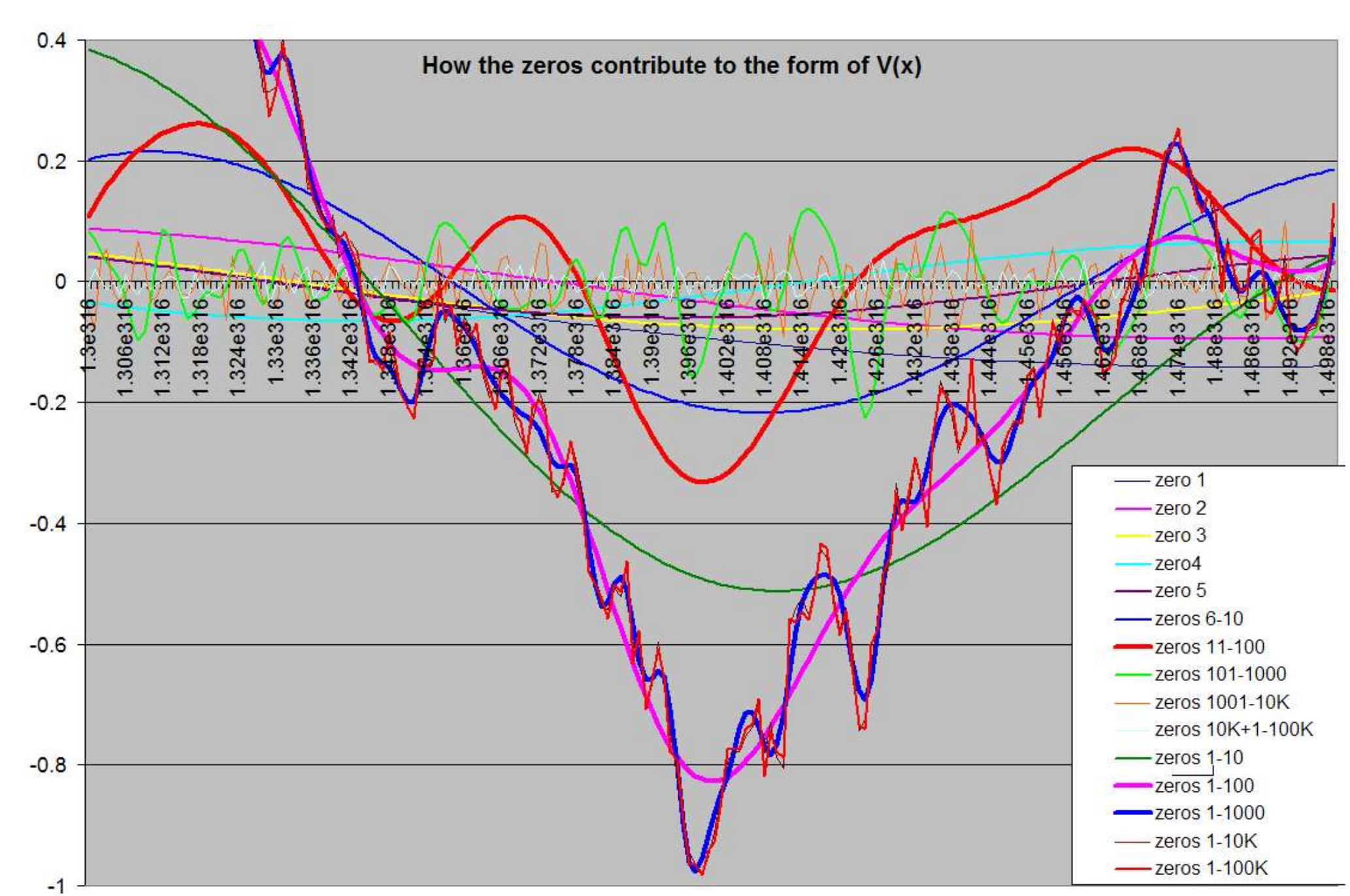
We can observe, that the worst observed error caused by not taking into account the zeros beyond a specific limit, is order of magnitude smaller than the maximal potential sum

This is because we use millions of sinusoids of different frequencies and similar slowly decreasing amplitudes.
The amplitude of one sinusoid is 2.0/(imaginary part of a zero)
The sinusoids have a natural tendency to cancel each others when we sum a huge number of them
Their sum tends to be very close to 0 in most places. The distribution of the values is also normal, and it is very rare and more and more difficult to find some absolute values for V(x) much larger than 1
The greatest values found today are just above 1.5
I predict we will never find a value greater than 2, please prove I am wrong ☺

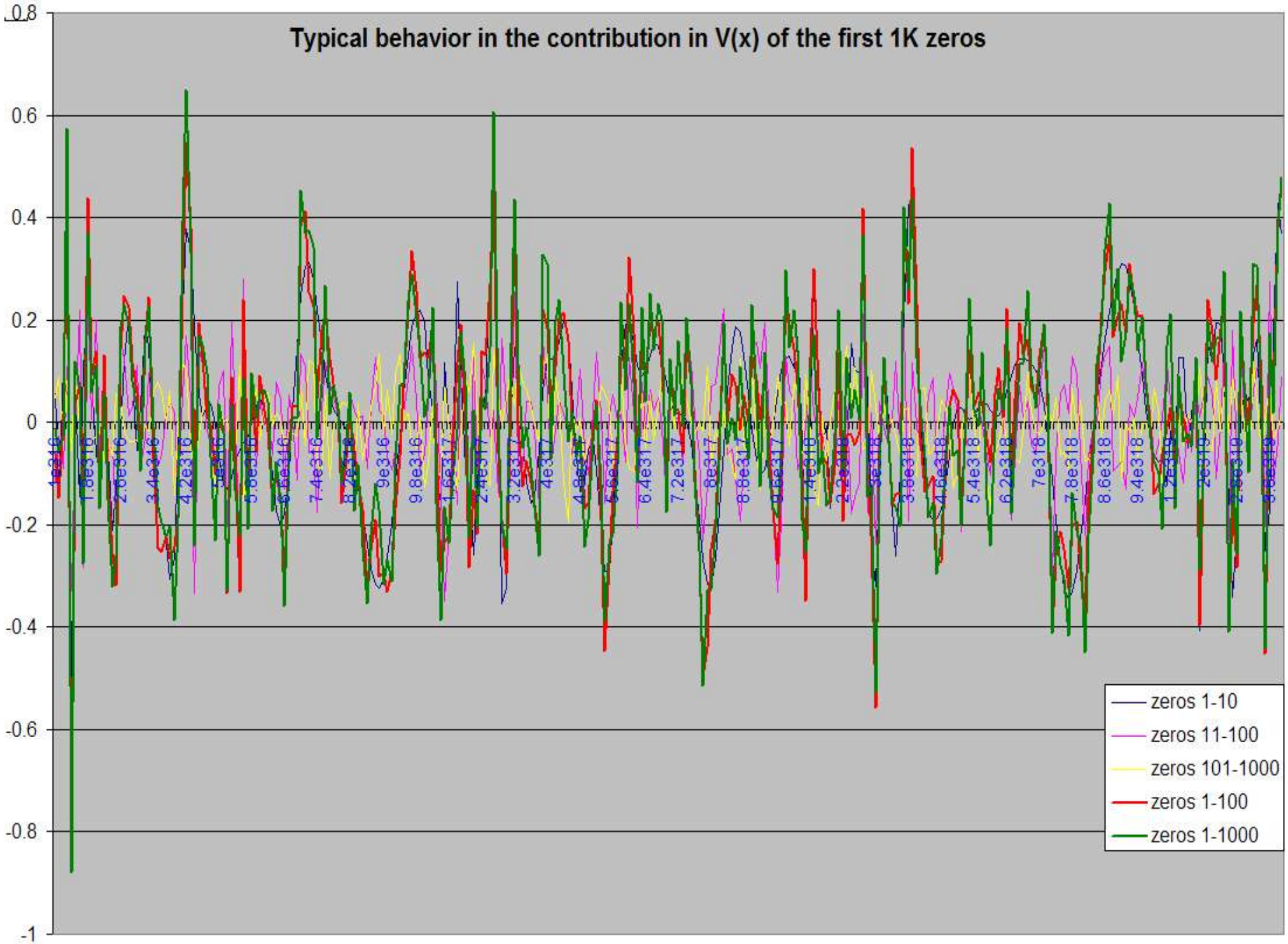
For example the zeros between 100K and 1M could contribute up to 20.88519-13.98401=6.90118 if they where all aligned for a specific value of x. This rare event, for sure occurs in many places, but those values are highly probably definitively out of humanity reach, since you have to align quite precisely 900000 sinusoids of different frequencies. In practice the worst observed error is around 0.01 and only for very rare values of x
In practice it is impossible to predict that there is not such a value with a small x < 1e316 where there is a perfect alignment of a huge number of sinusoids that would add a huge local contribution to V(x) and then having a crossover, but in practice the probability of such event for x is close to 0. To contribute for 1 we need more and more perfectly aligned sinusoids since the amplitude of the sinusoids for each zero tends to diminish rapidly. Then you see, the probability is very small, and except if there is something very special with the distribution of the values of the zeros, we should never find such an extraordinary place



We can observe that the zeros 1-10 tends to have a greater contribution than the zeros 11 thru 100
That's the same thing for the zeros 1-100 vs. the zeros 101-1000 and so on.
In theory it's the opposite; the maximal contribution is higher for higher groups of zeros, this will occurs in some very rare situations. In practice a higher group of zeros 10 times bigger tends to offer 2 times smaller contribution to the sum



With a zoom, we can observe that the first 1000 zeros are the biggest contribution
In fact, to pass the critical line y=-1, we will need 99000 additional zeros
We have this rare event, because the sinusoids 1-10,11-100,101-1000 have their negative peaks at almost the same value of x. Then adding each others to approach the critical value of -1



On this larger set of values for x, we can have a good idea on how V(x) is constructed

But this gives no precise idea on the probability when the V(x) reaches a typical value

Remember 1K zeros could potentially reach the value +-4.64193 if they synchronize all in some place
That is even harder to tossing a coin and having a suite of 1000 consecutive faces ☺☺☺☺……☺☺☺☺

To give a better idea of how rare is that event, let's imagine that all men that ever leaved on earth would spend all their life tossing day and night at a rate of 1 toss per second
The probability that this event occurred is 1/10^279 or 0.00000……000001 there are 278 zeros ☺

Let's play another game:
Imagine that all particles in our universe are playing machines, maybe they are
They play a game of taking 2 specific states let's call them 1 and 0
Now they randomly change their state at the maximum rate they can
Let's imagine a quantum time of 1e43 states per second then our particles can toss at a frantic rate of 1e43 times per second, since the birth of universe this is 4.7e60 tentatives per particle
The winning particle "Master of this universe" is the particle that first sees 1000 consecutive "0" or 1000 consecutive "1". I hope there will be no tie state ☺

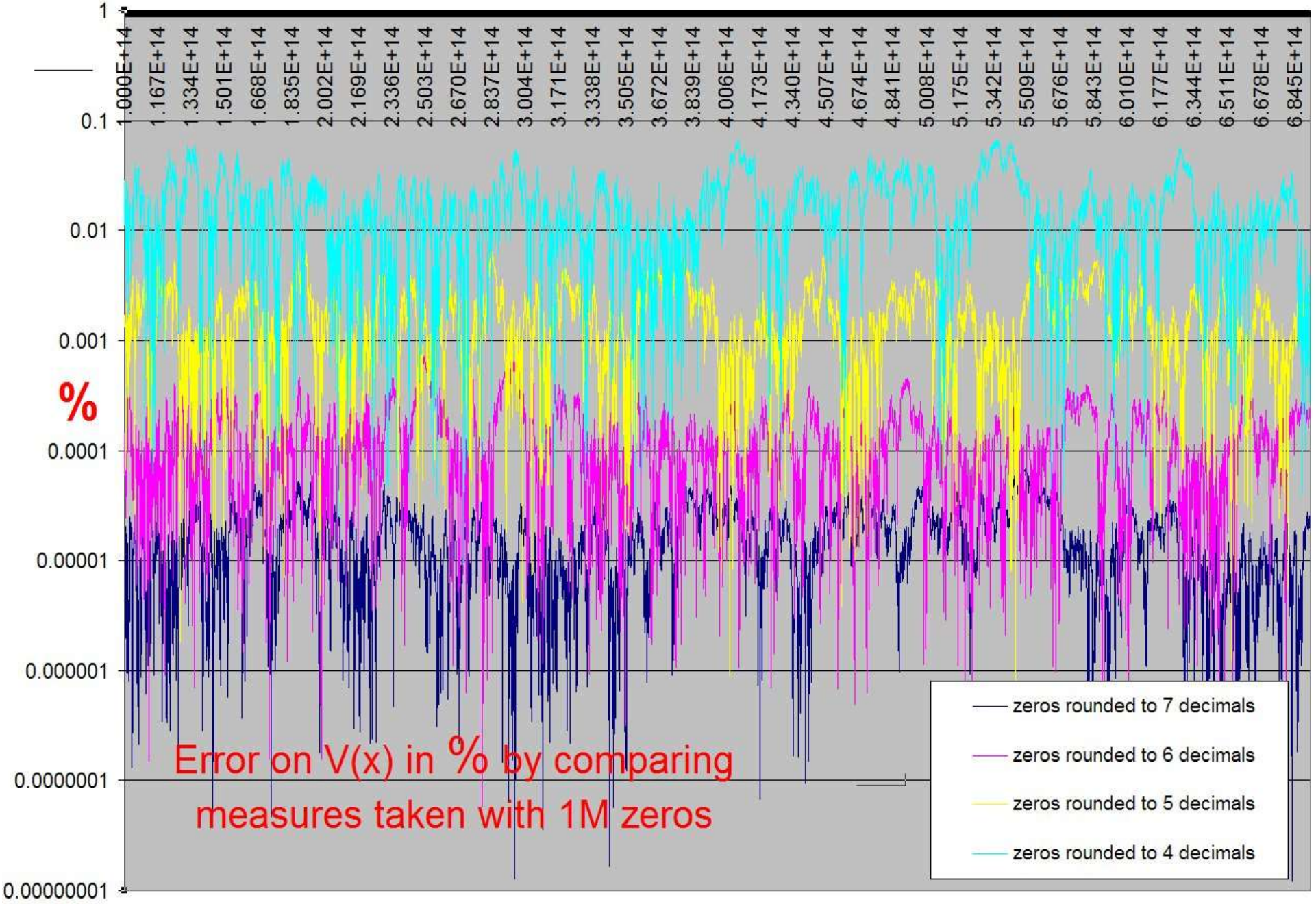
Then the probability that a single particle found 1000 consecutive 0 or 1 since the big bang is 1/10^152 or 0.000….00001. This time we have only 151 zeros ☺

If you doubt about my estimation, refer to the Margolus-Levitin theorem that estimates the compute capacity of our universe as 10^107 operations per second, quite low compute capacity. We can then imagine lot of questions without potential response if the optimal algorithm would require more than 10^200 operations

That's why in practice we observe very rarely |V(x)| > 0.6 if we take only 1K zeros into account
Meanwhile with only 27 zeros we could potentially observe a crossover: V(x) < -1

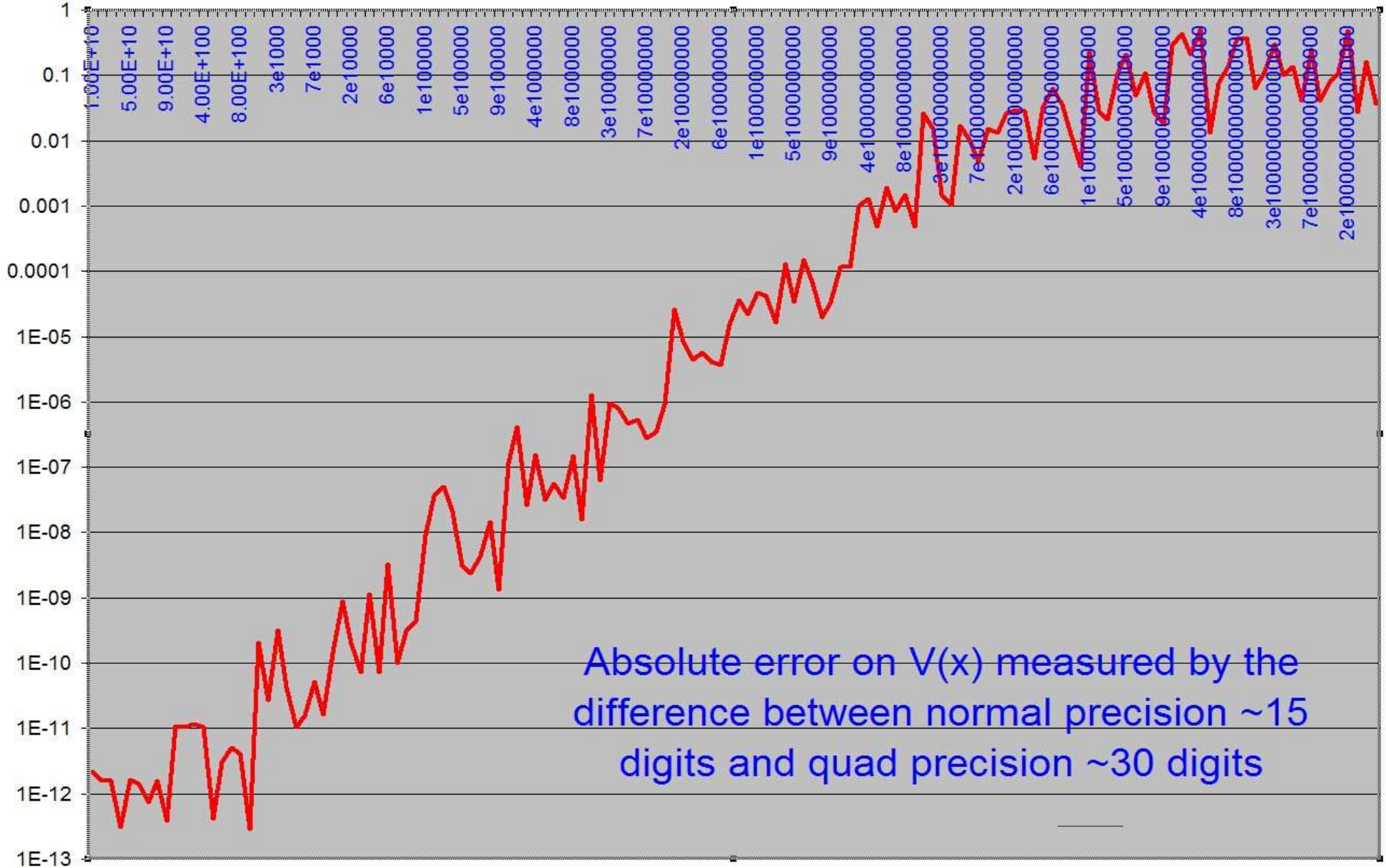
Naturally the alignment of 900 zeros would be already a very large anomaly, but that much more frequent event is also terribly rare for "humans"
We will study latter the distribution of the values of V(x), to be capable of evaluating with good precision, the probability of those events

What is the precision of V(x) measured with various precision of the zeros?
I deliberately degrade the zeros by rounding them to N decimals to measure the impact



With 7 digits of precision on the zeros we have at least 6 digits of precision on V(x)

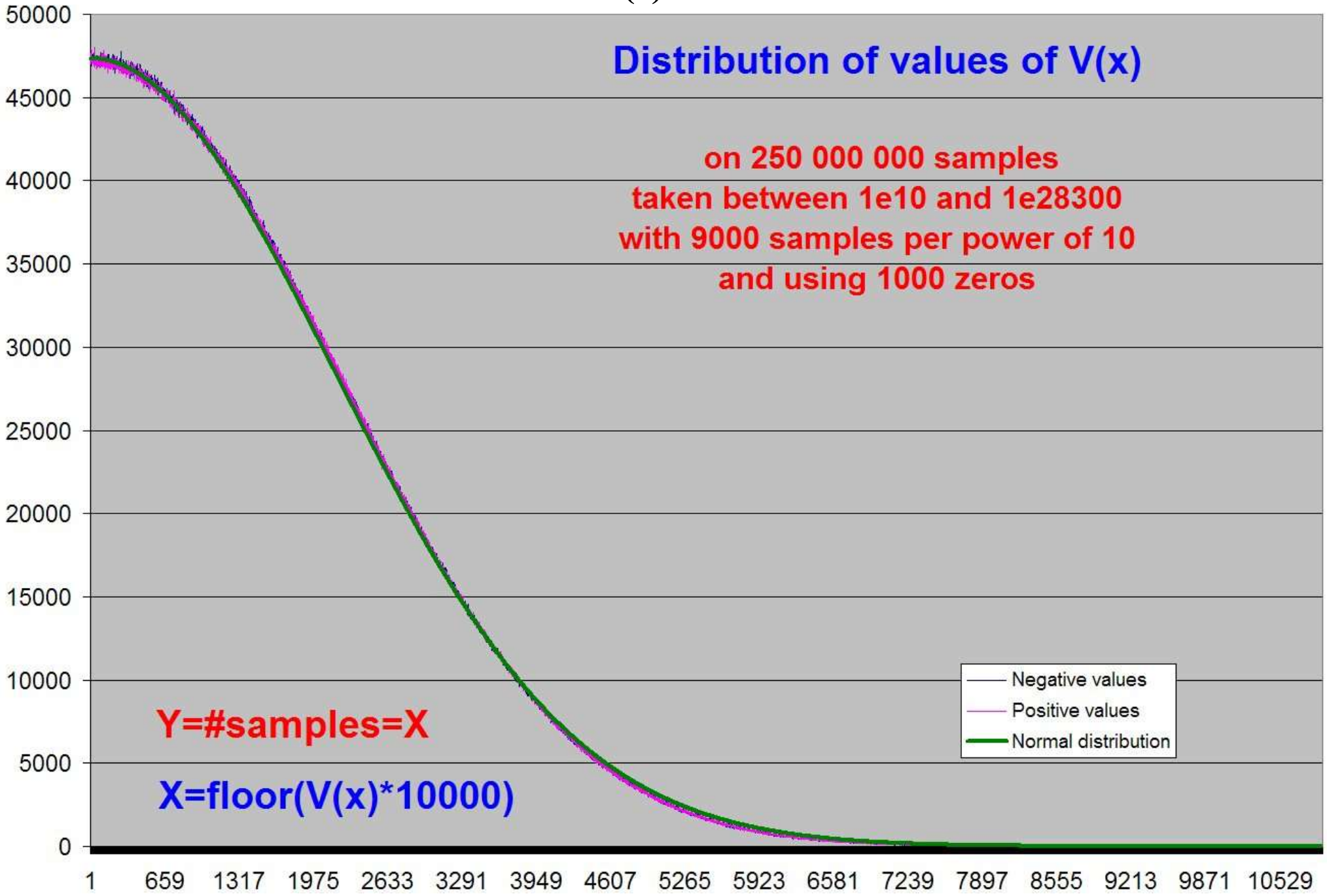
What is the precision of V(x) according the size of x ?



Since we need at least 3 or 4 digits of precision for V(x), whose absolute values are mostly below 0.1 it becomes risky to use the normal precision "double" with x > 1e100000000

But in the range < 1e1000, we have at least 9 digits of precision, then much more than needed, since the precision on the zeros is only 7 digits

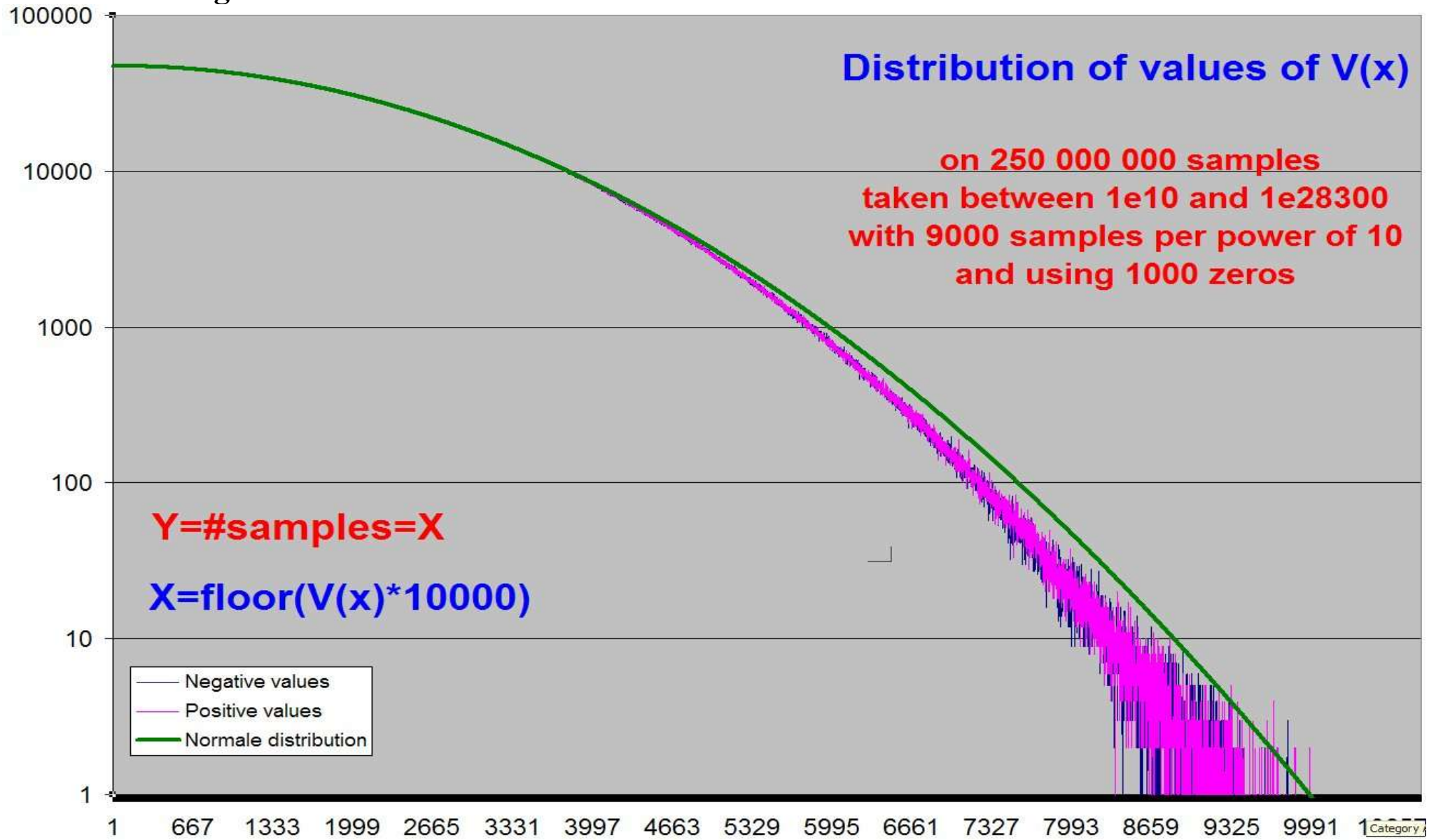
What is the distribution of the values of V(x)?



The curve is very close to the normal distribution.

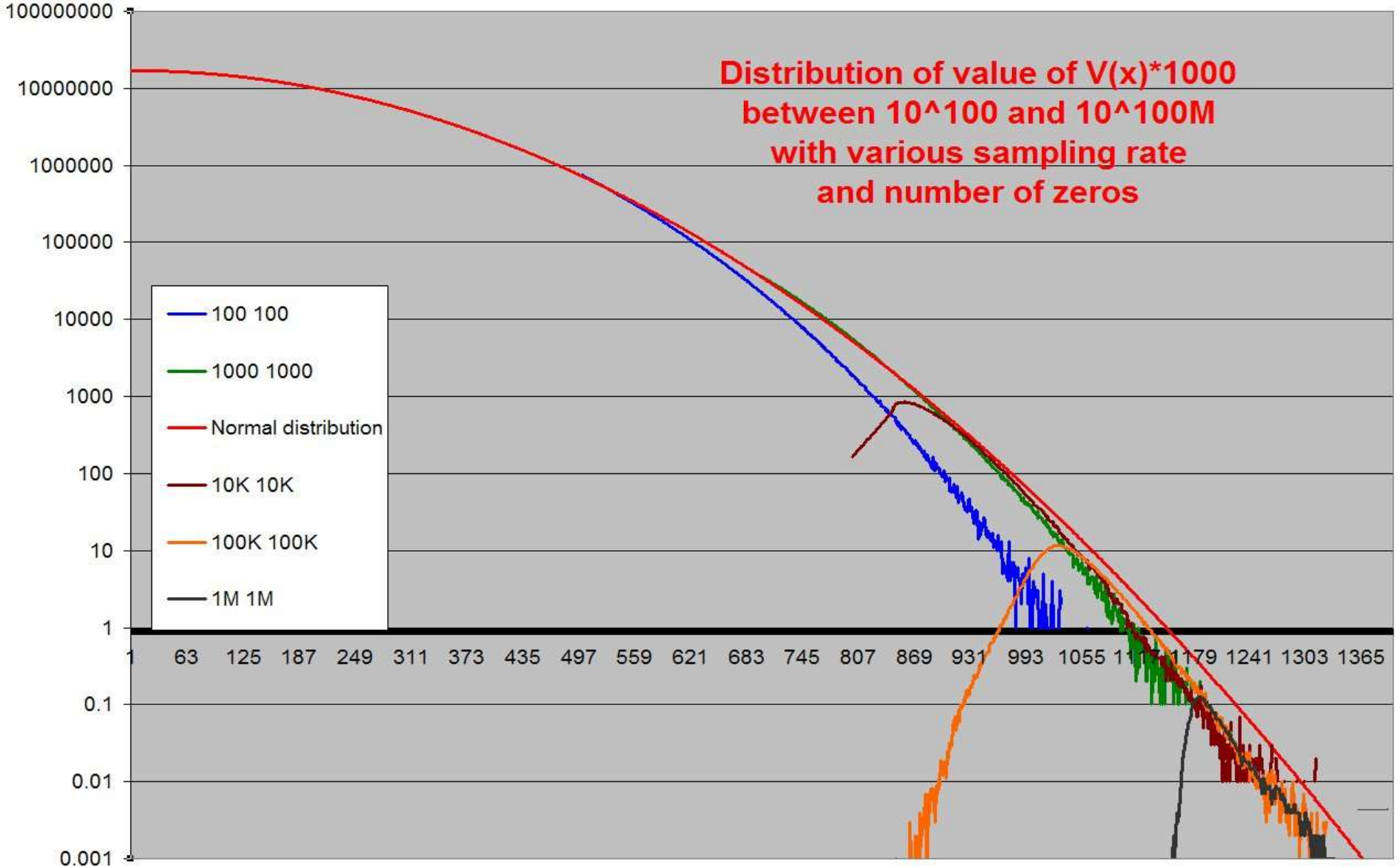
If we can confirm that the distribution is normal, this will give us a very precise formula for the error term

Same with a logarithmic scale



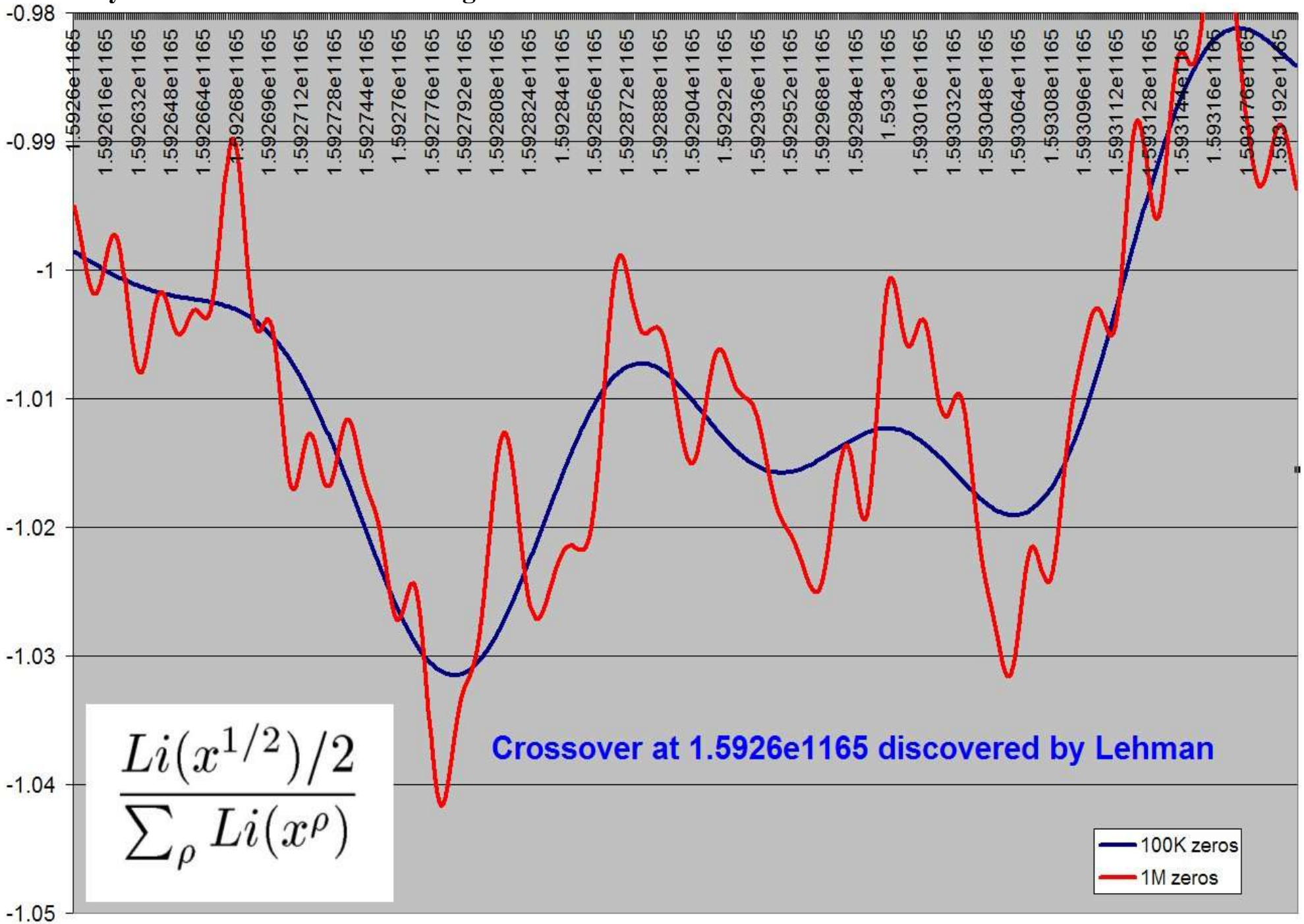
We can observe that the normal distribution diverges from the real curve, but if we would have taken more zeros we would have the curve much closer to the green line
This curve permits to determine the probability that V(x) could reach a specific value
We better understand now, why it is so hard to find a large value for |V(x)|
The green line could potentially reach x=46419.3 with 1000 zeros, but we need to compute a terribly large number of samples to have the chance to find one of the infinite occurrences of that event

A larger experiment to verify if the curve stays normal up to 10^100M

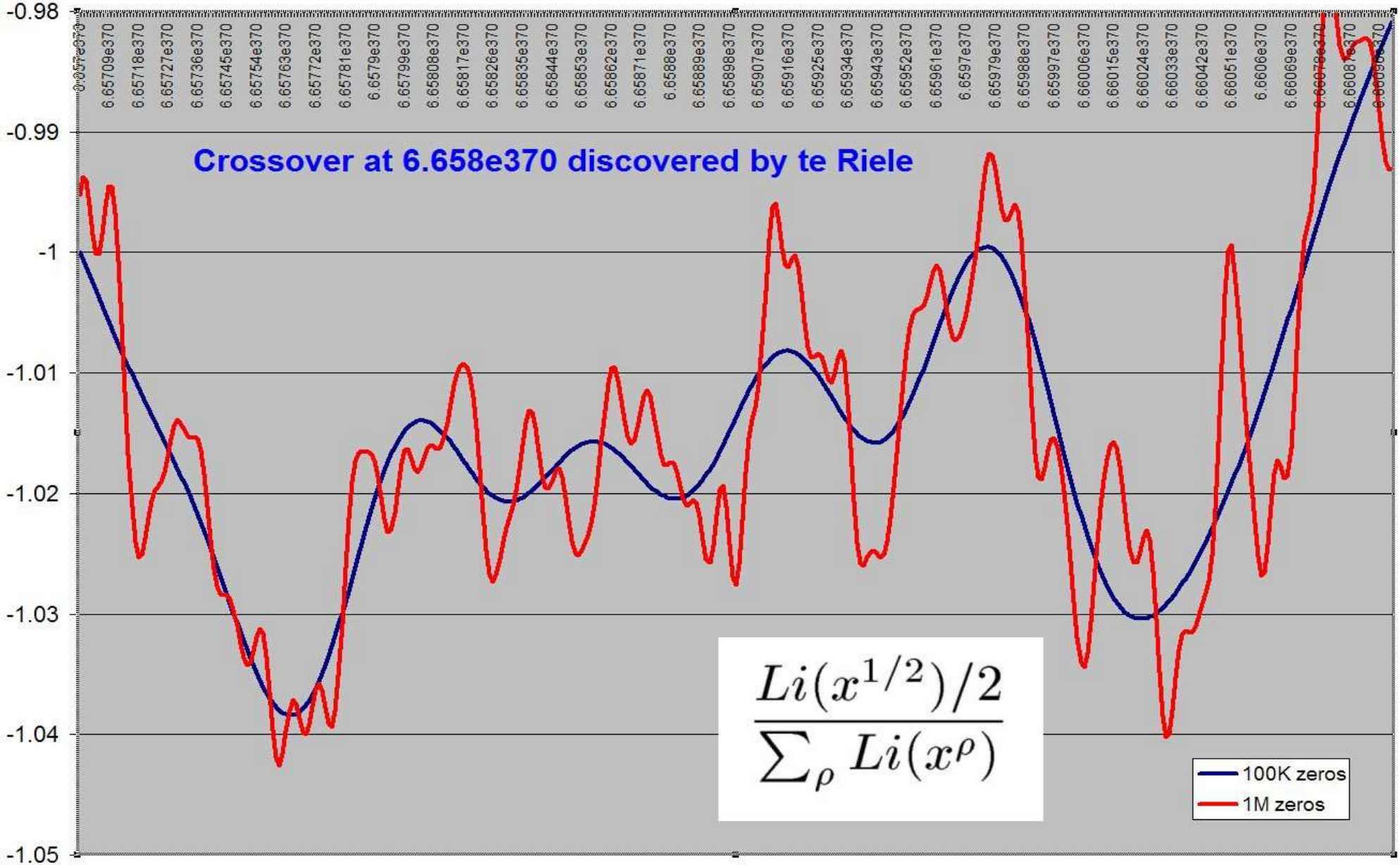


Since I need to measure with lot of zeros it is impossible to sample at a very high rate all the values up to 10^100M, then I start with 100 samples and 100 zeros per exponent, and I recursively increase by 10 the number of zeros and the sampling rate when I reach a specific interesting threshold
This explains the form of the curves like "10K 10K", the left part is an indication that I missed many samples, but the right part of the curve is valid. The curves approach the red normal curve as I increase the resolution

Previously identified crossovers starting with Lehman

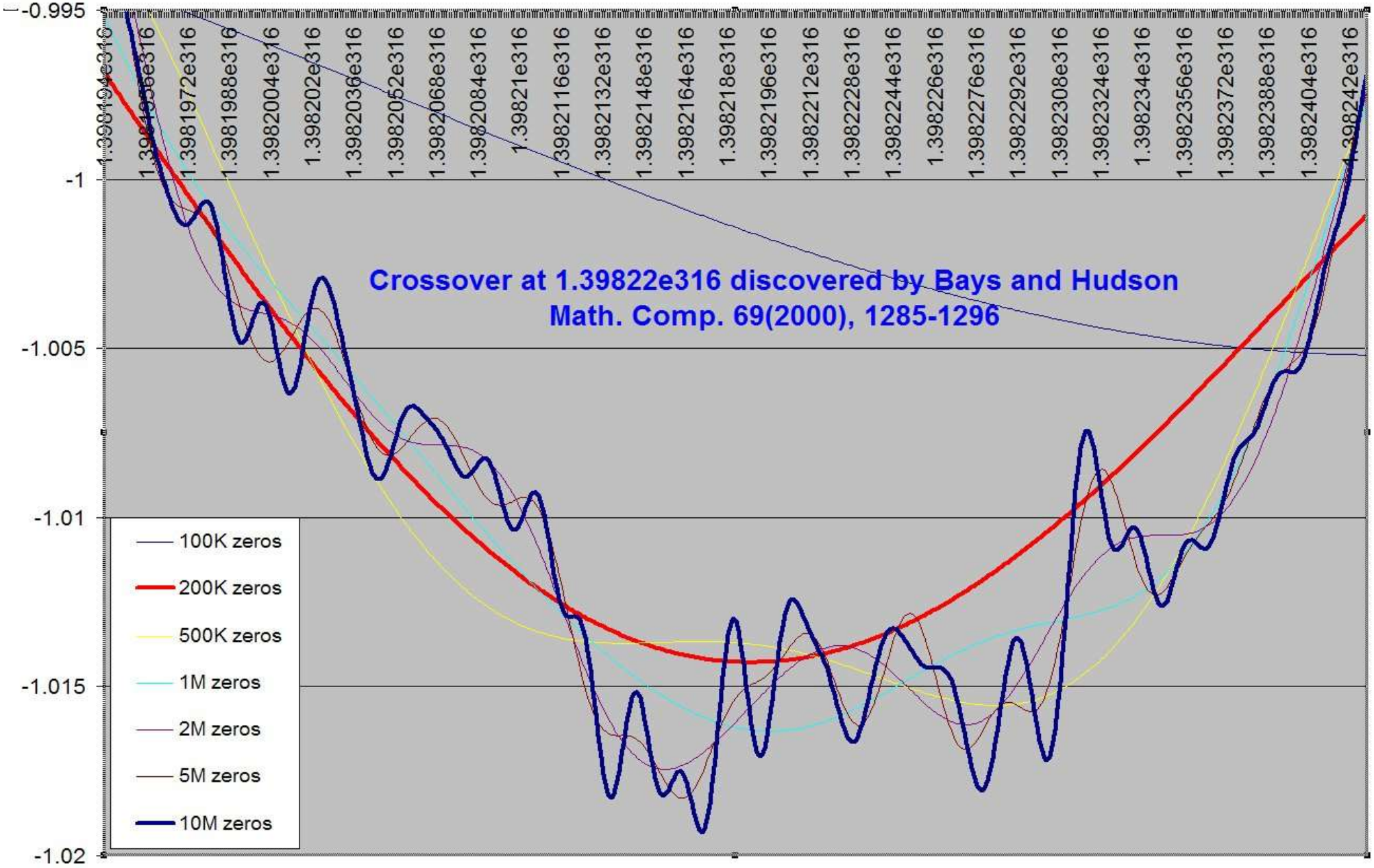


te Riele contribution



We can always observe some huge fluctuations when we increase the number of zeros
Those fluctuations are the principal cause of error when we limit the number of zeros
The fluctuations at higher frequency are also a guaranty, that if we use much more zeros, the crossover will still exists somewhere. The oscillations at higher frequency around the red curve will create some places where V(x) is overestimated but also some very interesting places where it is underestimated

The current best value for the first crossover, found in 2000 by Bays and Hudson but sometimes contested

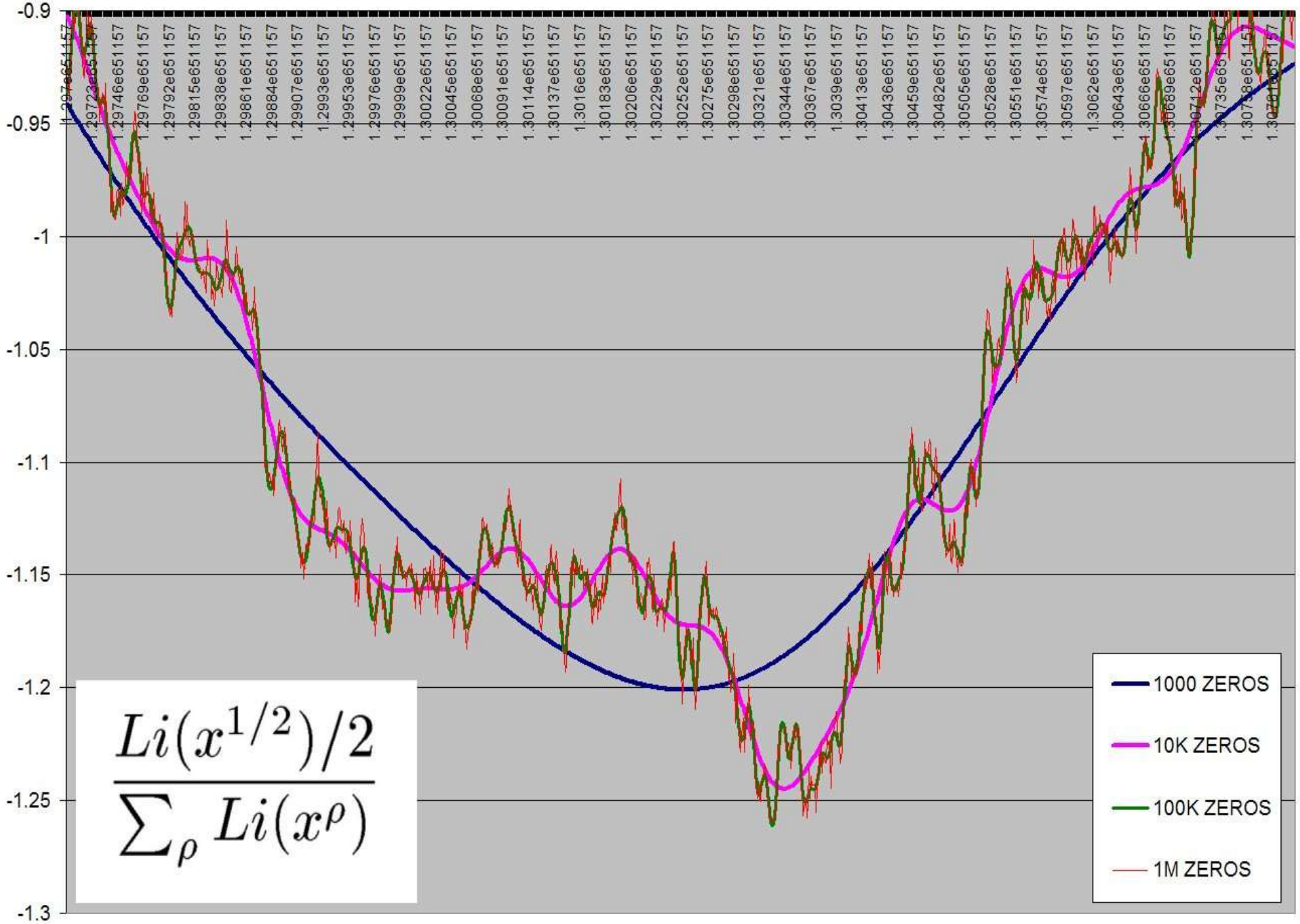


This was definitively the correct, best value for the first crossover in 2000

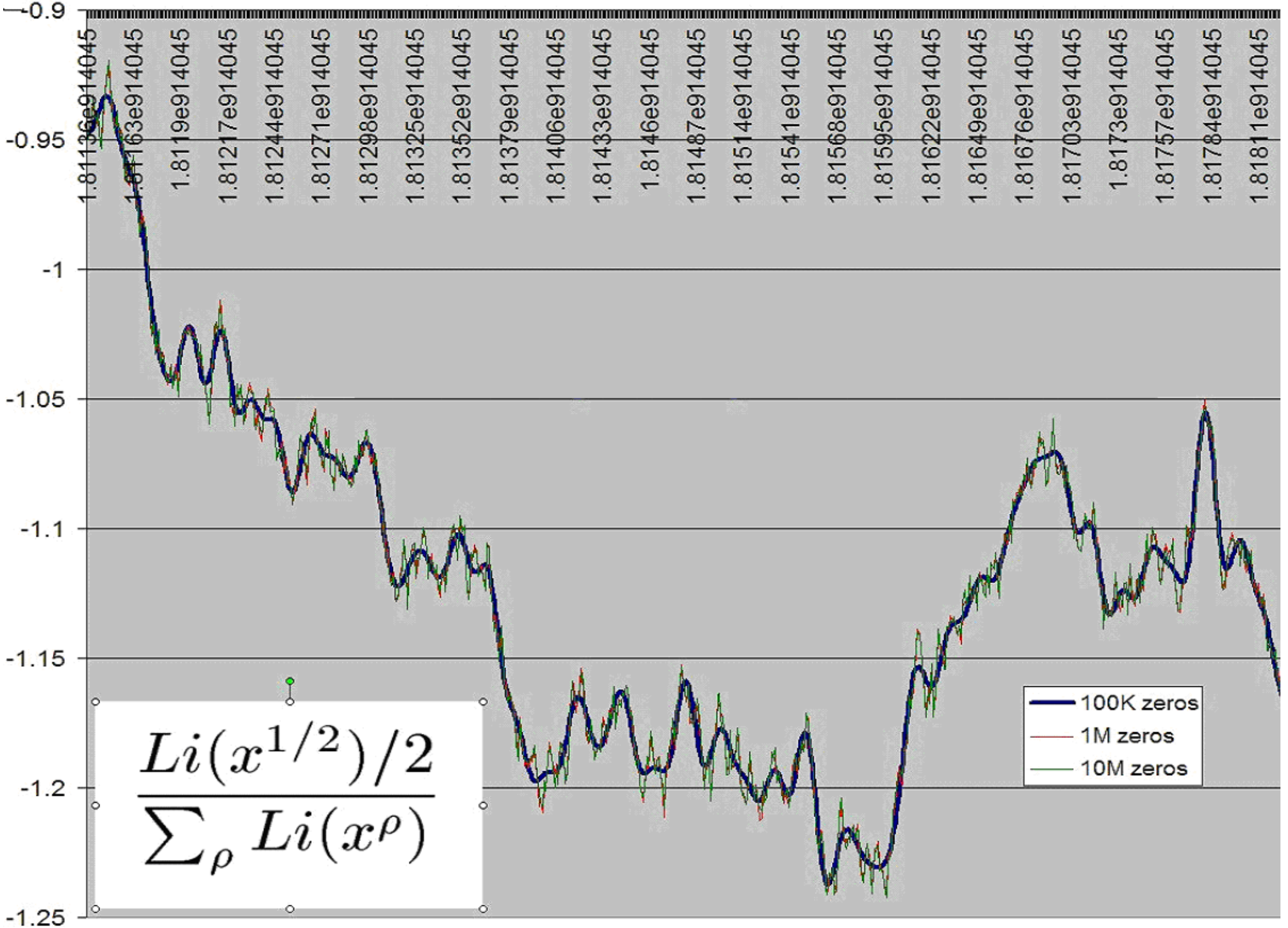
Observe how a curve with more zeros zigzag over a curve with less zeros, this is caused by the fact that higher group of zeros contribute by sinusoids of higher frequency and decreasing amplitude

Observe also that large errors occur frequently close to the peaks

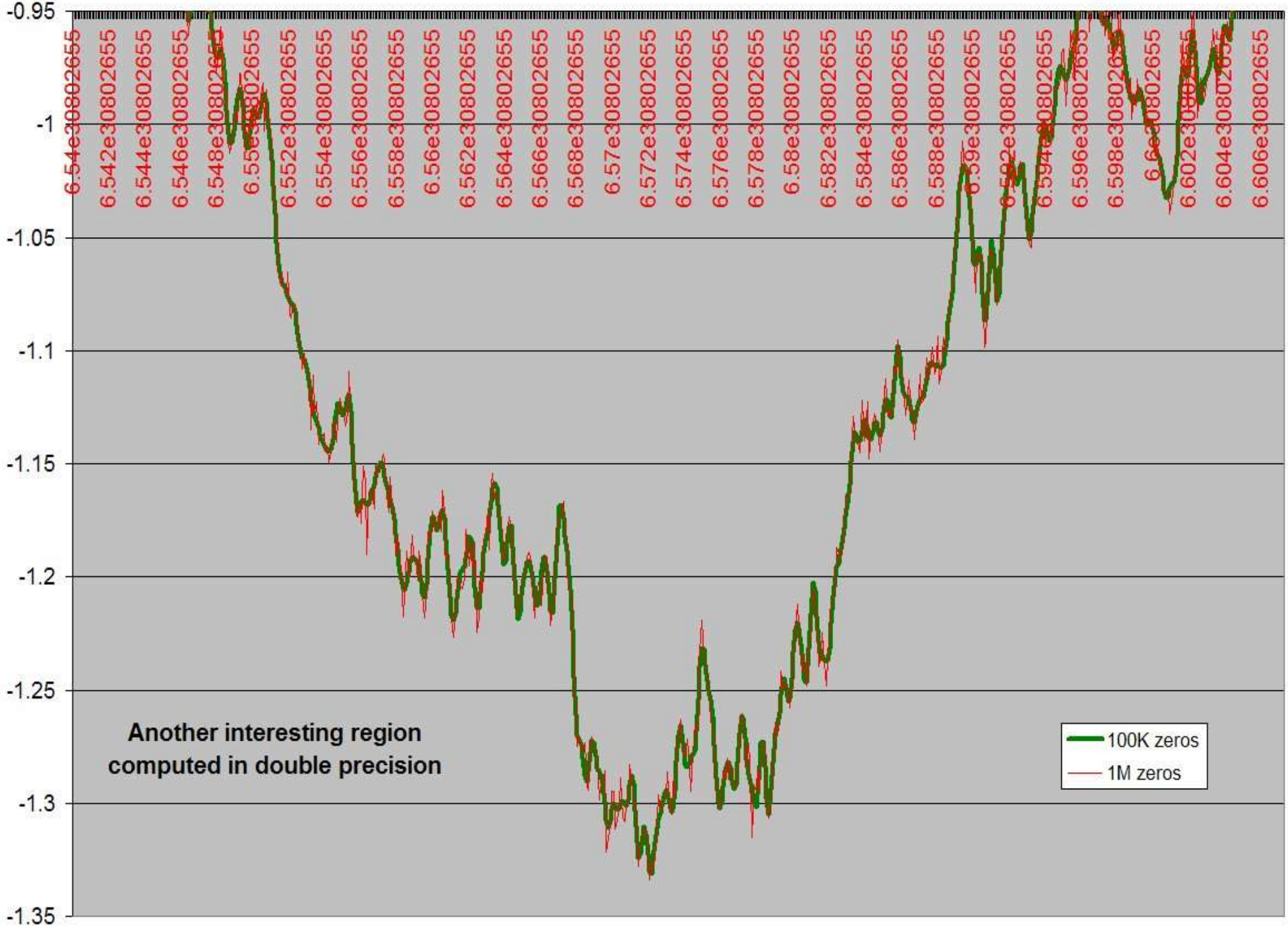
Some interesting large and profound crossovers
This one discovered by Bays and Hudson



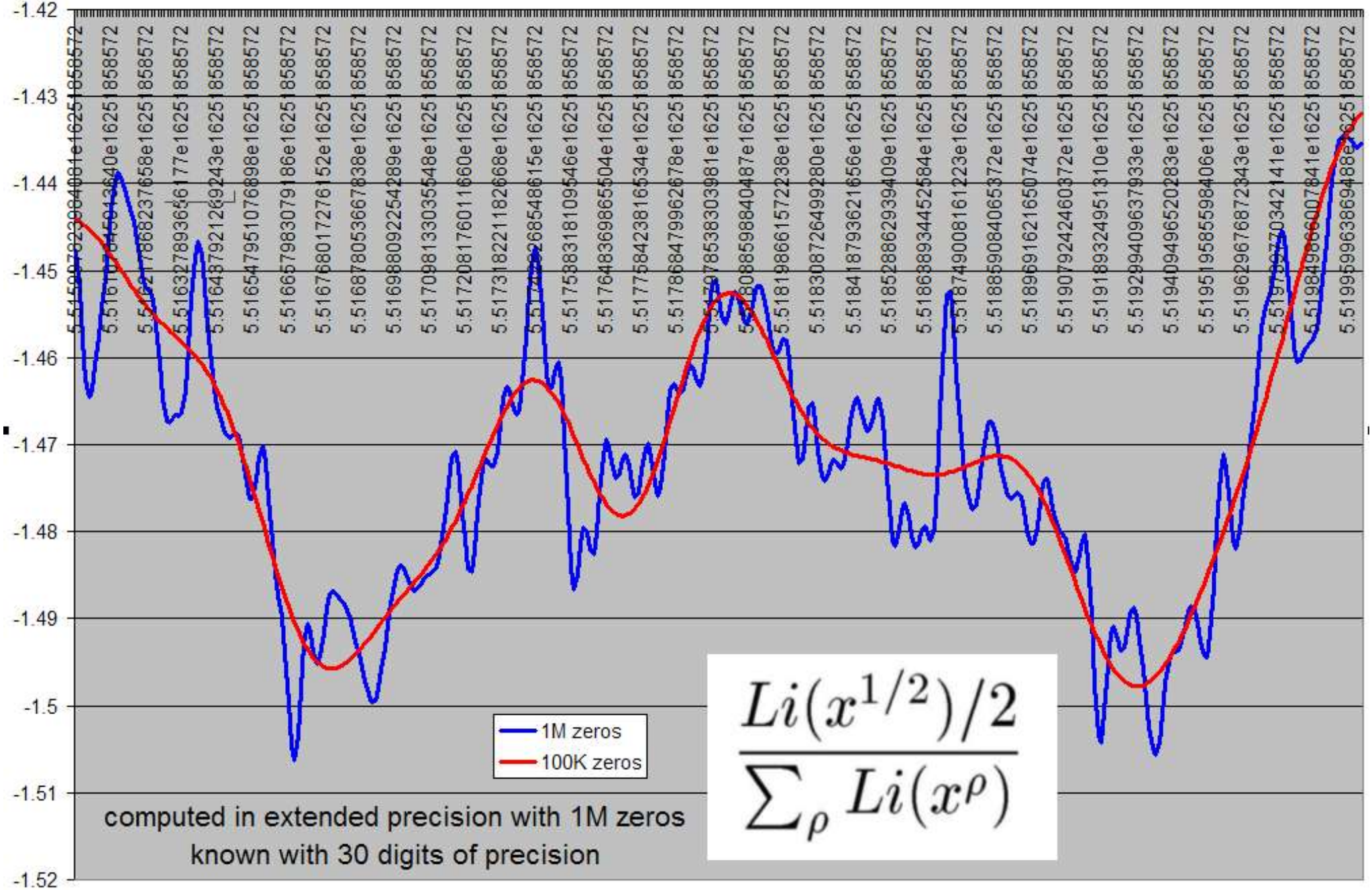
This one found by me



An even more profound crossover



One of the numerous profound crossovers I found with a new optimized code



This one can only be found when using extended precision with the computations and extended precision with the zeros. Also computed with 2 very different methods for validation of the algorithms
Try to beat this record ☺

Heuristic to find the first crossover with the very high confidence:

Let's imagine I want to compute all the samples from 10^10 to 10^320 with less than 1% of tolerance or .01 on the value of V(x) , I need to compute 1M samples per power of 10 and use 10M zeros
This would cost 310*1M*10s=3.1G seconds or ~100 years: OOPS

A better strategy is to accept a much larger tolerance like ~10%, then we need only 100K samples and 10K zeros. This will cost 310*100K*0.01s=310K seconds or ~90 hours: somewhat better

But since we have 10% margin of error, we need to take all results that have a value inferior to a specific limit like -0.8 and increase the sampling rate and the number of zeros around those regions to reduce the error

The distribution curve teaches us that only 0.005% of the samples for x have a negative value inferior to -0.8

Now computing those samples with our initial targeted precision of 1% cost:
310*1M*0.00005*10s = 155K seconds or ~40 hours
And now we have isolated all regions close to the crossover zone with 1% error tolerance in less than 5 days

By recursing this procedure, we can easily isolate all potential regions and measure them with very high precision and absolute confidence to have missed absolutely no interesting region

In fact this heuristic is so efficient, due to the form of the curve of distribution for the values of V(x), that we can take no risk by accepting at each step of the computation a significantly larger error margin, as large as possible and compatible with our compute power. This large tolerance corresponds to a very small probability to find an exception. In fact finding such an ultra rare exception could be considered as a record

Since I want to be absolutely certain to take no risk of missing a crossover, I will tolerate a much larger compute time and apply the following steps:

1: I sample between 10^10 to 10^350 with 100K samples per power of 10 and 10K zeros
I gathered 30 600 000 samples in 4 days

2: I know from the analysis that the worst error is below 0.1 or ~10%. Then I take a large margin by filtering the output file with all values below -0.6. This corresponds to 40% or 4 times the estimated maximal potential error, on the normal curve of distribution of error this correspond to a event that I could never find

From the 30M samples of the first iteration, I extract all the samples where V(x) < -0.6
I obtain only 59359 samples

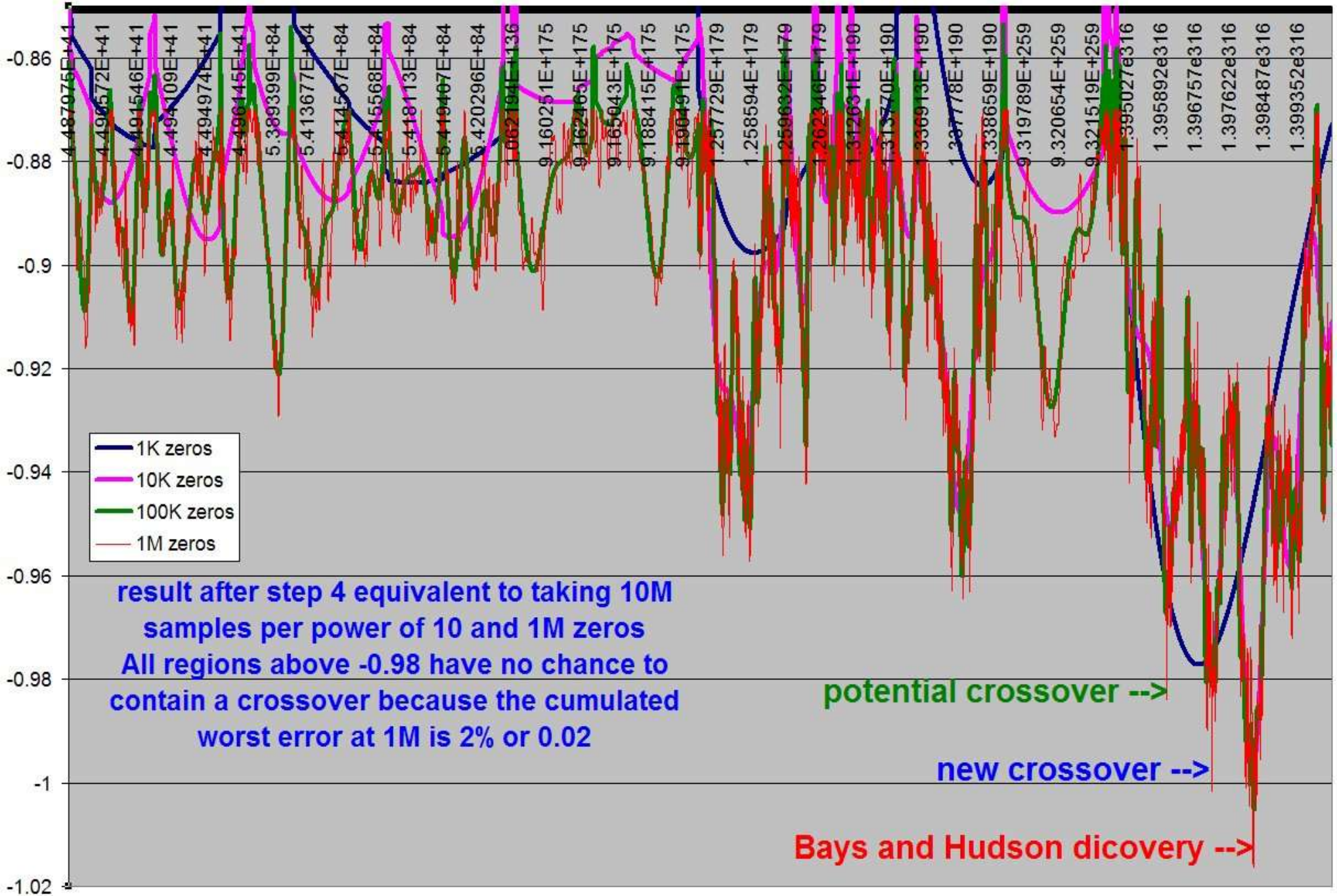
3: for each of the 59359 samples, I now take 5 samples before and 5 after with 10 times finer resolution, this corresponds to a 1M samples resolution. I also use 10 times more zeros or 100K zeros
The complete sampling with that resolution would have lasted 400 days, this filtered samples computation cost only 17 hours
I have now 593590 samples where I can now look what was the real worst error created at stage 1 of the computation by using only 10K. The worst error is 0.0667 or ~7% : below my initial estimation of 10%, you see you will never find a value with an error of 40% or 0.4

4: Now I filter all the samples that are below -0.8 or 20% error tolerance, this is a very large tolerance since I expect the error at 1M to be around 1%, but since the compute time is acceptable I can take an unrealistic tolerance. I obtain 15050 samples

5: I repeat the same procedure, increasing the sampling rate by 10 and the number of zeros by 10
The complete sampling would have lasted 100 years
Now the computation with 1M zeros cost only 42 hours, somewhat shorter ☺
The worst error at 100K is 0.025 or 2.5%, very close to the expectation
At 1M zeros the maximal observable error is ~2% = ~(1%+(1%/2.5)+(1%/2.5^2))+(1%/2.5^3)+…), as a consequence all regions that are above -0.98 have only a very small chance to contain a crossover, even if we take into account an infinite number of zeros

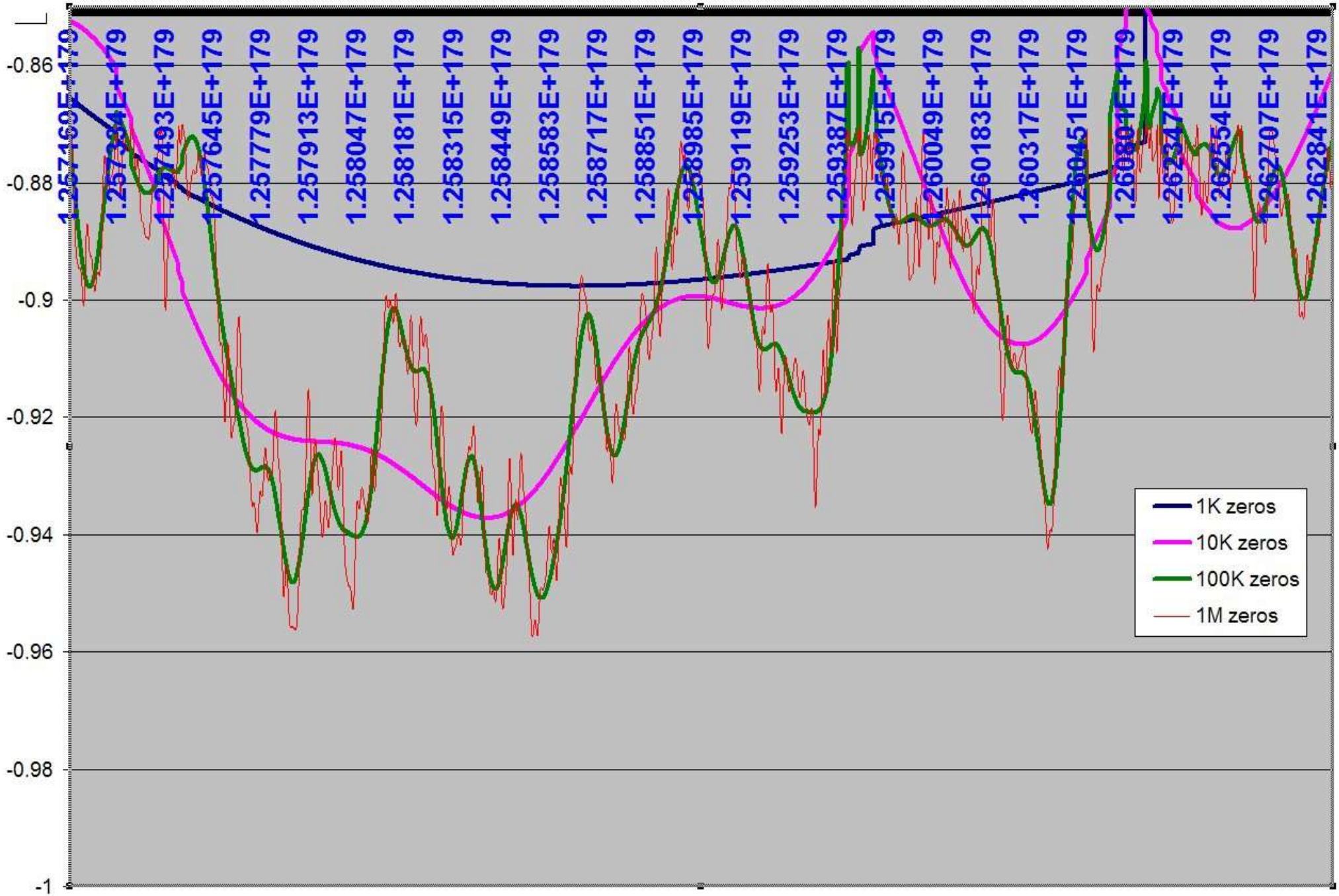
6: For increased security I will run another verification step:
I extract all samples from previous step that have a value below -0.9 and increase by 10 the sampling rate and multiple by 10 the number of zeros, this corresponds now to 100M samples per power of 10 and 10M zeros
I have now 11525 samples, then this will cost 320 hours for the last level of verification
I can now observe the worst error at 1M samples 1% or 0.01 as predicted

Results with 10M samples and 1M zeros



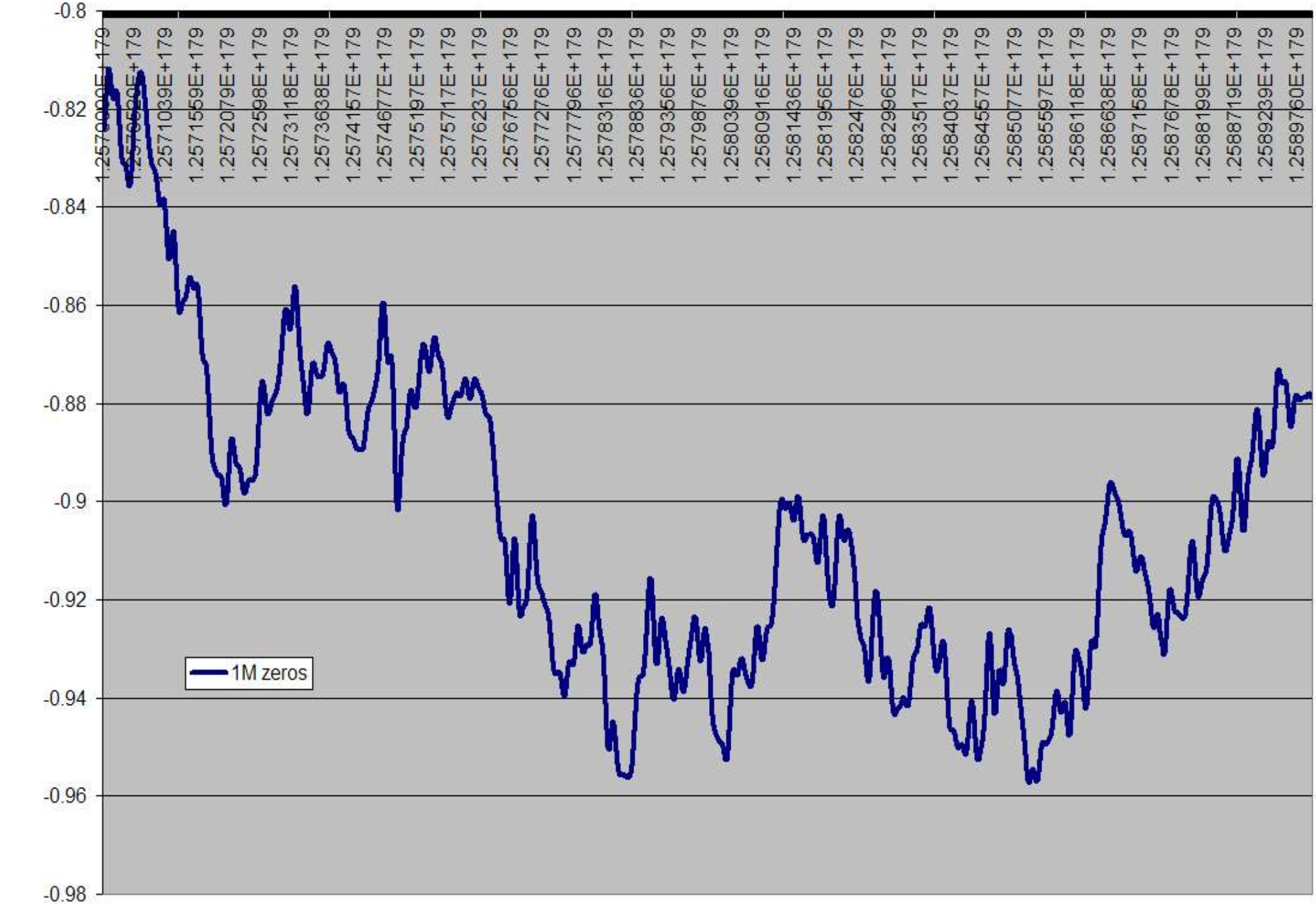
I will zoom the regions at 1e179, 1e190 and 1e316. Even if there is no risk to see a crossover in the regions 1e179 and 1e190 this need to be checked carefully with much more zeros

Zoom of region 1e179



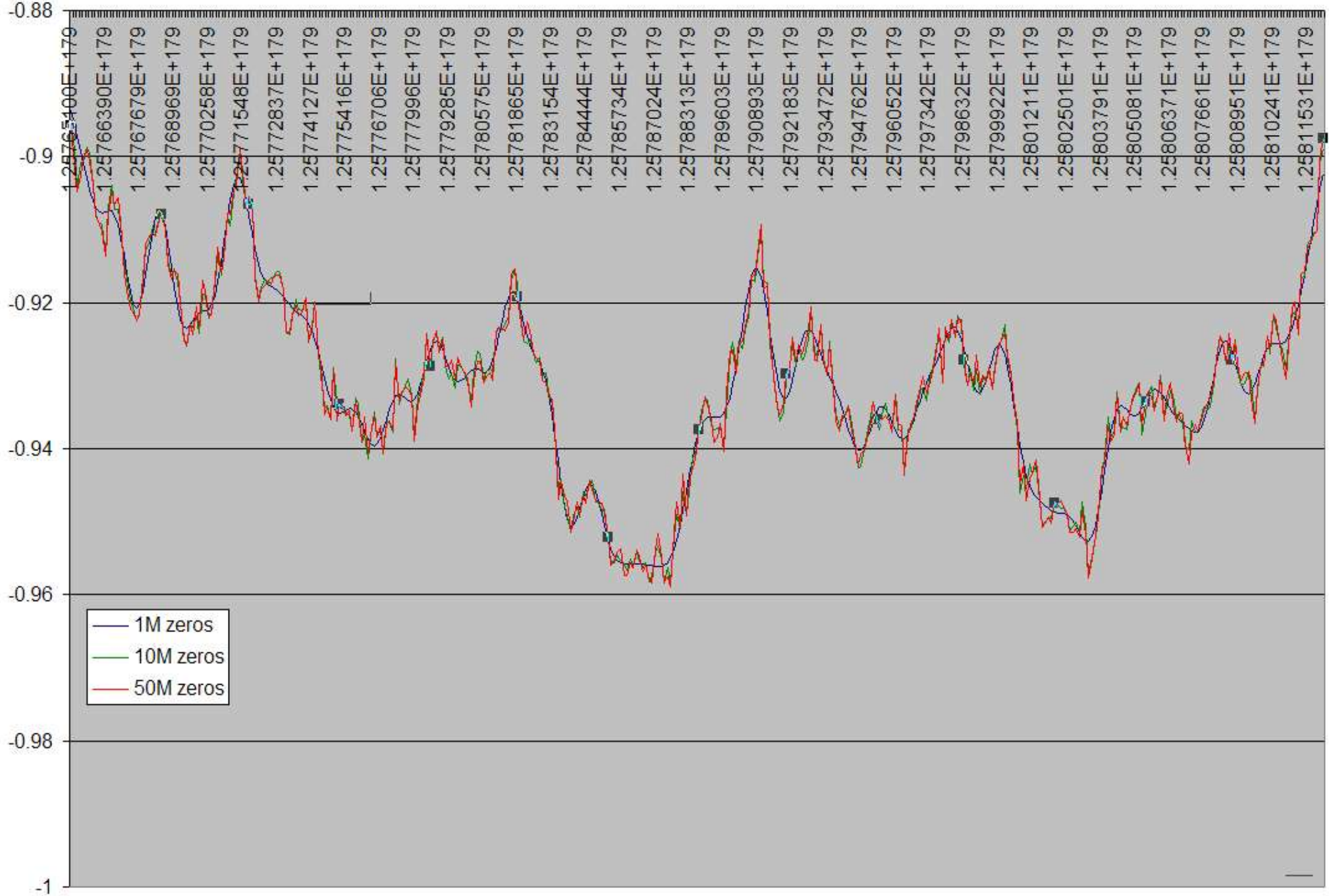
Since the greatest error at 1M zeros can only be 0.02, there is only an extremely low probability to find a new crossover in that region. Finding an error of 0.04 is an ultra rare event

Zoom of region around 1.258e179



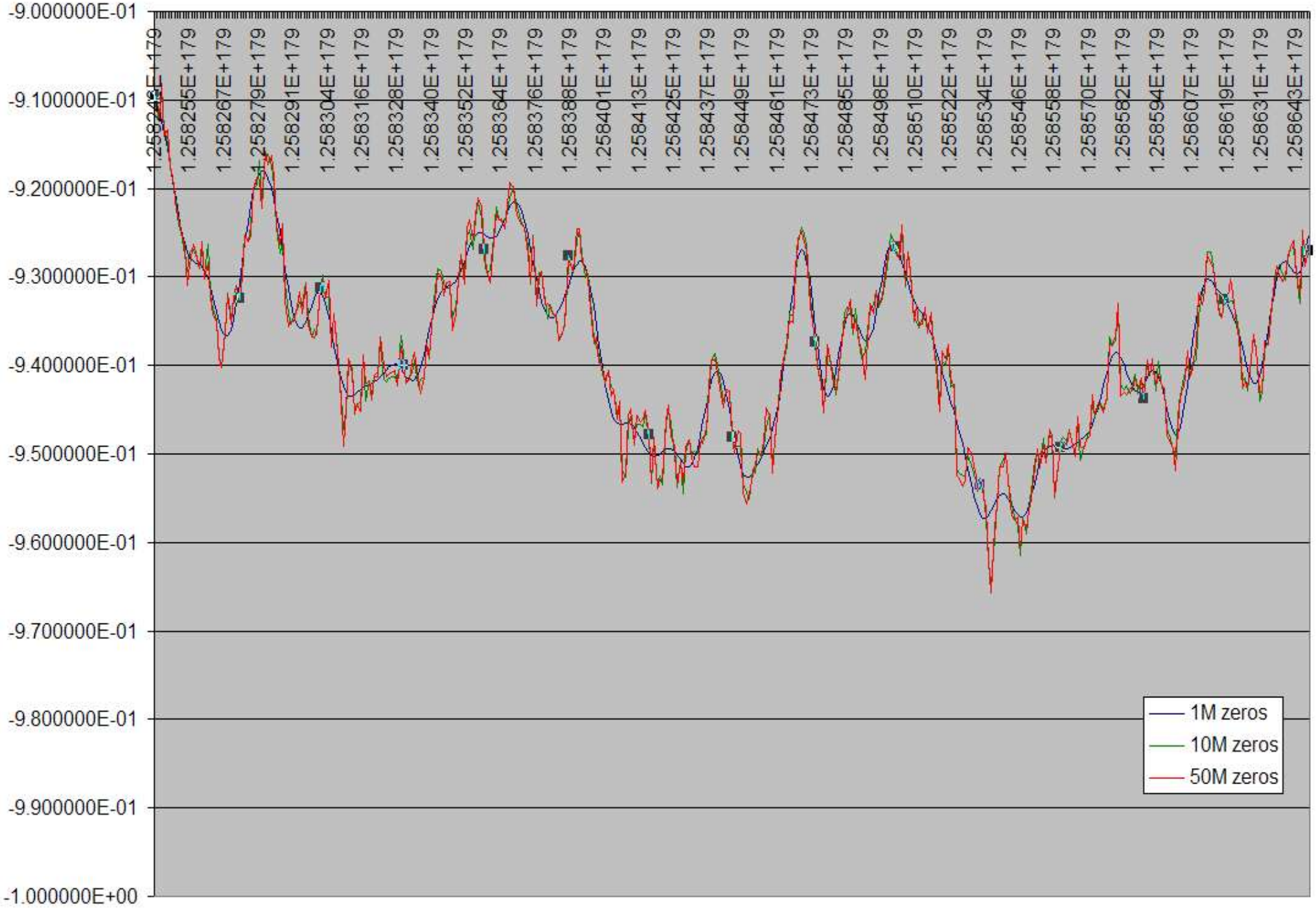
Same remarks

Zoom at region around between 1.2576e179 thru 1.2581e179



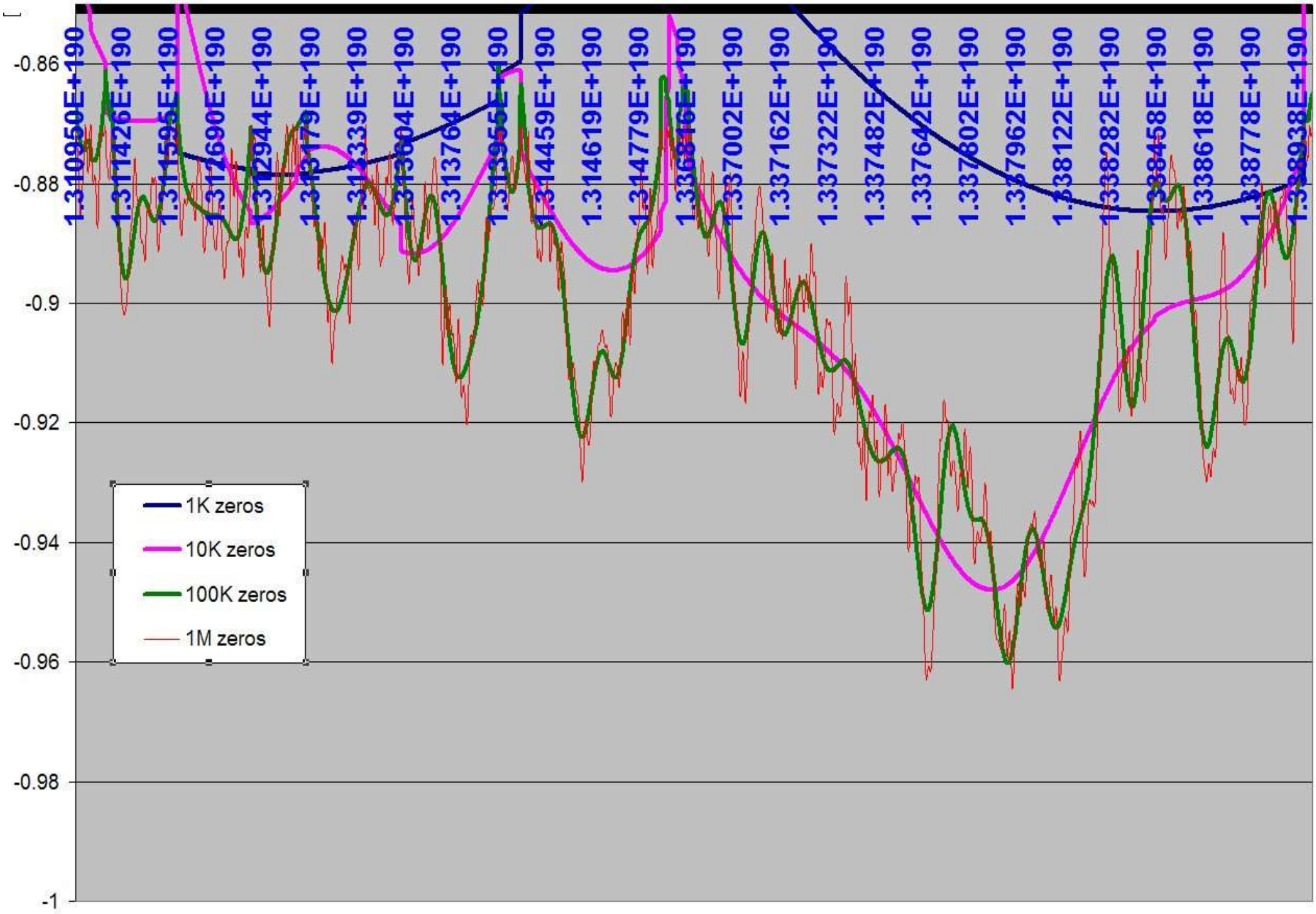
At 50M zeros, the greatest potentially observable error is below 0.005, almost 8 times smaller than what we need to see a crossover, then it is impossible to find a crossover there

Zoom region around between 1.2576e179 thru 1.2581e179



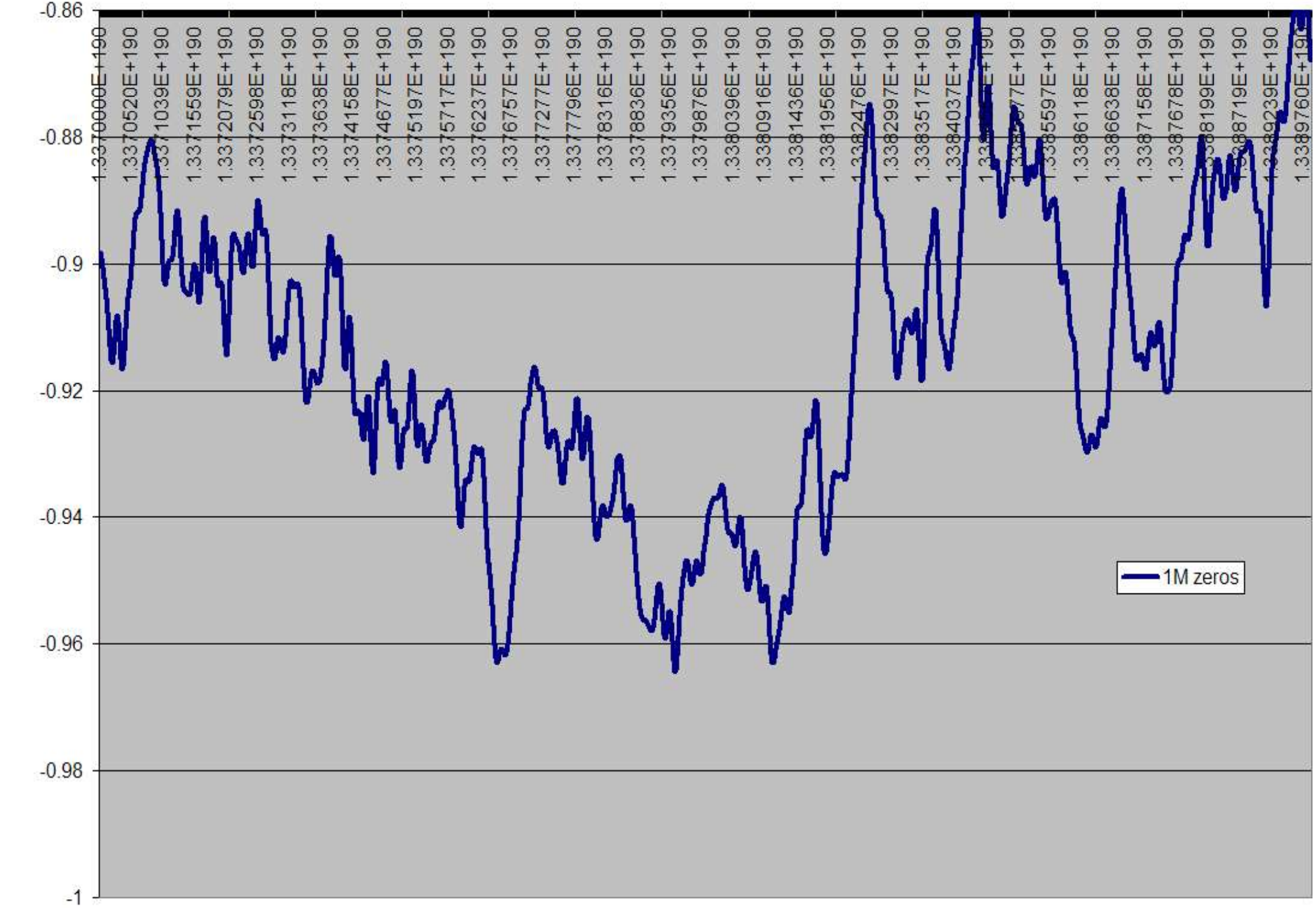
Similar remarks

Zoom at 1e190



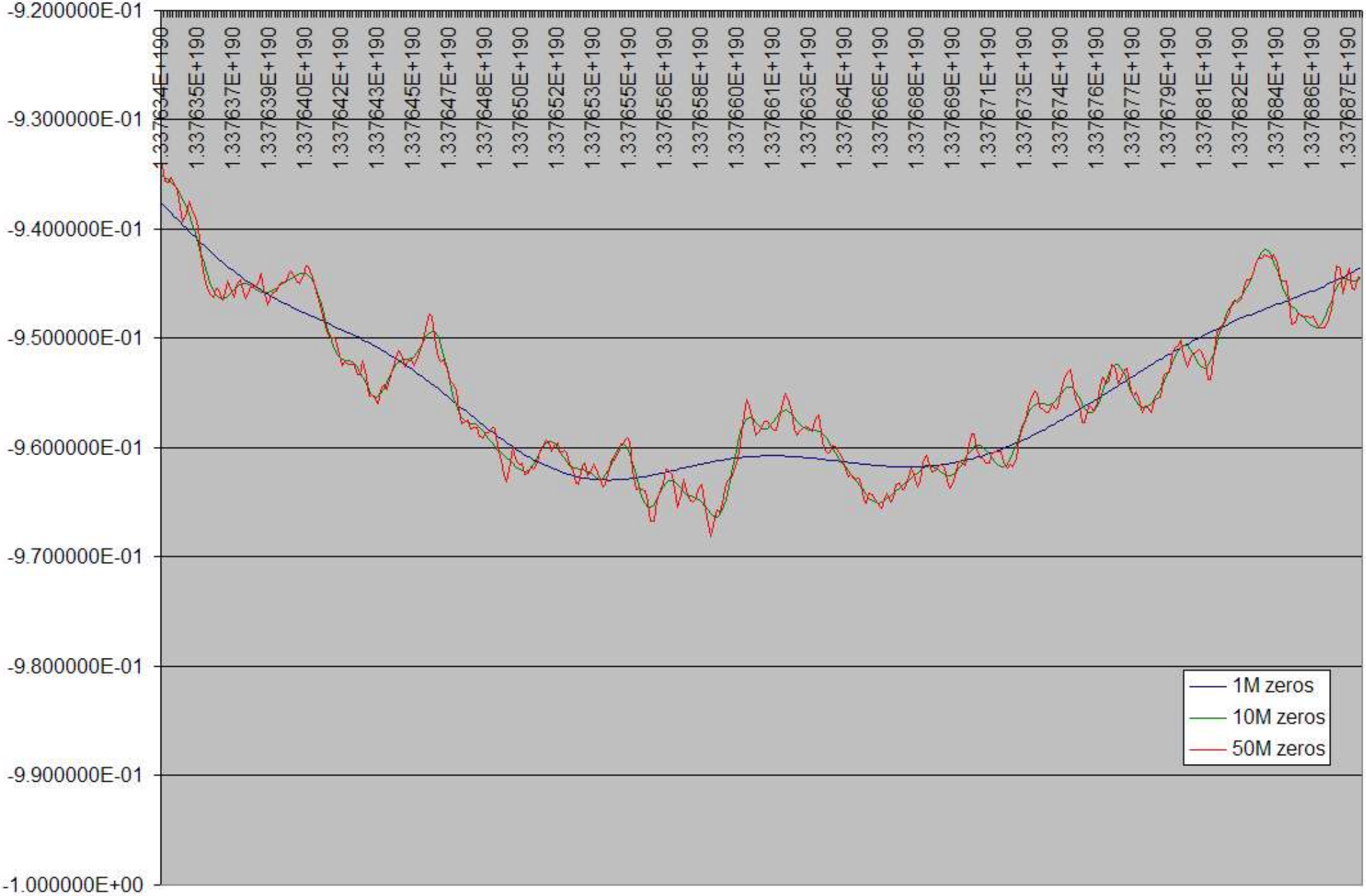
Again no chance to find a crossover for the same reason, but we will check with greater zoom

Zoom around 1.337e190



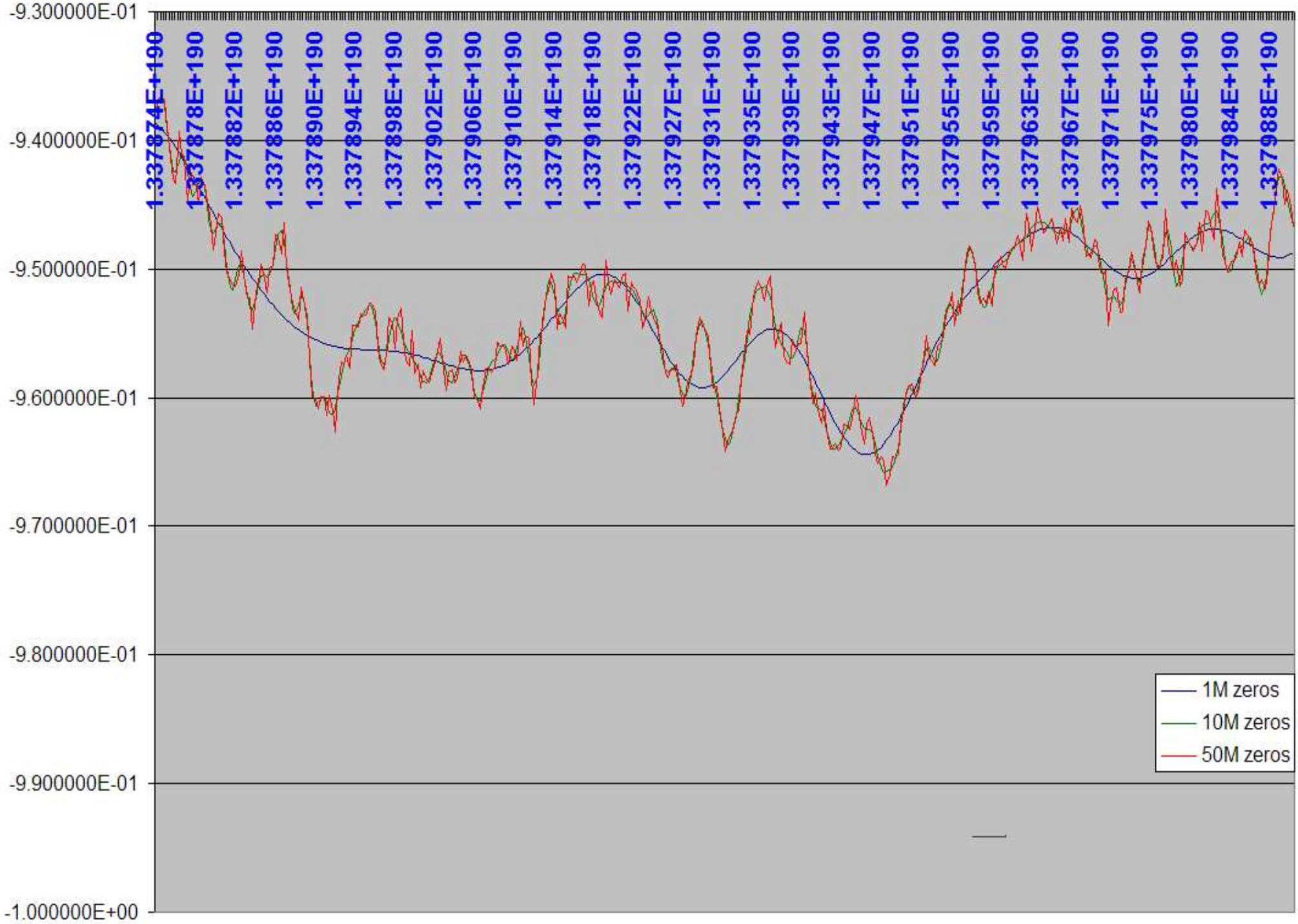
We can observe 3 critical regions that we will zoom

Zoom at region 1.3376e190



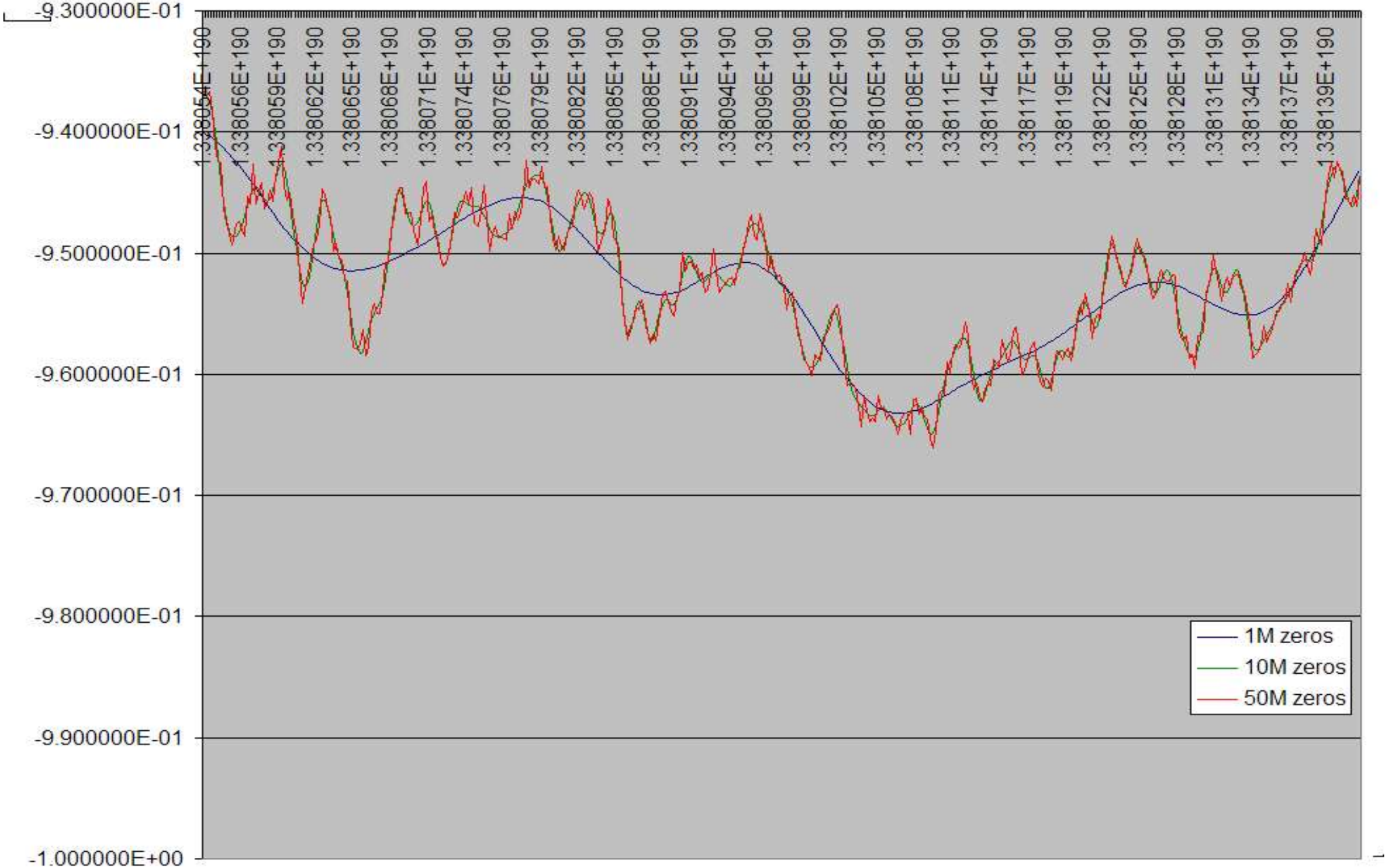
At 50M zeros is greatest observable error is ~0.005, almost 6 times smaller than what we need to see a crossover then it is impossible to find a crossover there

Zoom at region 1.3379e190



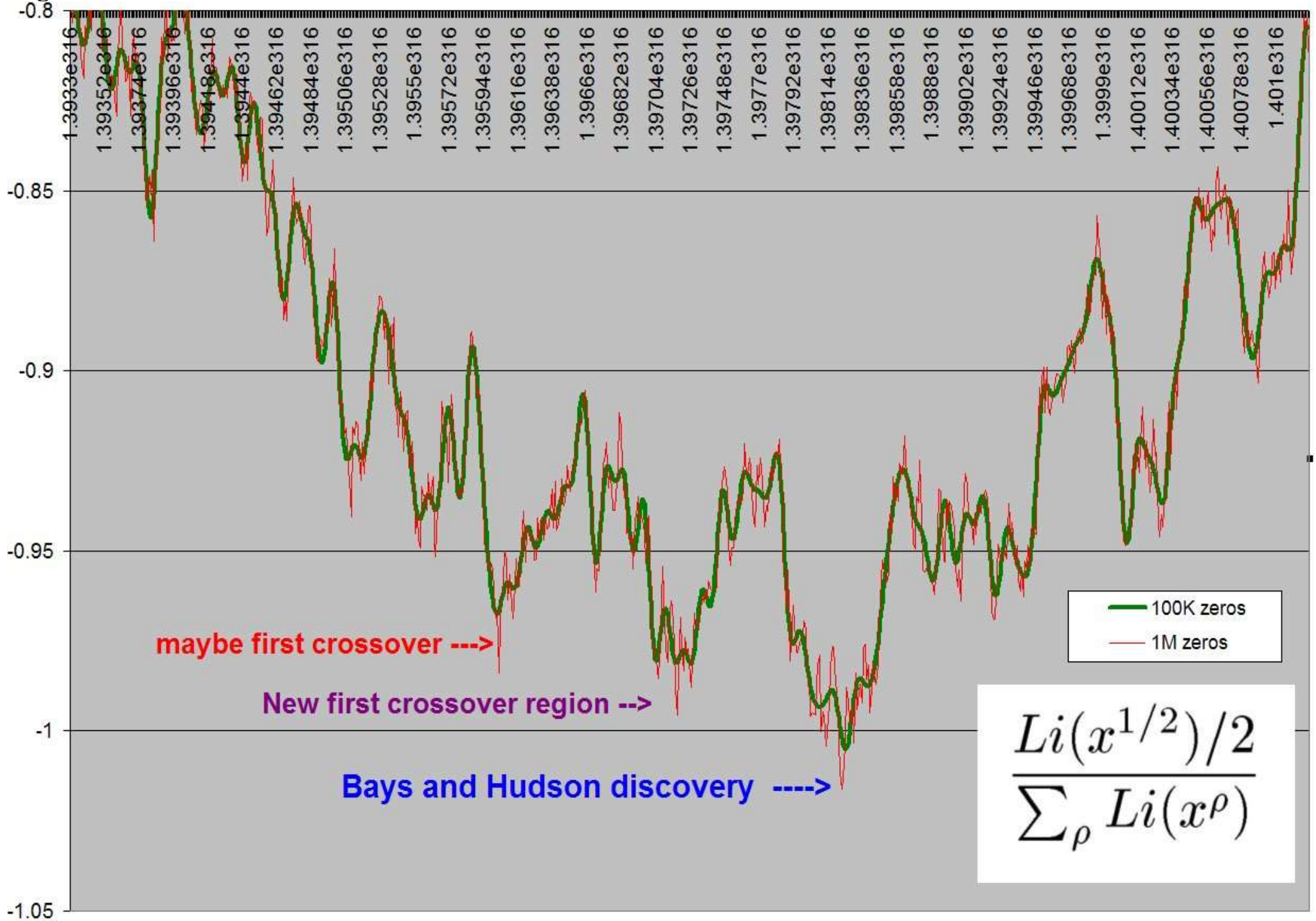
Same remarks

Zoom at region 1.3380e190



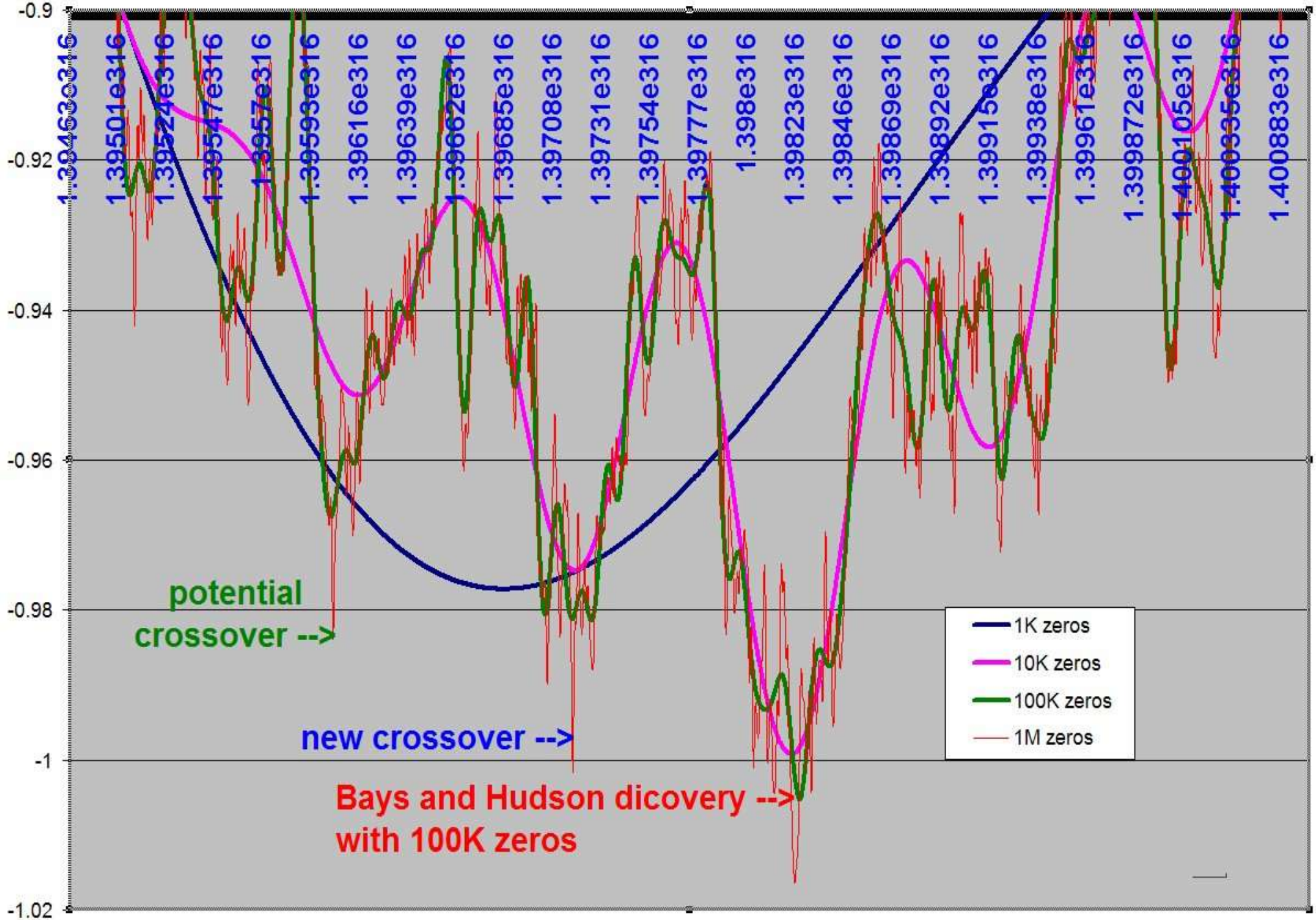
All those regions are more than 3% apart from the critical line
The greatest observed error at 50M zeros is ~0.5%, then taking into account the curve of the distribution of the high group of zeros, it seems improbable to find one value 6 times greater whatever the number of zeros
In fact this extremely low risk could easily be reduced significantly if you are not convinced

High altitude view at 1e316



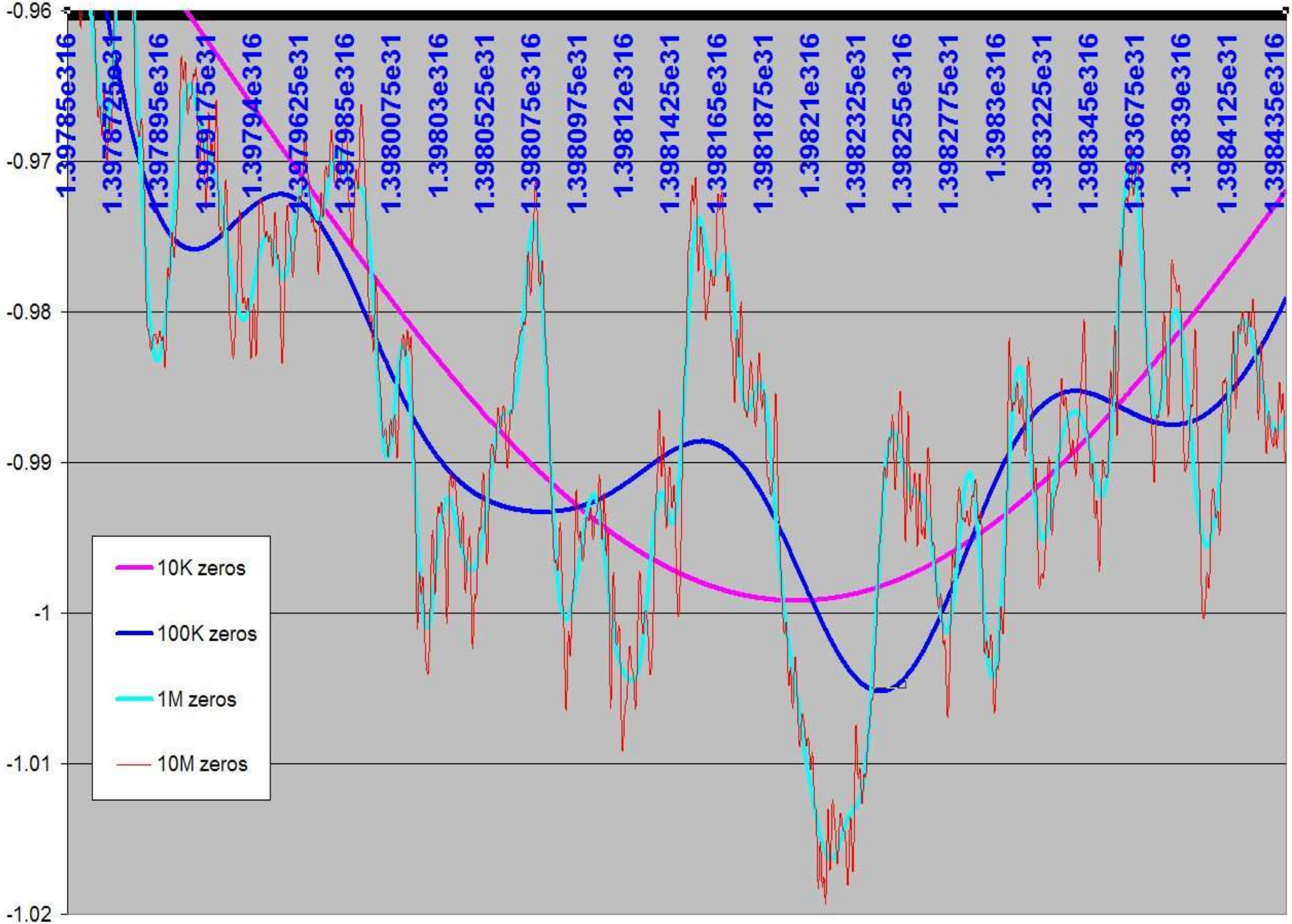
We have to test 2 regions 1.396e316 and 1.397e316

Zoom at 1e316

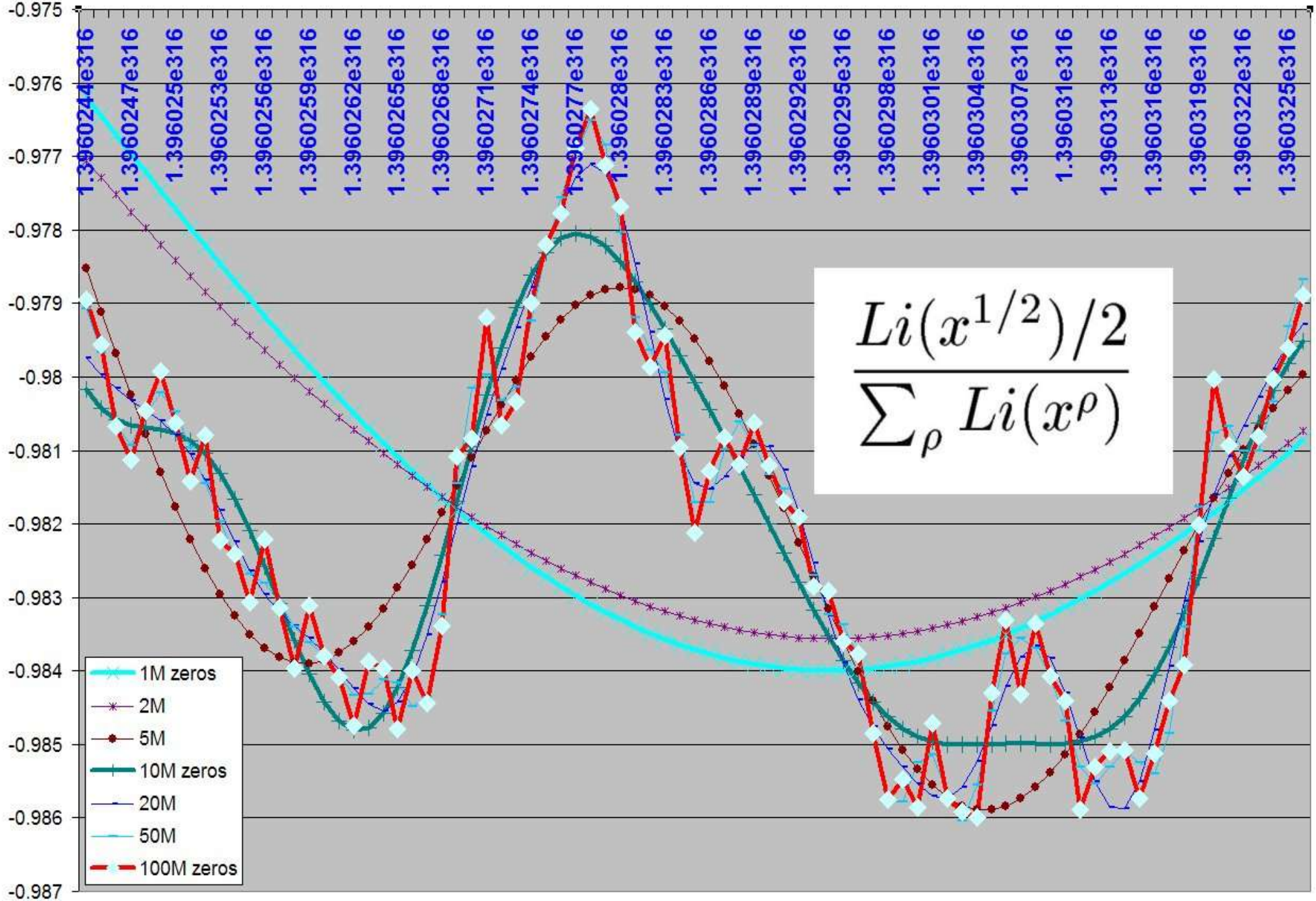


I need another zoom level to see the details of the potential crossover regions

Zoom on Bays Hudson region

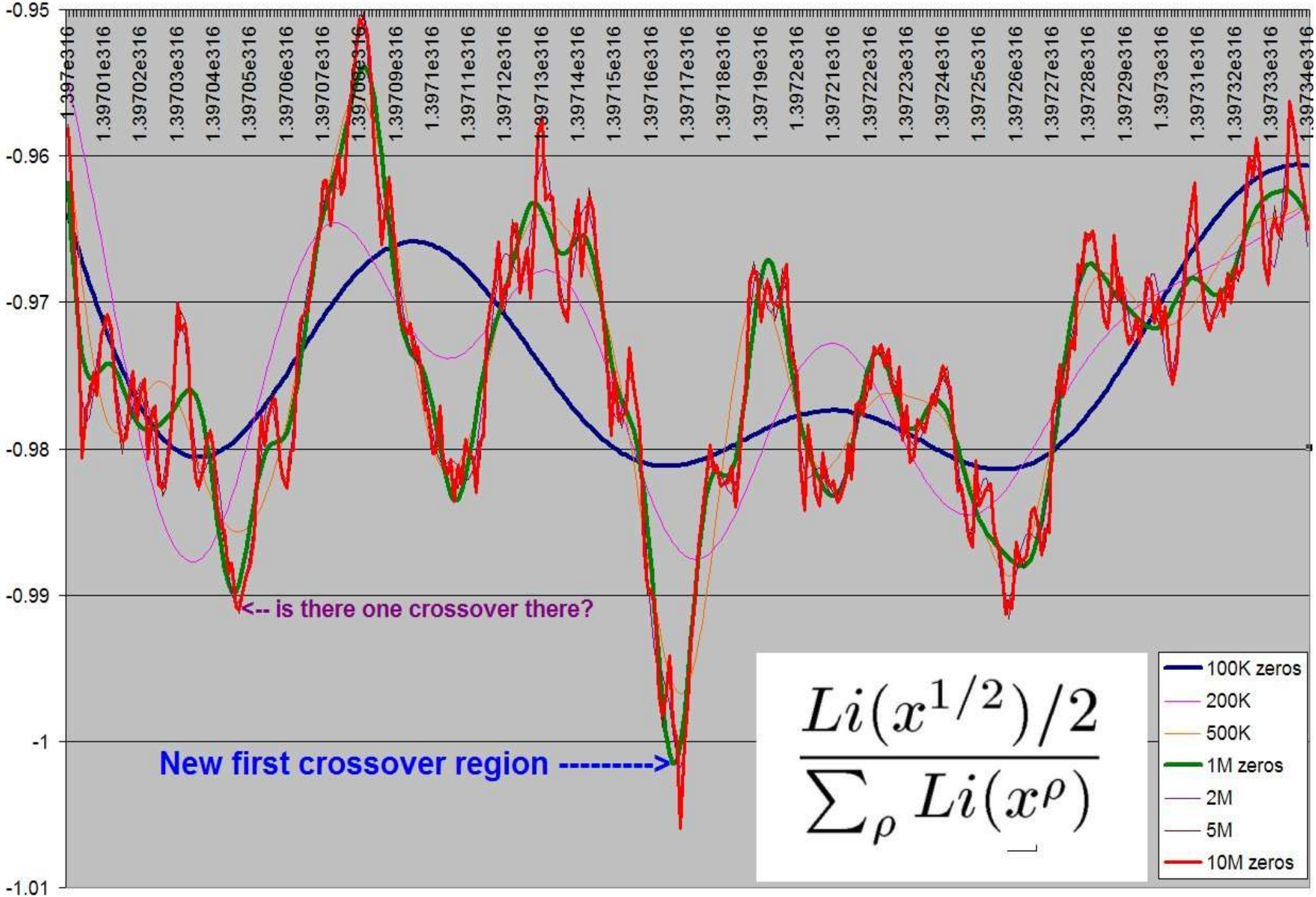


Big zoom at the potential region 1.396e316



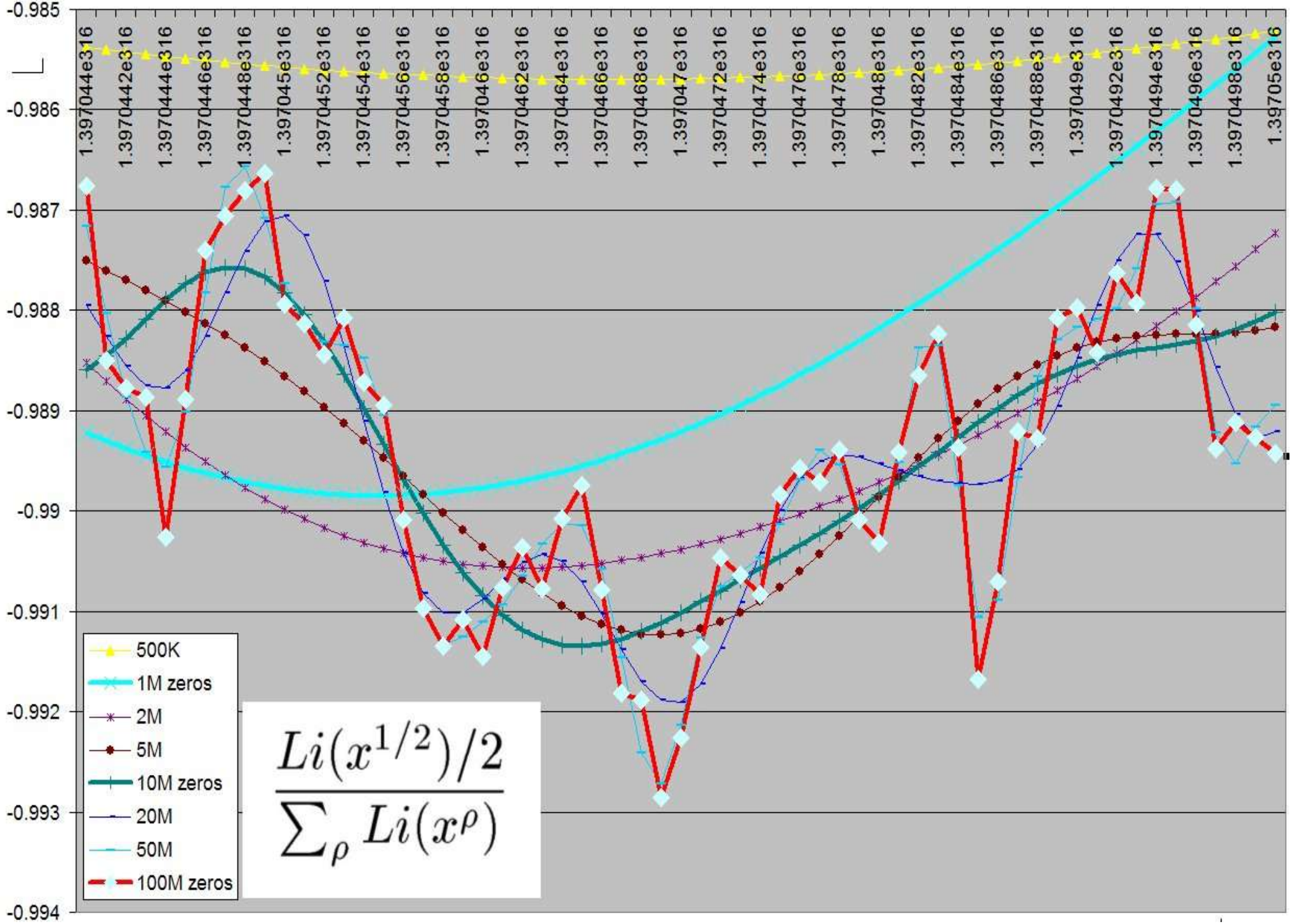
At 100M zeros the cumulative error is below 0.003 then it impossible to find a zero in that region

Big zoom at region 1.397e316



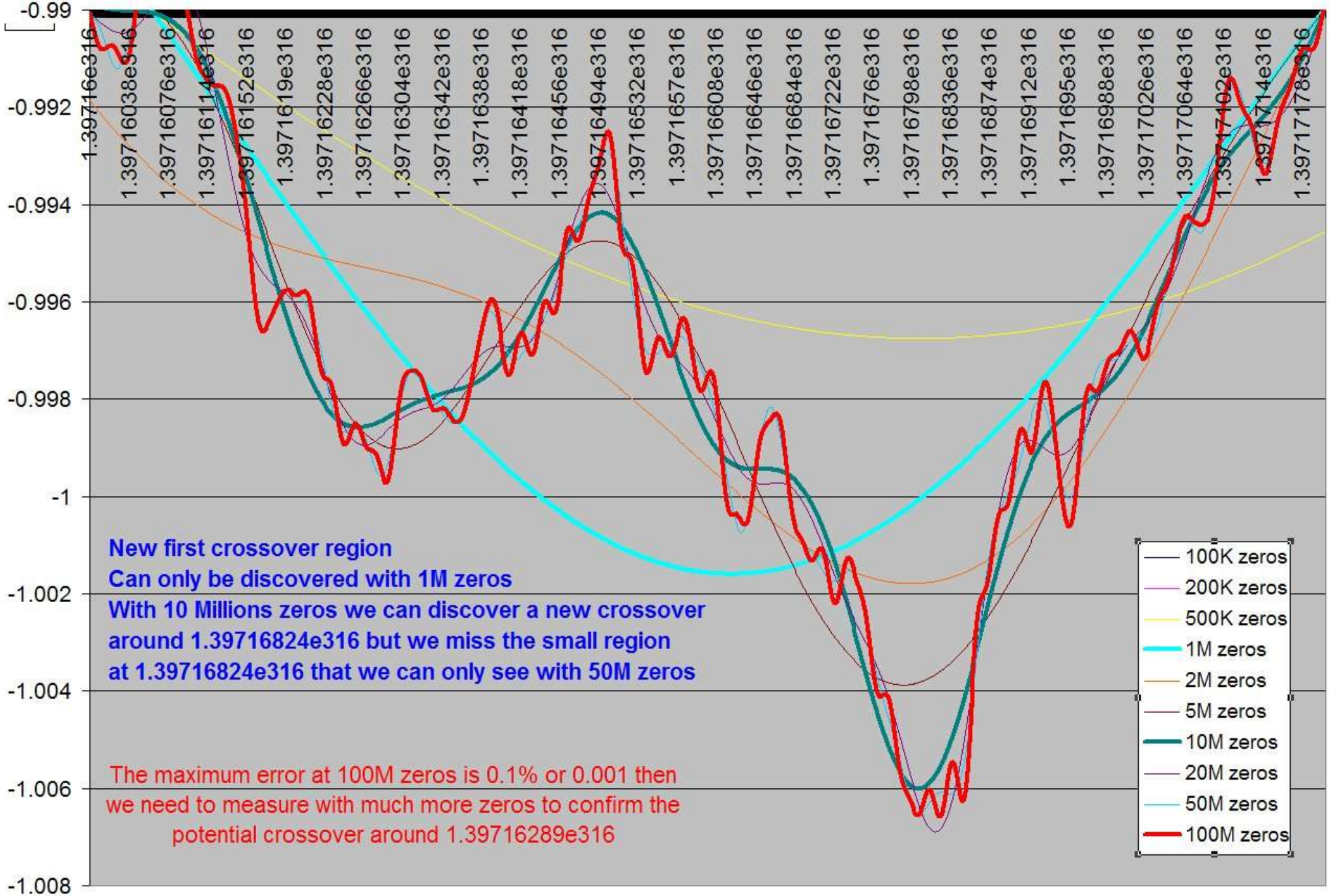
There is another potential crossover around 1.3970e316
There is a neat crossover around 1.3971e316

Zoom around 1.3970e316



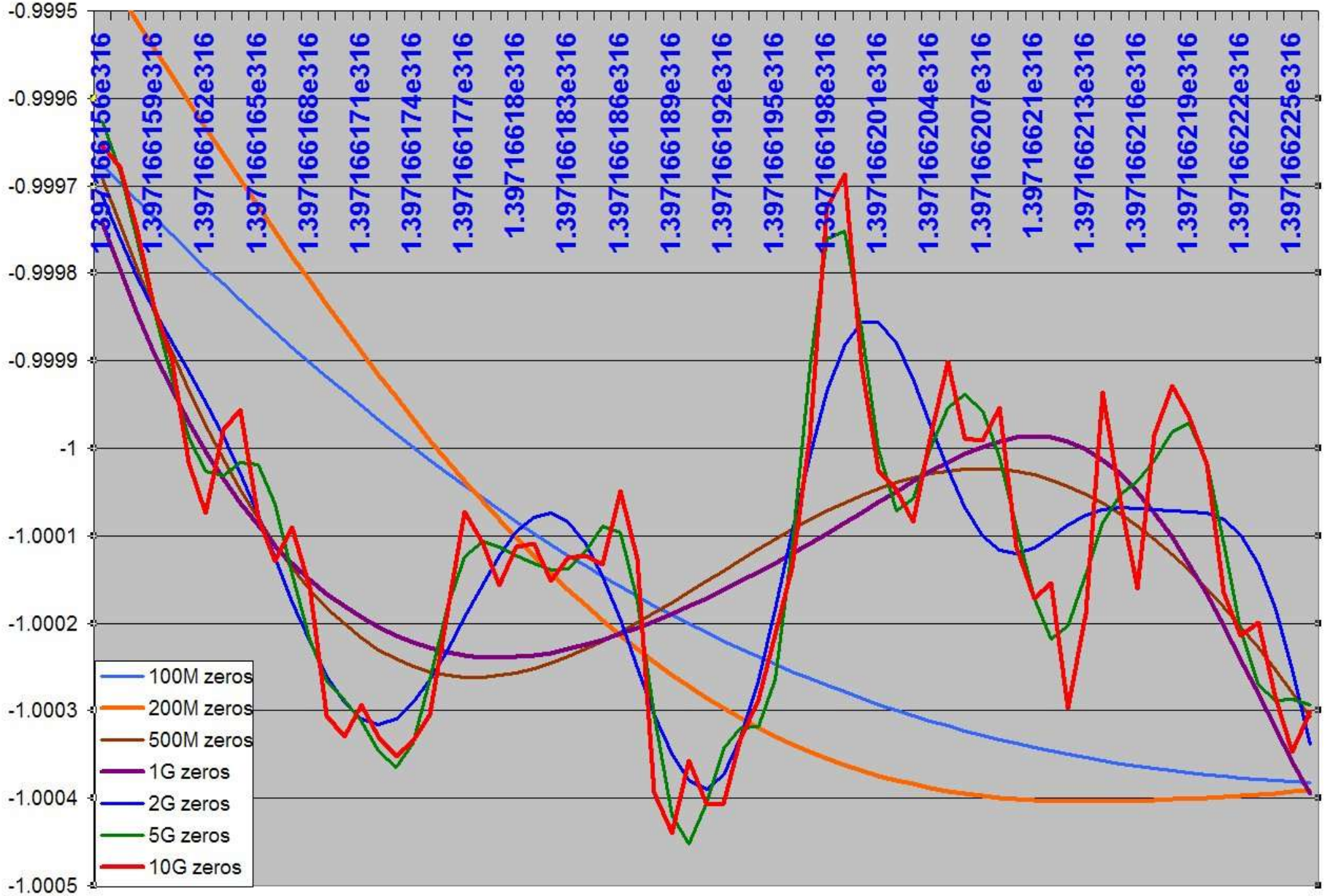
Again no chance to find a crossover there

A big zoom of the new first crossover region around 1.397168e316



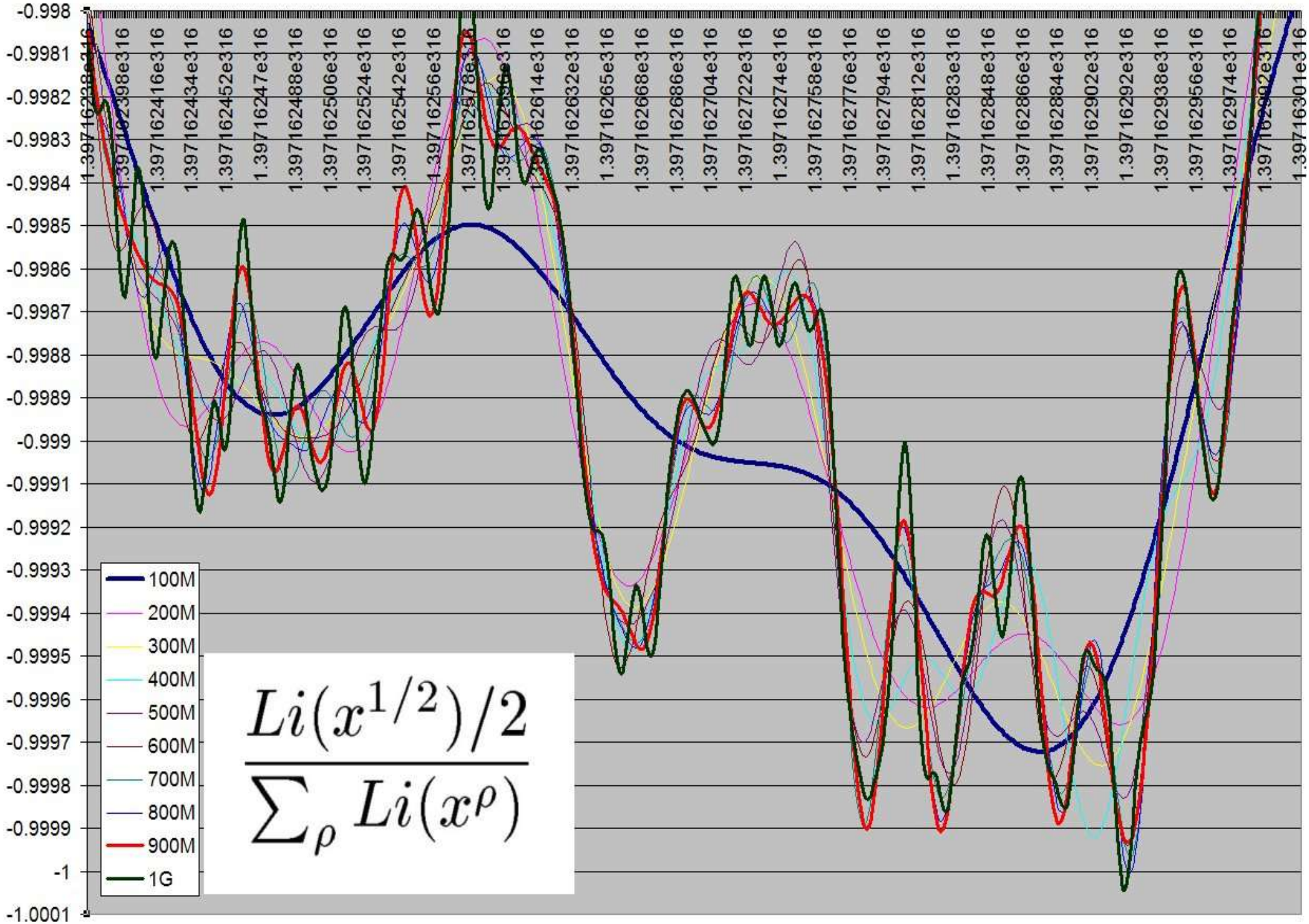
There is a potential crossover around 1.397162e316
A new first crossover appears around 1.397166e316

Zoom at 1.397166e316



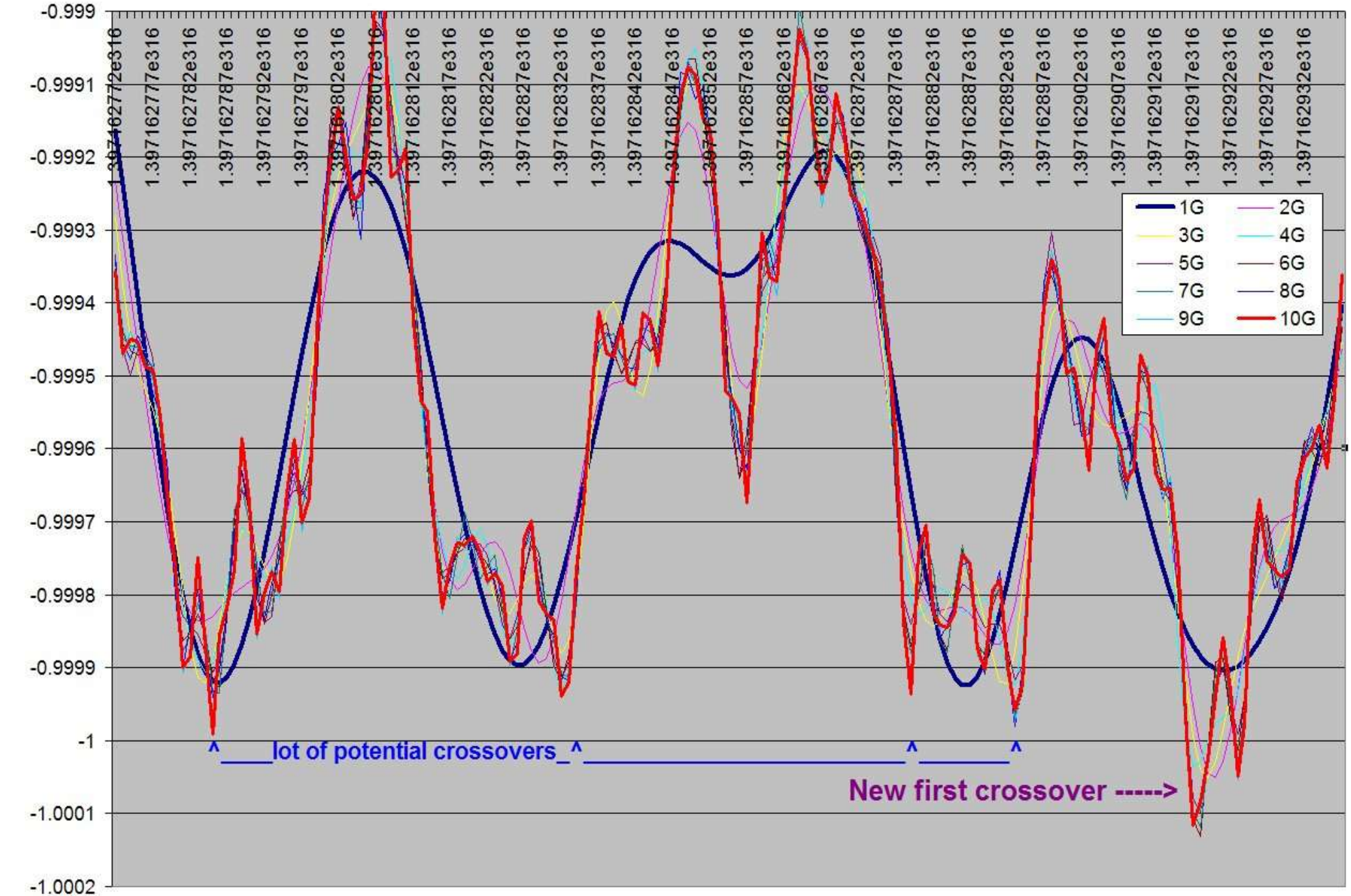
We can see clearly the details appearing, and another new crossover at 1.397166162e316
I use fewer points because the compute time becomes very huge due to the number of zeros taken into account

A big zoom at region 1.397162e316 where it seems we have a new crossover



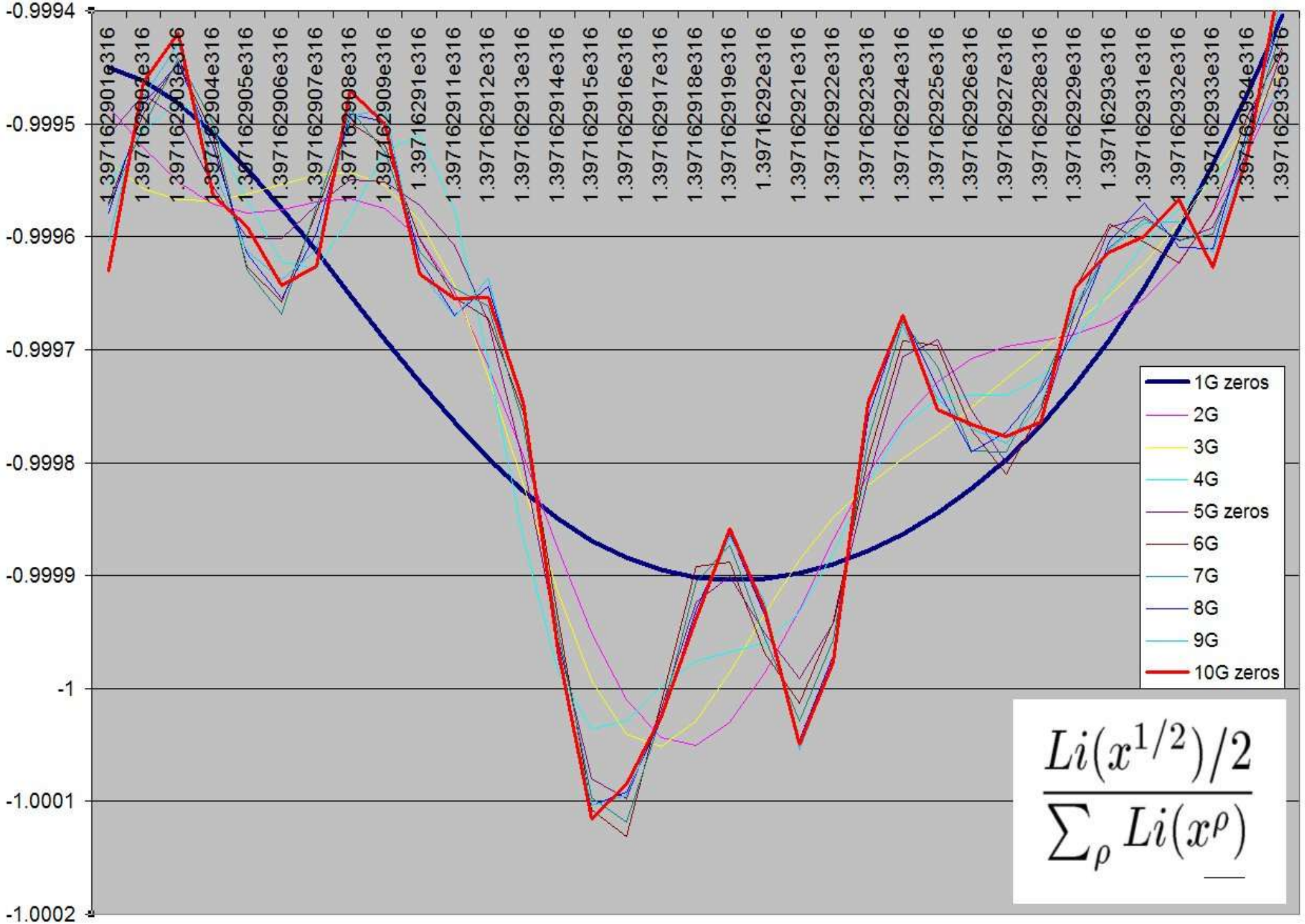
We need to zoom the right part there are 4 potentials regions, now extended precision is a MUST

The first crossover is there ☺ But where, there seems to be many candidates ?
This is becoming very hard and long, we need now much more than 10G zeros



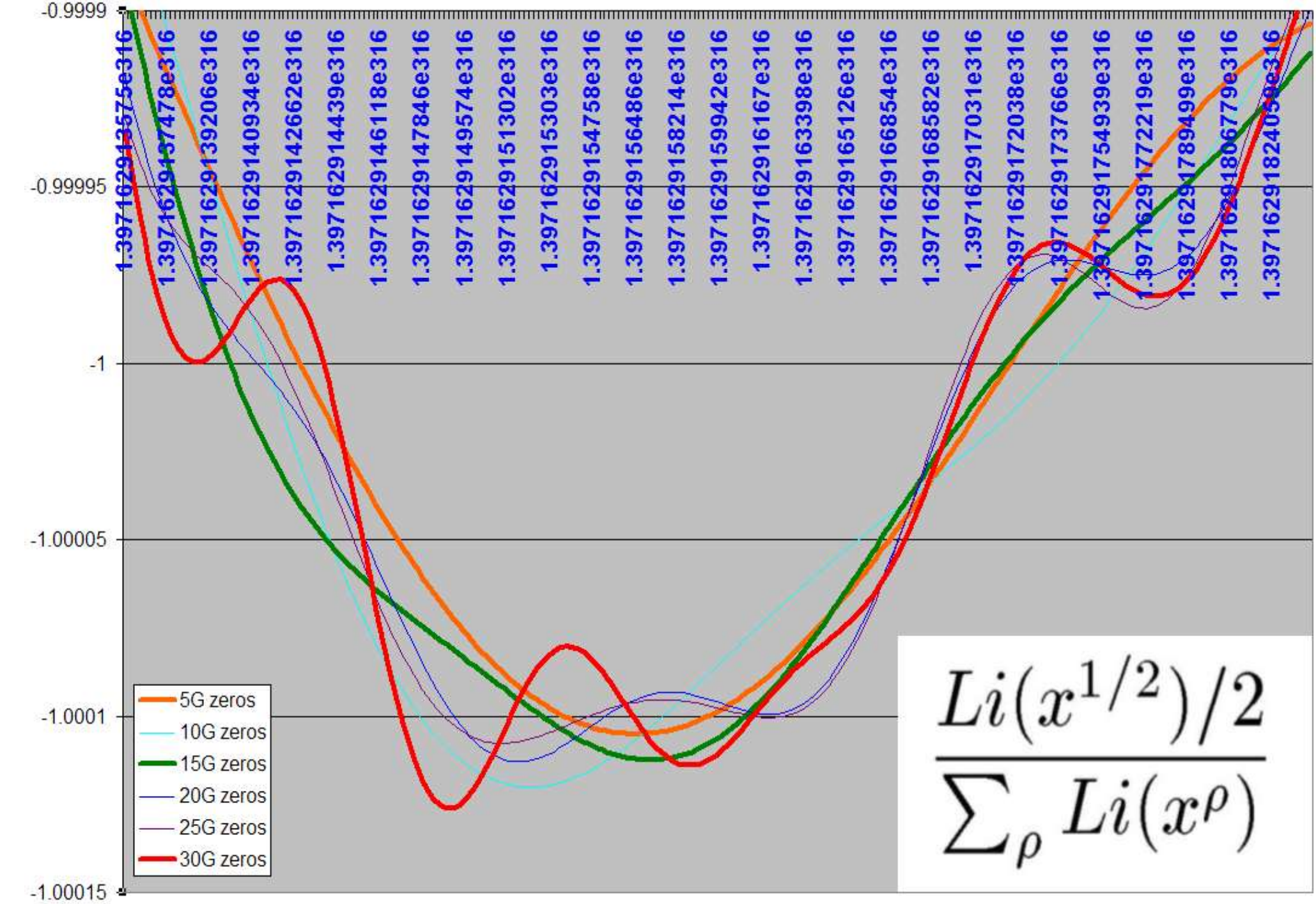
There are 3 potential regions and 1 clear crossover

First look at the new first crossover clearly appearing in the fourth region



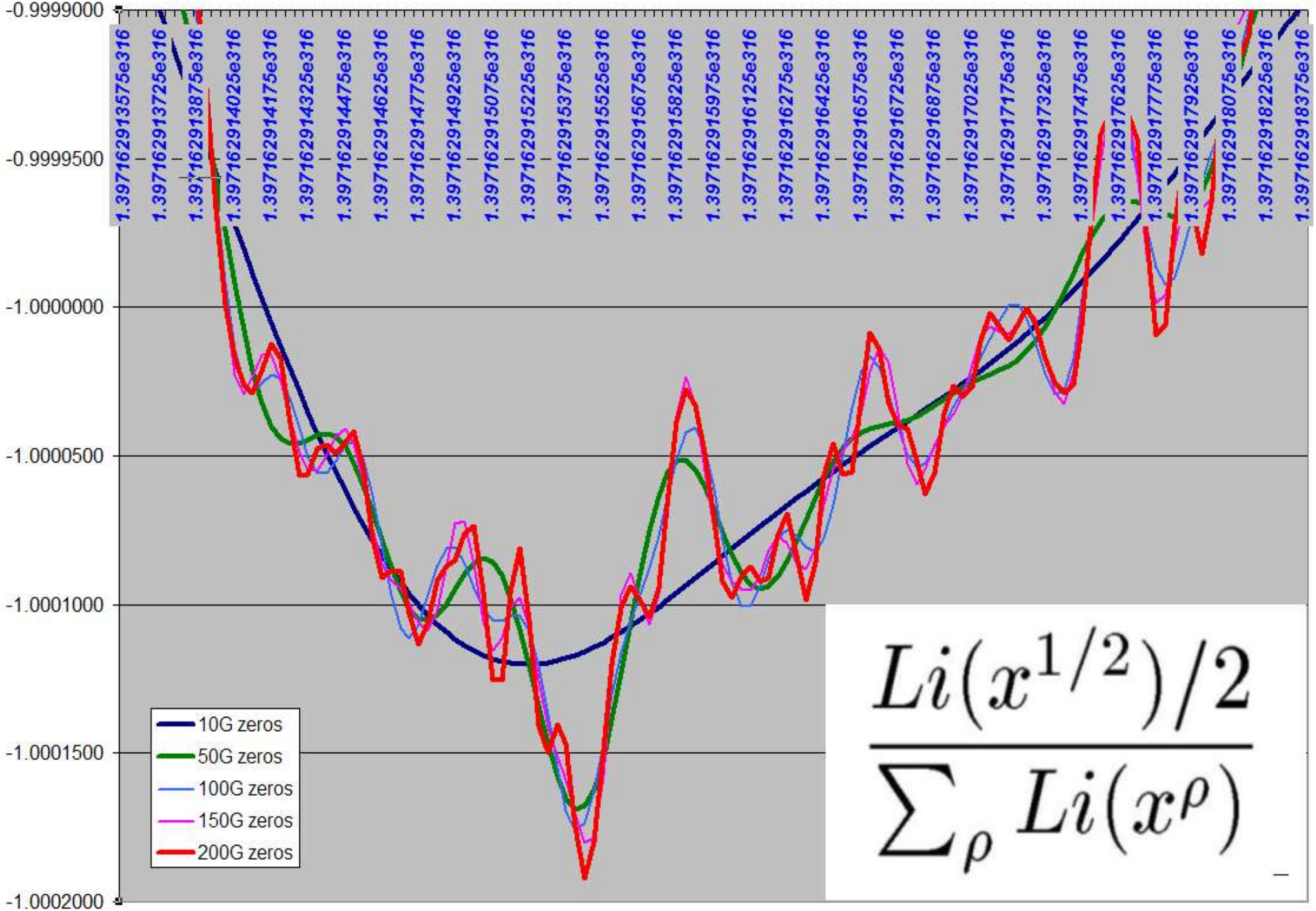
We need a zoom with more points to have a much better definition of the crossover.

Zoom of the previous slide computed in "extended" precision



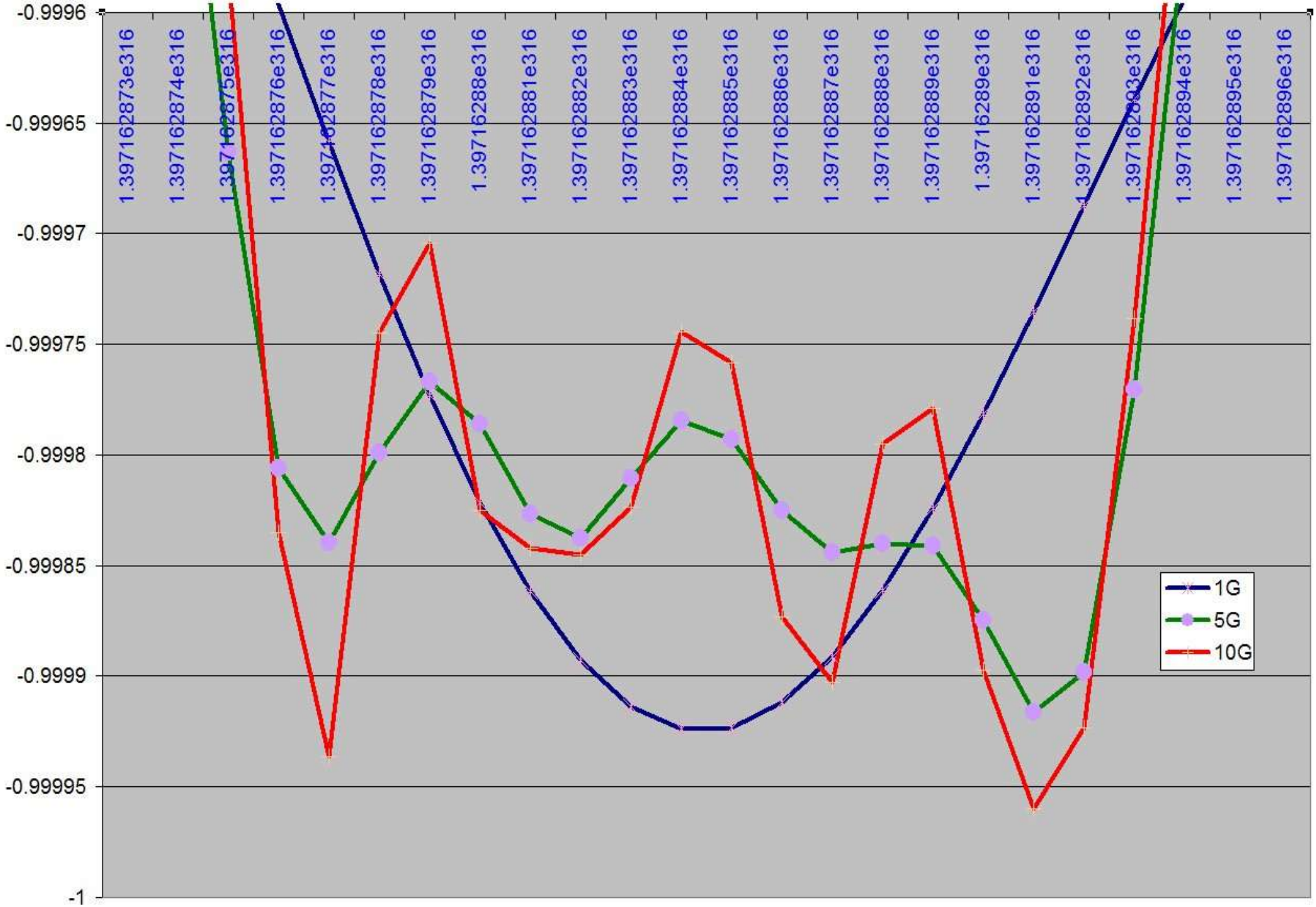
We can see a clear crossover, the current best value 1.397162914e316

The same region computed in “double” with 200G zeros



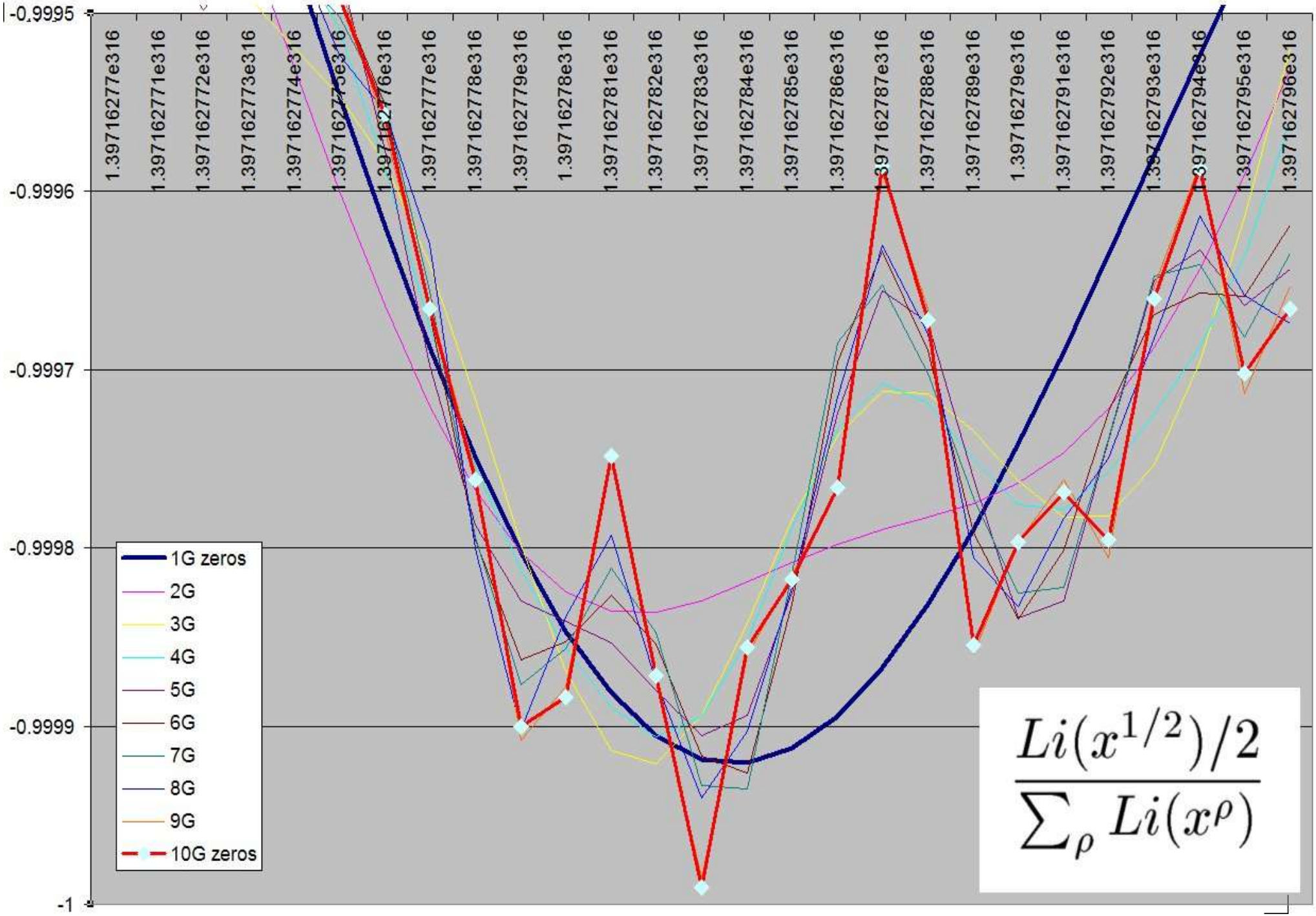
We can see that the first crossover is probably precisely around 1.397162914e316

A zoom at 3rd region between 1.397162878e316 and 1.397162895e316



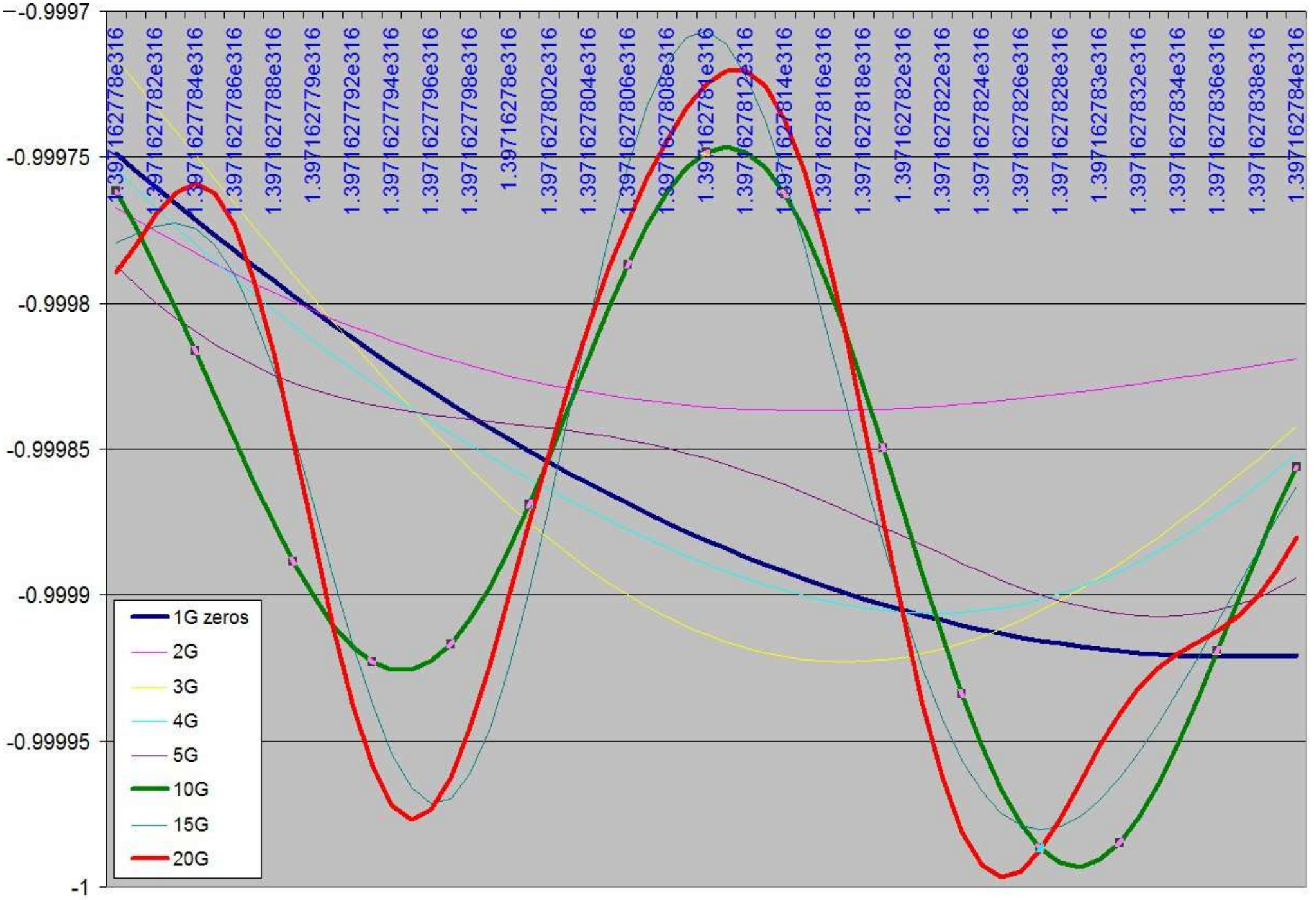
We will need to zoom the 3 regions around 1.397162877e316, 1.397162887e316 and 1.397162891e316

Zoom on the 1st very potential region



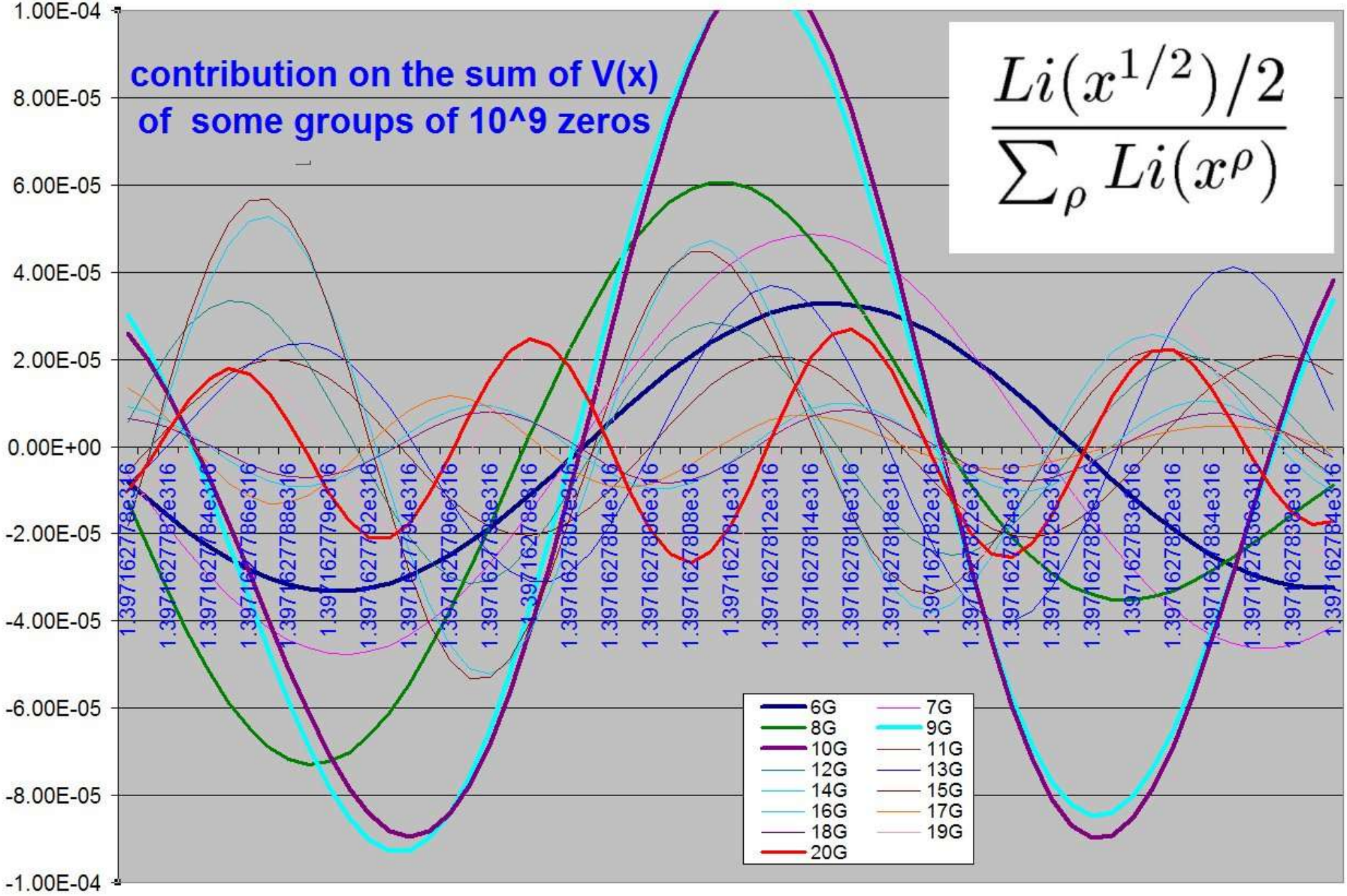
We will need another big zoom around 1.397162783e316 with much more points and zeros

Zoom of the previous slide



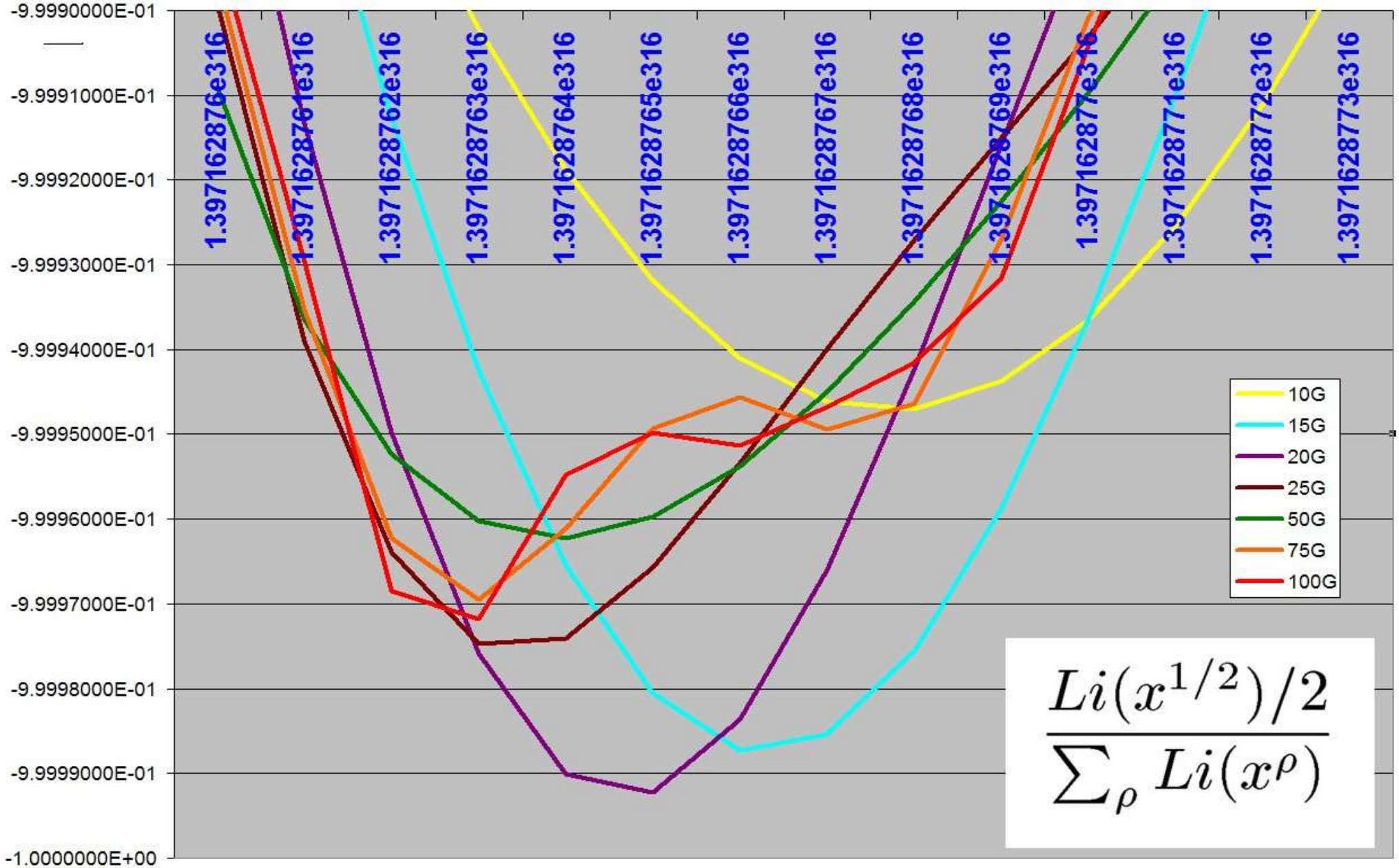
We can observe that there are 2 potentials regions that we need to zoom again we will probably need 50G zeros

I will show the contribution on the previous slide of some groups of 10^9 or 1G zeros on V(x)



We can observe on the previous page slide that the shape of the 10G is mostly caused by the coincidence that 2 major sinusoids "9G" and "10G" are almost perfectly aligned
You have probably now a very good idea how the curves add or cancel each others

A potential new first crossover computed in extended precision



This one is surprising, with 10G, 15G and 20G zeros, I had the impression that this could be the first crossover, but with much more zeros this turned out to be false

Imagine that the '20G line' had crossed the -1 V(x) line and I had stopped the computations, I could have been wrong, this demonstrates that we need to extent the computations until we have a clear evidence that this is a valid crossover

Some conjectures and problems, I hope some are new ☺

There exist many numbers x where $Li(x)$ crossovers $\pi(x)$ even if we use only 27 complex zeros in U(x) but you will never find a crossover if you use only 26 complex zeros
We can also search where the first crossover occurs with a small number of zeros for example with only 37 zeros we can discover a crossover at 2.6e304155332 or 6.7e367950350 or 9.34e77164561, but where is the first crossover with 36 zeros or less ? Remember to compute in double precision "30 digits" at least

The greatest absolute value for V(x) found today is 1.5 , can you find a value greater than 2.0 ?
2.0 should appears before 10^10000000000000000000 or 10^10^20
10.0 should appears before 10^10^10^3 ☺

In the same spirit, can you find a very large crossover covering entirely an integer exponent?
In others terms, is there a region of V(x) where the crossover exists from a*10^x to b*10^(x+2), the entire region 10^(x+1) being under the crossover line V(x) < -1?

Can you precise the parameters for the Gauss curve for the distribution of V(x)?

When we augment the number of zeros by 10, we can observe that the precision improves by a number very close to 2.5, can you give a more precise value for it?

Can you find another first crossover ?
Don't forget that even if the risk is extremely low, there is still an infinitesimal chance that a crossover exists before 1.397162914e316, and the zeta function is so surprising that maybe there is a trap region somewhere, where a huge number of zeros contribute all together to make a crossover somewhere where we don't expect it

Thanks to the courageous people that went thru the complete paper ☺
I hope some of you will be encouraged to extend this work

But remember all you need before starting the work:

- A huge list of zeros: probably more than 10^10 but some problems need very few zeros
- A huge list of true values for $\pi(x)$ with large values of x
- A very efficient way to compute V(x) for very large values of x
- A complex trigonometric library in extended precision larger than 64 bits
- Lot of compute power and patience it will be probably a multi years effort
- And as always inspiration and luck

Don't be discouraged by the numerous challenges ☺

Friendly,
Patrick