Number Theory / *Théorie des nombres*

Hilbert's tenth problem.
Enumeration of the number of solutions on an elliptic curve by the local/global sieve.

Hubert Schaetzel

Abstract In various articles, we have proposed a method of counting integers solutions of an asymptotic branch diophantine equation, i.e. an equation with an infinity of solutions such as for example 2n = p-q, n integer and p and q prime numbers. We had seen that the accuracy of the results increased with the size of the calculations made. This method makes it possible to find all the formulas of mathematical literature. We want to see here whether it retains a part of accuracy when applied to a diophantine equation with a finite number of solutions. We chose essentially for the purpose the case of an elliptic equation curve $y^2 = x^3+5x+c$.

Dixième problème de Hilbert. Dénombrement du nombre de solutions sur une courbe elliptique par crible local / global.

Résumé Nous avons proposé, dans divers articles, une méthode de dénombrement des solutions entières d'équations diophantines ayant une infinité de solutions comme par exemple 2n = p-q, n entier et p et q nombres premiers. Nous avions vu que la précision des résultats augmentait avec la quantité de calculs effectués. Cette méthode algorithmique permet de trouver l'ensemble des formules de la littérature mathématique. Nous souhaitons voir ici si elle garde une part d'exactitude lorsqu'elle est appliquée à une équation diophantine à nombre fini de solutions. Nous avons choisi principalement pour la discussion le cas d'une courbe elliptique d'équation $y^2 = x^3+5x+c$.

Statute Preprint
Date Version 1 : June 13, 2020
Version 2 : June 29, 2020

Summary

Signs and abbreviations

If(x, y, z) : If x is true then y otherwise z.

1. Preamble.

A diophantine equation is a polynomial equation with one or more unknows and integer coefficients. David Hilbert's tenth problem "Of the possibility of solving a diophantine equation" raised the question in 1900 of the existence of a general algorithmic method with a finite number of steps enabling to decide, for any diophantine equation, whether this equation has integer solutions or not.
Yuri Matiyasevich's theorem answered in 1970 by the negative. It establishes that diophantine sets, which are the sets of integer solutions of a diophantine equation with parameters, are exactly all recursively enumerable sets, which means that such an algorithm cannot exist [1].

In Hilbert's problem, there is no need to explicitly find solutions, as the problem is in terms of "decision". Thus, determining the number of solutions in a given diophantine equation, or of a given type, ought to be considered as a specific problem. The rest of our presentation contradicts that statement, **except** that we considerer now an algorithm with an infinite number of steps of calculations.

In another article [2] and various related articles, we proposed a method of counting integer solutions of diophantine equations with asymptotic branches, i.e. having an infinite number of solutions, such as 2n = p-q, where n is an integer and p and q are prime numbers. This method leads to all the know formulas of mathematical literature, such as that of Goldbach, Vinogradov, Hardy-Littlewood, Iwaniec/Friedlander, Waring, quadratic forms, not limited to these at all. The relative accuracy of the results on the asymptotes gradually increases with the size of the calculations performed. One of our articles [3] focuses also on the local-global Hasse principle and the derived problem of obstructions which is somewhat the first step in what will follow here. We have shown the richness of this principle as there is in fact a whole range of behaviours according to the chosen diophantine equation varying between obstruction and its opposite, the influx, and that this whole range emerges and can be explained by the "local-global" sieve implemented thereupon.

We would like now to apply the same algorithm in the case of a diophantine equation with a finite number of solutions and to see up to which stage this algorithm may still be efficient. We chose the case of an elliptic curve of equation $y^2 = x^3+5x+c$. Here c is a parameter, called the target, and depending on its value, we want to know whether the equation has a solution or not, and more precisely, whether the number of solutions associated with each value of the parameter can be found indirectly.

The object here is not to re-explain the whole method. The reader will refer to our online site for more details. We will only repeat the minimum useful technic here, especially since the most important matrix tools are omitted here.

Let us thus have the elliptic equation :

$$y^2 = x^3+5x+c \qquad (1)$$

We examine the integer solutions of this equation by two approaches :

- directly by solving : $c = y^2-(x^3+5x)$
- indirectly by solving : $c = y^2-(x^3+5x) \bmod \infty$

2. Direct evaluation.

To find the number of solutions of $c = y^2-(x^3+5x)$, we give to x and y ever-increasing values and collect c-impacts for values between 0 and t.

$$0 \le y^2-(x^3+5x) < t$$

That is :

$$\text{if}(x < 0, 0, (x^3+5x)^{1/2}) \le |y| < (x^3+5x+t)^{1/2}$$

The list of solutions (c, x, y) for t = 200 and $-5 \le x < 10^{10}$ is given in Appendix 1. To a given x are related two solutions that are y and -y. For c = 0, we considered y = 0 as a single solution (the double solution case giving in any case roughly the same conclusion). It should be noted that there is no solution for $10^7 \le x < 10^{10}$ and if other solutions exist, they are a priori few.

The numbers of solutions #c, function of c, are (#(c = 2 mod 3) = 0 and ∏fac(c = 2 mod 3) = 0) :

Table 1

c	0	1	3	4	6	7	9	10	12	13	15
#c	3	2	2	2	1	10	2	8	0	0	2
∏fac(c)	1,3423	1,0198	1,2429	1,2834	0,1967	33,7016	4,5730	9,5176	0,0203	0,0324	2,0750
c	16	18	19	21	22	24	25	27	28	30	31
#c	4	5	6	0	10	2	4	2	0	2	6
∏fac(c)	4,8373	1,9255	9,2034	0,3411	16,654	0,2621	1,9546	0,8933	0,0430	1,6910	11,192
c	33	34	36	37	39	40	42	43	45	46	48
#c	0	2	2	2	6	0	5	18	0	8	2
∏fac(c)	0,0410	0,5754	1,9298	0,3094	5,8117	0,0327	5,7569	105,335	0,1318	12,846	0,1558
c	49	51	52	54	55	57	58	60	61	63	64
#c	4	2	0	2	2	2	12	2	2	6	2
∏fac(c)	3,9765	0,4489	0,0475	0,3784	0,9427	0,1551	39,297	0,2505	0,1861	7,6791	0,4570
c	66	67	69	70	72	73	75	76	78	79	81
#c	2	8	0	2	0	2	6	0	4	2	2
∏fac(c)	0,1982	9,8458	0,1006	1,5641	0,0804	0,2013	12,463	0,1915	4,6440	1,3939	1,7551
c	82	84	85	87	88	90	91	93	94	96	97
#c	4	1	4	2	2	0	2	2	2	0	0
∏fac(c)	5,5830	0,6950	1,8116	0,4800	0,4111	0,0525	0,6344	0,5506	1,2290	0,1087	0,0263
c	99	100	102	103	105	106	108	109	111	112	114
#c	2	6	6	2	0	14	0	6	0	2	0
∏fac(c)	0,5702	7,2978	15,0028	0,6345	0,0796	45,1877	0,0997	2,1580	0,0631	0,4764	0,0427
c	115	117	118	120	121	123	124	126	127	129	130
#c	6	0	2	2	2	2	2	6	6	0	0
∏fac(c)	6,5510	0,2287	0,5266	0,5030	0,5807	0,4634	0,2938	5,6389	8,6671	0,1295	0,0409
c	132	133	135	136	138	139	141	142	144	145	147
#c	2	2	2	0	2	6	2	2	2	0	0
∏fac(c)	0,0838	2,0068	0,4178	0,0267	2,0486	5,7498	1,1714	0,8913	1,2303	0,1792	0,0661
c	148	150	151	153	154	156	157	159	160	162	163
#c	4	5	8	0	8	0	0	2	0	4	6
∏fac(c)	3,2223	8,9185	7,8917	0,0911	10,792	0,1392	0,1924	0,6675	0,1088	2,2782	7,2483
c	165	166	168	169	171	172	174	175	177	178	180
#c	2	4	0	2	0	2	2	10	2	2	2
∏fac(c)	0,9395	1,6488	0,0550	1,3497	0,2882	0,4390	2,5166	16,5386	0,6366	0,3705	0,7797
c	181	183	184	186	187	189	190	192	193	195	196
#c	2	2	4	8	4	0	2	0	0	4	4
∏fac(c)	0,2863	0,4629	1,2692	14,2421	1,8924	0,0808	1,9291	0,1503	0,1817	2,2700	1,7433
c	198										
#c	2										
∏fac(c)	1,0794										

Graphic representations are more striking and we favour to analyse them rather than this table. The corrective factors ∏fac(c), reflecting the quantities of solutions, will pop up in the indirect method of determination and are thus explained below.

3. Indirect evaluation.

3.1. Modulo infinity.

We propose to solve parametrically $c = y^2-(x^3+5x)$. This is like searching the number of solutions of $c = y^2-(x^3+5x) \bmod n$ where $n \to +\infty$. If the number of solutions is finite then, for n large enough, all solutions are detected.
At infinity, we have all the solutions in the equation. Thus we write n in the form $n = 2^{n2}.3^{n3}.5^{n5}\dots p_i^{npi}$ where as well i as the powers np_i diverge ($p_i \to +\infty$, $np_i \to +\infty$).
With the Chinese theorem's background, we say that solving :

$$c = y^2-(x^3+5x) \bmod 2^{n2}.3^{n3}.5^{n5}\dots p_i^{npi} \qquad (2)$$

is like making the product of the results obtained for k = 0 à i :

$$c = y^2-(x^3+5x) \bmod p_k^{npk} \qquad (3)$$

The question being not to find c, but the number of solutions for given c, which we note #c, we write :

$$\{\#c \setminus c = y^2-(x^3+5x) \bmod p_k^{npk}\} \qquad (4)$$

varying x and y from 0 to p_k^{npk}-1.

If we accept that we equiprobably collect the number of solutions of the equation in this way, it is essential also in order to collect this probability to "normalize" the result because of the range of results [0, p_k^{npk}-1] includes p_k^{npk} elements, while we perform $p_k^{npk}.p_k^{npk}$ tests. A p_k^{npk} division therefore reduces the result to that probability, which we do in implemented computer program.

3.2. Degree of stability.

Doing the calculations with $np_i \to +\infty$ would be very complicated if you had to go to infinity. We call degree of stability modulo p_i, the np_i exponent from which on the respective proportions no longer evolve.

This is illustrated in the program below (where dgst is the degree of stability):

```
{p = 7; dgst = 4; pp = p^dgst;
nb = 200; qc = vector(nb+1,i,0);
for(i = 0, pp-1,
for(j = 0, pp-1,
c = j^2-i^3-5*i; c = c % pp;
d = c+1;
if(d < nb, qc[d] = qc[d]+1)));
default(realprecision, 5);
for(i = 1, nb, print(qc[i]/pp+0.0))
```

Here % performs the modulo operation (on the online application Pari gp). The number of solutions #c is given by qc[c]. As vector indices start at 1 in this computer language, we added 1 to c for quantities recording.

We do our modulo pp operations, which distributes the results in pp boxes. The number of operations is pp^2. To give the results a probability value, we divide them by pp according to the normalisation procedure.

Table 2

p_i	« dgst »	c = 0	c = 1	c = 2	c = 3	c = 4	c = 5	c = 6	c = 7	c = 8	c = 9	c = 10	c = 11	c = 12	c = 13
		$fac_{pi}(c)$													
2	10	0,5	0,5	0,5	1,5	0,5	0,5	1,5	1,5	0,5	0,5	3,5	1,5	0,5	0,5
3	1	1	2	0	1	2	0	1	2	0	1	2	0	1	2
5	2	1,8	1	1	1	1	0,8	1	1	1	1	0,8	1	1	1
7	1	1	1,571	1,143	0,714	1,286	0,857	0,429	1	1,571	1,143	0,714	1,286	0,857	0,429
11	1	1	0,909	0,818	0,727	0,636	1,546	0,455	1,364	1,273	1,182	1,091	1	0,909	0,818
13	1	0,692	0,539	1,077	1,308	1,231	0,846	1,154	1,154	0,846	1,231	1,308	1,077	0,539	0,692
17	2	1,471	1	0,824	0,765	0,882	1,177	0,647	1,294	1,118	1,118	1,294	0,647	1,176	0,882
19	2	1	0,790	1	1,263	0,895	1,421	1,316	1,053	1,105	0,947	1,053	0,895	0,947	0,684
23	2	1	1,304	0,870	0,957	0,826	0,739	0,913	0,739	1,348	1,044	1,087	0,739	1,261	0,913
29	1	1,345	1,069	0,897	0,793	1,035	1,207	1,207	1,241	0,828	0,793	1,276	0,724	0,897	0,862
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Expressions consisting only of monomials have consistent and periodic evolutions in $fac_{pi}(c)$ factors. The assessment of the degree of stability can be done quite naturally even in the absence of a solid theory. For expressions that include polynomials, the behaviour may be erratic for some targets. Here, on the basis of observations, and certainly because of the presence of y^2, the degree of stability seems to be less than 2 in general. We get an exception with p_0 == 2 (as it usually does). We did not observe stabilization until after power 10. Here "dgst" is put in brackets to say that this is a estimated or assumed degree of stability.

From p_{10} = 31 on, we systematically made the calculations with dgst = 2 (although this degree is possibly sometimes only 1) up to p_{45} = 199. Beyond that, the calculations were conducted by assuming dgst = 1. This is done in order to collect a lot of data (up to p_{999} = 7919). It should be noted nevertheless that, in fact, the degree is not always 1. On the other hand, the relative error induced, where it exists, is equal to $1/p_i$ and therefore low (less than 0.5% for p_i > 200), although always the same sign. It does not appear on the same columns and therefore rarely cumulates.

Thus, even if the results are possibly inaccurate, in any case imprecise, the overall errors are certainly small.

3.3. Factors' product.

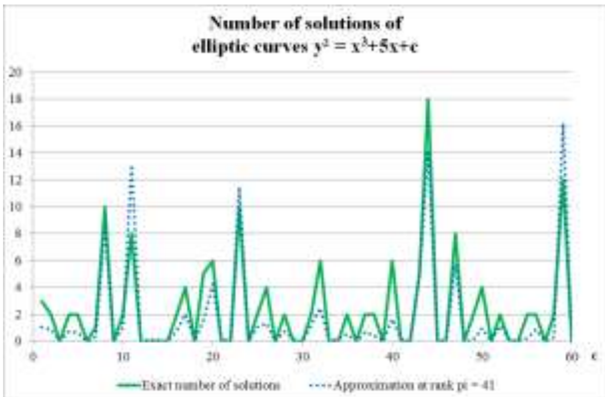
The next step in the algorithm is to make the products $\prod fac_{pi}(c)$ from $p_i = 2$ to p_{imax}. To get the factors, the previous results are multiplied, according to each column, line after line (for example for c = 0, we have 0,5 = 0,5, 0,5 = (0,5.1), 0,9 = (0,5.1.1,8), 0,9 = (0,5.1.1,8.1), 0,9 = (0,5.1.1,8.1.1), 0,623 = (0,5.1.1,8.1.1.0,692), and so on) :

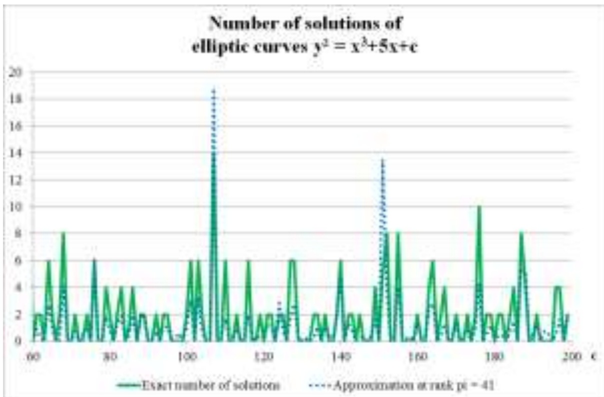
Table 3

p_i	c = 0	c = 1	c = 2	c = 3	c = 4	c = 5	c = 6	c = 7	c = 8	c = 9	c = 10	c = 11	c = 12	c = 13
2	0,5	0,5	0,5	1,5	0,5	0,5	1,5	1,5	0,5	0,5	3,5	1,5	0,5	0,5
3	0,5	1	0	1,5	1	0	1,5	3	0	0,5	7	0	0,5	1
5	0,9	1	0	1,5	1	0	1,5	3	0	0,5	5,6	0	0,5	1
7	0,9	1,571	0	1,072	1,286	0	0,643	3	0	0,572	4	0	0,429	0,429
11	0,9	1,429	0	0,779	0,818	0	0,292	4,091	0	0,675	4,364	0	0,390	0,351
13	0,623	0,769	0	1,019	1,007	0	0,337	4,720	0	0,831	5,706	0	0,210	0,243
17	0,916	0,769	0	0,779	0,889	0	0,218	6,108	0	0,929	7,385	0	0,247	0,214
19	0,916	0,607	0	0,984	0,795	0	0,287	6,429	0	0,880	7,773	0	0,234	0,147
23	0,916	0,792	0	0,942	0,657	0	0,262	4,752	0	0,918	8,449	0	0,295	0,134
29	1,232	0,847	0	0,747	0,679	0	0,316	5,899	0	0,728	10,78	0	0,264	0,115
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
41	1,036	0,860	0	0,709	0,650	0	0,329	8,307	0	1,000	12,82	0	0,154	0,106
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
7919	1,342	1,02	0	1,243	1,283	0	0,197	33,70	0	4,573	9,518	0	0,020	0,032

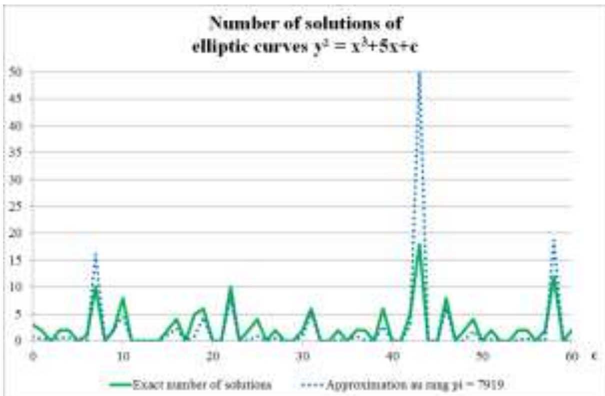
The graphs make it easy to present this data while adding other c-targets. Thus, the comparative graphs between the number of actual solutions and the approximate forecasts from the calculation, for c = 0 to c = 180, line up as follows with an additional uniform factor of 1,024 for calculations conducted up to $p_i = 41$ ($\prod fac_{pi}(c)$ is replaced by 1,024 $\prod fac_{pi}(c)$) and of 0,477 for calculations conducted up to $p_i = 7919$ ($\prod fac_{pi}(c)$ is replaced by 0,477 $\prod fac_{pi}(c)$).

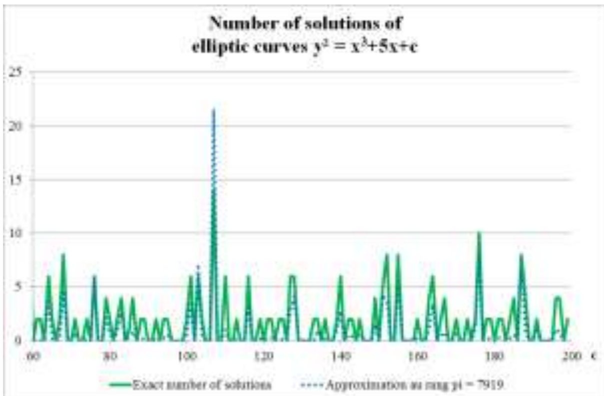
Graphics 1 and 2





Graphics 3 and 4

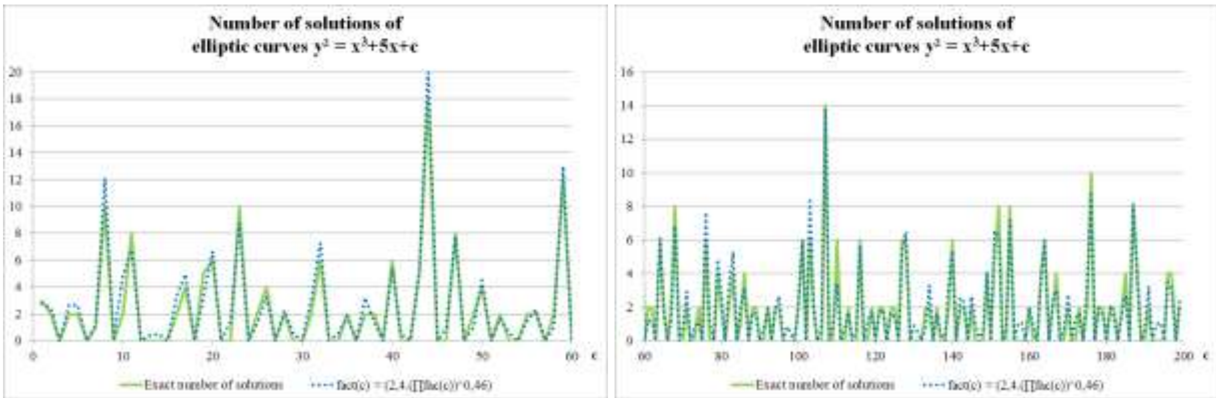




These graphs show that approximation peaks are overestimated when the number of solutions is large and underestimated when the number of solutions is low, with average peaks here deliberately adjusted for intermediate solution numbers. This situation is increasingly marked when the p_i-rank of calculations is pursued further.

When $\prod fac_{pi}(c)$ is replaced by $(2,4).(\prod fac_{pi}(c))^{0,46}$, the comparative graphs for $p_i = 7919$ are the following. Despite some discrepancies, the reader will now recognize a good correlation.

Graphics 5 and 6



Below, we explain how to get the expression $(2,4).(\prod fac_{pi}(c))^{0,46}$.

4. Analysis of the method.

4.1. Accuracy of results.

The accuracy of the number of solutions found here for each of the targets analysed may be questioned. The existence of other solutions cannot be completely ruled out. However, the overlay of the data, highlighted by the graphs, cannot be ignored.

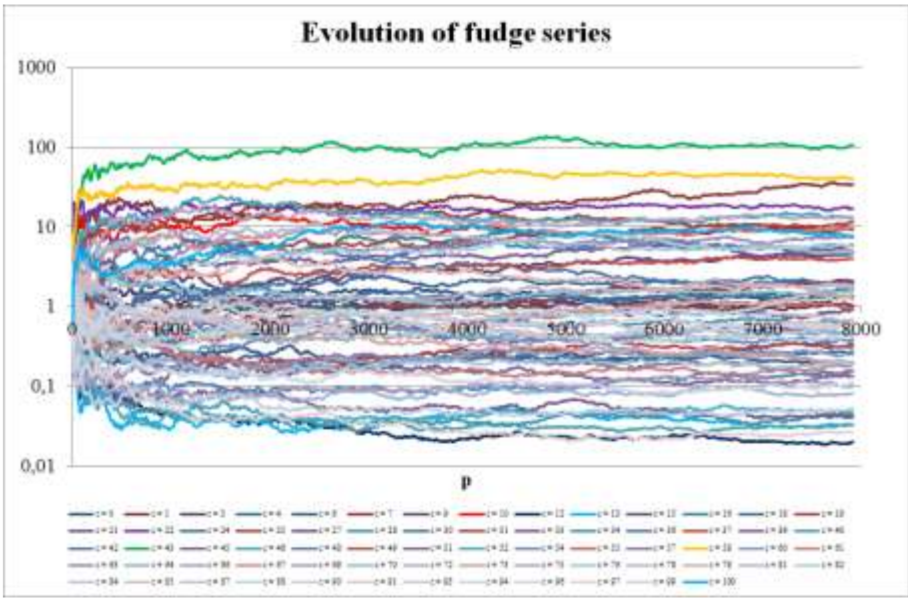
4.2. Discrimination between absence and existence of solutions.

When $fac_{pi}(c) = 0$ for a certain p_i, there is necessarily no solutions for target c. This is a well-known result.
On the other hand, when the corrective factor $\prod fac_{pi}(c)$ is low in the possible range of calculations, there remains an ambiguity, for this factor cannot be evaluated at infinity (especially in the absence of a literal formula as is the case here). However, we see here, for all cases where no solution has been identified, that the factor $\prod fac_{pi}(c)$ is less than 1 (and the opposite if not).
For the proposed algorithm, therefore, there is effectively a discrimination between targets that have solutions and those that do not.

4.3. Limitations.

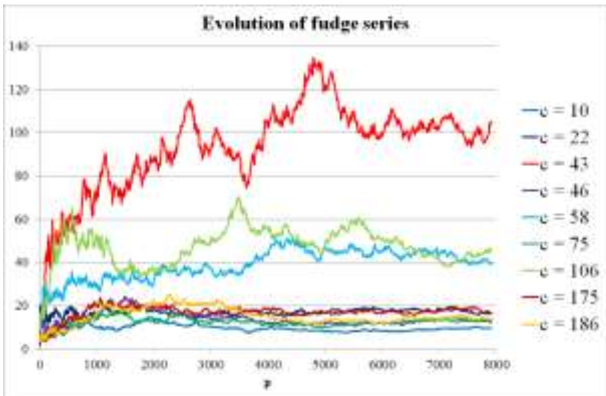
When it comes to determining the number of solutions, the problem may be somewhat more difficult. In our example, there is (in our opinion) a very good correlation. The results are very satisfactory, even though the factors $\prod fac_{pi}(c)$ still undergo significant variations (variations are less pronounced in asymptomatic branch equations case) at the stage where we conducted the calculations.

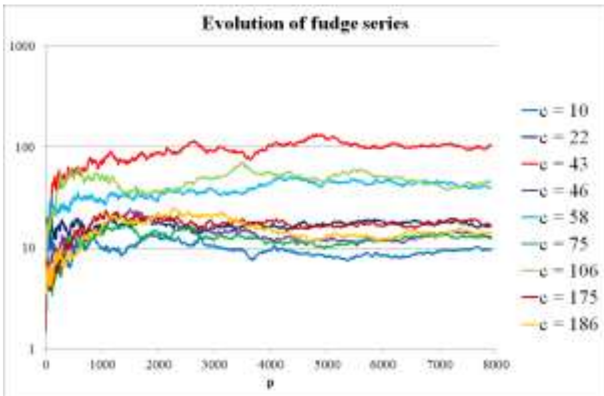
Graphic 7



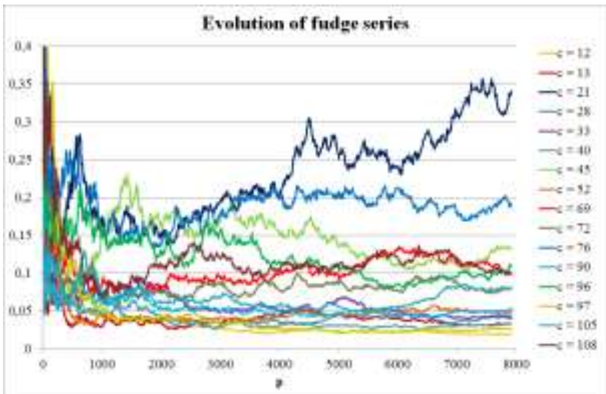
In the previous graph, the corrective factors seem to converge indeed towards stabilized values. But it should be noted here that the y-axis is in logarithmic coordinates which flatten the slopes.
In linear coordinates (below on the left), the wanderings are much more visible.

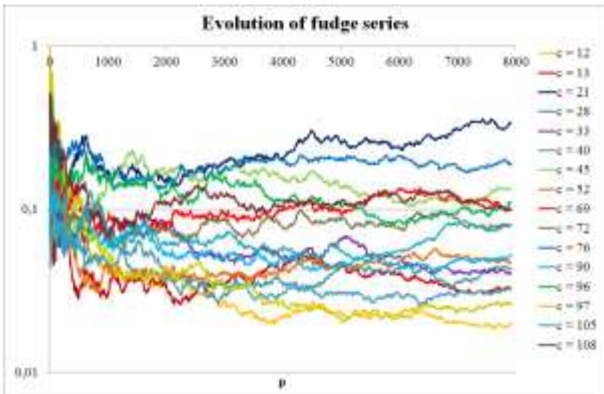
Graphics 8 and 9



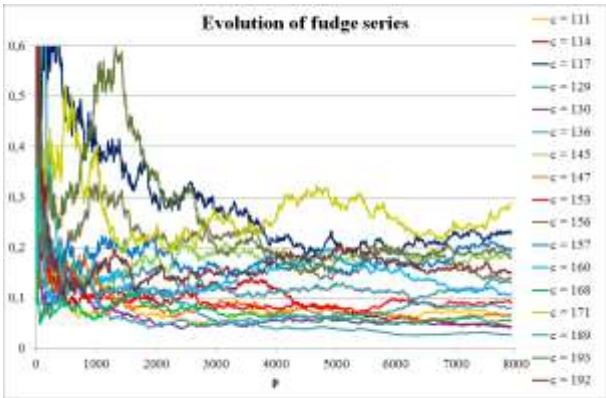


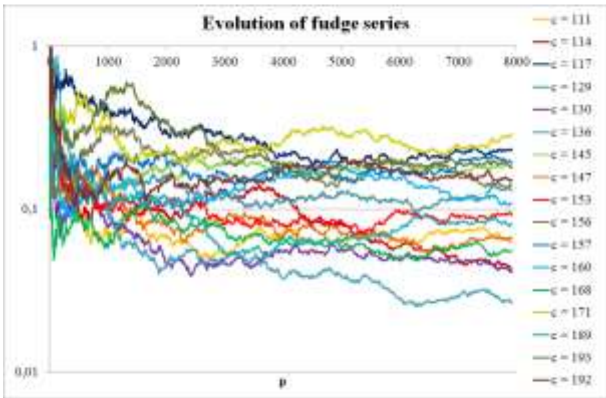
Graphics 10 and 11





Graphics 12 and 13





4.4. Concept of volume.

Generalities

The most important and new point is now being addressed.
Let us first recall the asymptotic formula in the case of the prime numbers of difference 2n, whose equation is p-q = 2n, and whose quantity (seemingly infinite) of solutions is given asymptotically by :

$$\pi_{2n}(x) \approx C_n.x/\ln^2(x) \qquad (5)$$

$$C_2 \approx 2 \prod_{p>3} p.(p-2)/(p-1)^2 \qquad (6)$$

$$C_n \approx C_2 \prod_{q \backslash n} (q-1)/(q-2) \qquad (7)$$

The C_n factor is sometimes called fudge series.

We see here that the number of solutions evolves in the same way, in $x/\ln^2(x)$, asymptotically up to the C_n factor. The term $x/\ln^2(x)$ can be considered as a volume V(x) in which the solutions take position, and depending on whether the value of target c = 2n has a favourable C_n factor, the number of solutions is denser in the said volume, thus giving $\pi_{2n}(x) = C_n(x).V(x)$.

The question now arises for the equation $c = y^2-(x^3+5x)$, an equation that we will write c = Q(y)-P(x). For the fudge series C_n, we have seen how to calculate it. But what is the volume V(x,y,c) to take into account here?

Obviously, it cannot be a function of x^r and y^s, where r and s are positive reals, since there is no divergence of the number of solutions. Similarly, negative r or s are not appropriate, since this would mean fewer and fewer solutions by extending the scope of research, which is absurd.

However, let us still consider P(x) and Q(y) and especially the degree of these polynomials. When x grows asymptotically the value $P(x)/x^{\text{degree}(P)}$ tends to 1, that is, the term of higher degree is predominant over all others. For a given c parameter, the higher the degree of P(x) (respectively of Q(y)), the lower will be the number of solutions and likely within a $z^{1/\text{degree}(P)}$ ratio (and the same for Q(y)). The volume cannot be measured by x or y, so we first define an alternative measure that we note z. This is therefore to be adjusted in the form $z^{1/d}$, where 1/d is intermediate between 1/degree(P) and 1/degree(Q).

The z volume depends on c, or rather, the possibility of having solutions for target c. We know however perfectly well the factor synonymous with this possibility : it is of course $\prod fac_{pi}(c)$. The z parameter is thus linearly connected to $(\prod fac_{pi}(c))^{1/d}$ and the factor $\prod fac_{pi}(c)$ is no longer a multiplier factor of the volume, as in the case of preceding asymptotic branch equation, but is integrated into the volume itself.

Hence the expression of the number of solutions:

$$\{\#c \backslash c = Q(y)-P(x)\} = \text{cte}.(\prod fac_{pi}(c))^{1/d} \text{ et } d \in [1/\text{degree}(P), 1/\text{degree}(Q)] \qquad (8)$$

Another example

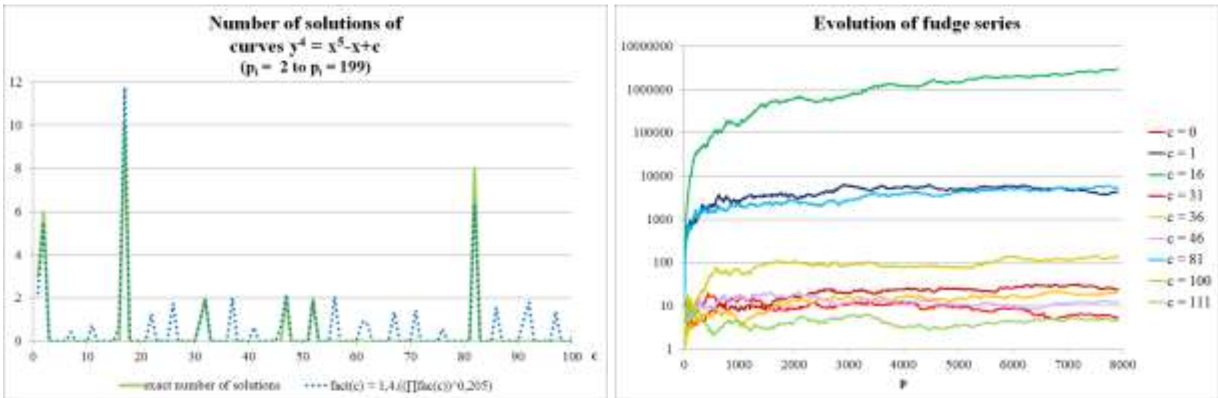
Let us take the example of higher degrees of P(x) and Q(x).
Let us have hence the equation c = Q(y)-P(x) where $P(x) = x^5-x$ and $Q(y) = y^4$.

The list of solutions (c, x, y) for $0 \leq c < 100$ and $-2 \leq x < 10^7$ is given in Appendix 2. To given x again corresponds two solutions that are y and -y. For c = 0, we considered again y = 0 as a single solution.

Here 1/d is between 1/5 and 1/4 and the numeric application shows that peaks of $\prod fac_{pi}(c)$ are growing much faster than previously (see graph on the right below) and the estimate of 1/d is in the order of magnitude of 0.205.

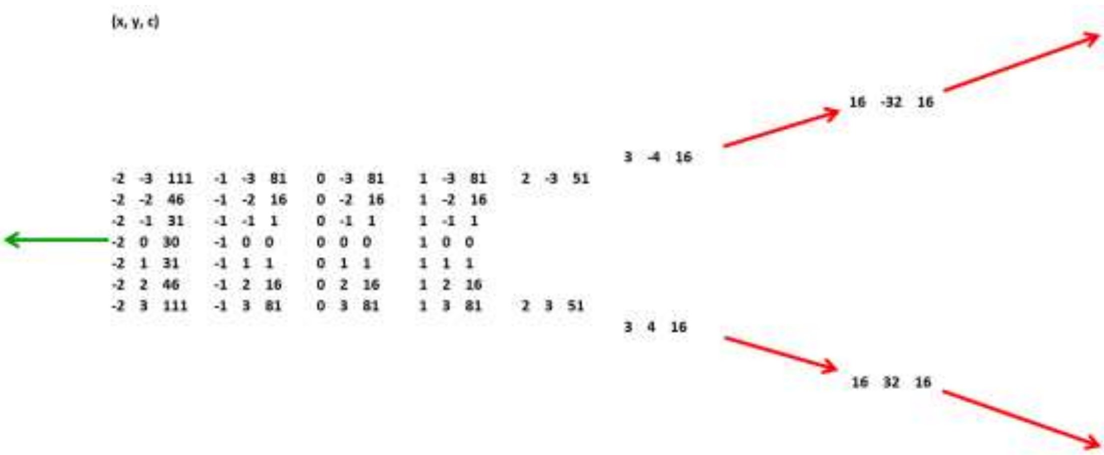
However, at this stage we note many "small" peaks as can be seen in the graph below (on the left side).

Graphics 14 and 15



Two explanations to these small peaks among others can be given.

First, the list of effective solutions may be incomplete and some small peaks correspond to solutions in (x,y) that remain to be found. The picture of the solutions on a spreadsheet, x horizontally and y vertically, is shown below :



On the green arrow side, there can be no other solutions, the account is good. On the red arrow side, the figure is distorted, bringing the two solutions (16, -32) and (16, 32) closer to the rest of the solutions that form a compact set. Beyond, at the right side, there is no way to say whether other solutions exist or not.

Second, the calculations of the fudge series are too imprecise. Indeed, we have made assumptions about the degrees of stability close to the previous ones. After checking that dgst = 4 for $p_0 = 2$, 1 for $p_1 = 3$, 1 for $p_2 = 5$, 1 for $p_3 = 7$ and 3 for $p_4 = 11$, we systematically performed the calculations with dgst = 2 up to $p_{38} = 167$. Beyond that, the calculations were conducted by taking dgst = 1.
However, there is no guarantee of such low levels of stability. On the contrary, these are "expected" in principle with values higher than the degrees of polynomials, or even infinite. Unfortunately, without being totally out of reach for "small" p_i values, more advanced calculations quickly become tedious. In our view, the origin of the observed discrepancies is to be sought in this second explanation. In addition, we note that the convergence to 0 of the peaks $(\prod fac_{pi}(c))^{1/d}$ for the targets involved, if this convergence exists, is extremely slow, as we have not seen any

improvement between stage p_i = 167 and p_i = 7919. For these reasons, we cannot remove the doubt here.

In the case of “high” degrees, using this method to make sure whether or not a solution exists for some c target, is therefore unfortunately not an utmost panacea.

Finally, let us note that if we consider an equation with an infinite number of solutions for some targets and a finite number of solutions for others, the fudge series (after normalization) of the first targets will diverge while the others will not. The method allows thus to manage a mixed situation such as $c = x^6-y^3$ without any particular difficulty in distinguishing target c = 0 from other targets.

4.5. Conclusion.

The correlation between the number of solutions in a diophantine equation and the algorithmic approximation proposed here is good when the degree of the equation is small (max = 3).

For higher degrees, only "numerous" solutions (quantities greater than 1) (or 2 if the second solution is trivial,...) are easily readily discernible due to the presence of small peaks causing ambiguity. This ambiguity is assigned to the none-pertinent task of performing an "infinity" of calculations.

However, even if the algorithm implementation cannot be universal from a practical point of view, the principle of the underlying sieve, in itself, has everything to be an answer to this universality.

Of course, if we also extend the universality to equations containing either an exponentiation, a factorial, a truncation,… the proposed algorithm obviously will not apply (but these are not equations classified as diophantine equations).

REFERENCES

[1] Tenth Hilbert problem: https://fr.wikipedia.org/wiki/Dixi%C3%A8me_probl%C3%A8me_de_Hilbert
[2] http://sites.google.com/site/schaetzelhubertdiophantien/
Diophantine asymptotic equations enumerations. Hypervolumes’ method.
[3] http://sites.google.com/site/schaetzelhubertdiophantien/
Hasse local-global principle and obstructions.

5. Appendix 1.

List of solutions of $y^2-(x^3+5x)-c = 0$ for $0 \leq c < 200$ and $-5 \leq x < 10^{10}$.

c	x	y
0	0	0
0	20	90
1	0	1
3	1	3
4	0	2
6	-1	0
7	-1	1
7	2	5
7	3	7
7	18	77
7	139	1639
9	0	3
10	-1	2
10	1	4
10	6	16
10	9	28
15	-1	3
16	0	4
16	4	10
18	-2	0
18	2	6
18	2446	120972
19	-2	1
19	1	5
19	5	13
22	-2	2
22	-1	4
22	3	8
22	7	20
22	39	244
24	8	24
25	0	5
25	1640	66415
27	-2	3
30	1	6
31	-1	5
31	2	7
31	15	59
34	-2	4
36	0	6
37	4	11
39	3	9
39	10	33
39	23	111
42	-3	0
42	-1	6
42	13	48
43	-3	1
43	-2	5
43	1	7
43	6	17
43	17	71
43	21	97
43	41	263

43	262	4241
43	1726	71707
46	-3	2
46	2	8
46	5	14
46	70	586
48	208	3000
49	0	7
49	16	65
51	-3	3
54	-2	6
55	-1	7
57	48	333
58	-3	4
58	1	8
58	3	10
58	11	38
58	22	104
58	77	676
60	4	12
61	12	43
63	2	9
63	7	21
63	27	141
64	0	8
66	50	354
67	-3	5
67	-2	7
67	9	29
67	149	1819
70	-1	8
73	8	25
75	1	9
75	5	15
75	30	165
78	-3	6
78	6	18
79	3	11
81	0	9
82	-2	8
82	2	10
84	-4	0
85	-4	1
85	4	13
87	-1	9
88	-4	2
91	-3	7
93	-4	3
94	1	10
99	-2	9
100	-4	4
100	0	10
100	102560	32844830
102	3	12
102	14	54
102	19	84
103	2	11
106	-3	8
106	-1	10
106	5	16

106	7	22
106	10	34
106	85	784
106	349	6520
109	-4	5
109	28	149
109	1988	88639
112	4	14
115	1	11
115	6	19
115	29	157
118	-2	10
120	-4	6
121	0	11
123	-3	9
124	8	26
126	2	12
126	9	30
126	25	126
127	-1	11
127	3	13
127	34	199
132	348	6492
133	-4	7
135	11	39
138	1	12
139	-2	11
139	5	17
139	13	49
141	4	15
142	-3	10
144	0	12
148	-4	8
148	12	44
150	-5	0
150	-1	12
150	15	60
151	-5	1
151	2	13
151	7	23
151	1610	64601
154	-5	2
154	3	14
154	6	20
154	410	8302
159	-5	3
162	-2	12
162	18	78
163	-3	11
163	1	13
163	38	235
165	-4	9
166	-5	4
166	110	1154
169	0	13
172	4	16
174	5	18
175	-5	5
175	-1	13
175	10	35

175	74	637
175	2884419	4898775323
177	8	27
178	2	14
180	16	66
181	20	91
183	3	15
184	-4	10
184	8261596	23746298930
186	-5	6
186	-3	12
186	17	72
186	89	840
187	-2	13
187	9	31
190	1	14
195	6	21
195	166	2139
196	0	14
196	32	182
198	7	24
199	-5	7

6. Appendix 2.

List of solutions of $y^4-(x^5-x)-c = 0$ for $0 \leq c < 100$ and $-2 \leq x < 10^7$.

c	x	y
0	-1	0
0	0	0
0	1	0
1	-1	1
1	0	1
1	1	1
16	-1	2
16	0	2
16	1	2
16	3	4
16	16	32
30	-2	0
31	-2	1
46	-2	2
51	2	3
81	-1	3
81	0	3
81	1	3
81	81	243