

A survey of Integer Relations algorithms and rational numbers

Real Numbers
and Computers 6
Schloss Dagstuhl, Germany



Simon Plouffe
LaCim, UQAM
Montréal, Canada

Introduction

LLL algorithms, PSLO algorithm, Integer Relations algorithm.

The computation of rational numbers

Pisot Sequences, rational numbers and the limit of algebraic numbers

The zeolites vs GFUN

Introduction

The comprehension of rational numbers began with the computation of continued fractions.

In short,

If we have a real number x , the approximation of x will be given by the continued fraction expansion of x .

If $x = \text{gamma} = 0.57721566\dots$ then one of the approximation is $71/226$, that is

$$226 \text{ gamma} \approx 71.$$

For us, this is $ax - b = 0$, 0 being small with a, b being integers.

The algorithm:

$$y_n = \left[\frac{1}{x_n} \right] \text{ and } x_{n+1} = \left\{ \frac{1}{x_n} \right\}$$

where the $y(n)$ are the quotients, $\{ \}$ is the fractional part and $[]$ is the floor function.

Actually there is a way to compute the $y(n)$ by using only additions.

From an old *Math. Of Computation* article :

If $0 < x < 1$, pose $x_0 = x$

do $x_k \rightarrow x_k, 2x_k, 3x_k, \dots$ until $nx > 1$.

The quotient is then $n-1$.

$$1 - (n-1)x_k = x_{k+1}$$

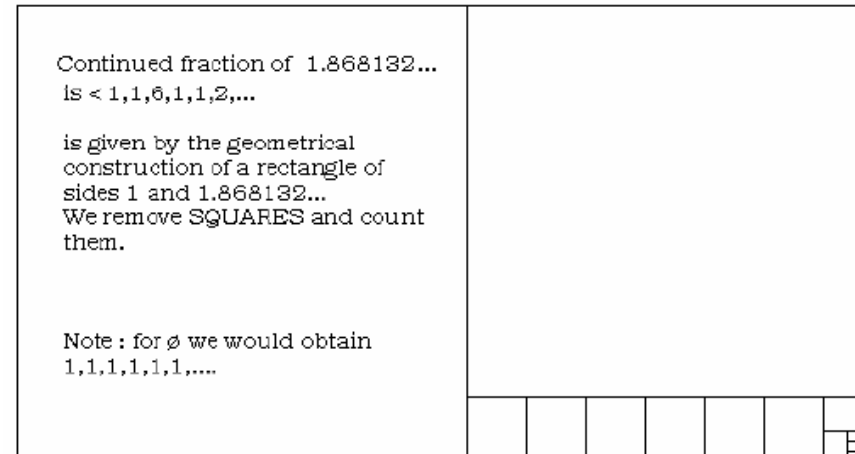
end do.

It is long but no special multiplication algorithm is necessary,
 here $2x_k = x_{k+1} + x_k$, $3x_k = x_{k+1} + x_k + x_k$, ...

To go from 2 to more dimensions took many years and many generations of mathematicians.

Gauss 60 degrees algorithm,
 Hermite, Jacobi, Poincaré, Perron, Brun,
 Ferguson and Forcade 1979,
 Lenstra-Lenstra-Lovasz 1982,
 Lagarias and Odlyzko 1985, ...

The PSLQ, LLL, Integer Relations algorithm are now implemented in most CAS like Maple, Mathematica and Pari-GP now but they are still not exactly the exact or proper generalizations.



For 1 number this is the geometrical interpretation, for 2 numbers...

Well, here is an algorithm :

Take 2 real numbers, a, b

```
do
{ | a - b | } = c
a ← b
b ← c
od
```

{ } is the fractional part.

Will return eventually something like

$$aX + bY + Z \approx 0$$

where X, Y, Z are all integers.

For example with π , $\exp(1)$ after 100 iterations I have (still by using only additions and $>$)...

;-)

$$9257454 e - 5824723 \pi + = 6865462$$

With $0 = .9089493e-9$

It does work but it is NOT the best solution.

This is exactly the point, there is a way to make it to work but to find the simplest solution for the length that's another story.

NOTE : there is a way to generalize with 3, 4, 5 entries and more but again it finds a solution but not the best.

PSLQ LLL and Integer Relations algorithms

There are new algorithms that are widely used since the late 80's but fast enough in the early 1990's.

They are called LLL algorithms, PSLQ algorithms or more generally speaking : Integer Relations algorithms.

We focus on this version of the algorithm :

Given a vector of real numbers

$$[x_1, x_2, x_3, \dots, x_n]$$

the algorithm is able to find integers for which

$$\sum_{i=1}^n a_i x_i = 0 \text{ or near } 0$$

When $n=2$, the problem of finding suitable integers is widely known as the continued fraction algorithm (equivalent to the euclidian algorithm).

If we look at the equation then one can point out that it could be used to reverse the fundamental theorem of algebra.

In which (one of the form of it) every polynomial as a root.

Given a polynomial there is always a way to find 1 root.

In reverse : given an algebraic number X how can we find the simplest or shortest polynomial having X as a root?

We just have to use the algorithm to find it. $[1, x, x^2, x^3, x^4, x^5, \dots, x^n]$

The reasons for the difficulty came from

- 1) The continued fraction algorithm is simple and is a greedy algorithm.
- 2) There are many possible generalizations of the euclidian algorithm that leads nowhere.
- 3) The generalizations of the E.A. are far from being greedy algorithms.

Here is an example :

Let's take $v = (\frac{1}{2} + \sqrt{5}/2)^{48}$

As we know v is a linear combination of $\sqrt{5}$ and 1, in short $v = [\sqrt{5}, 1]$

Or $v = 5374978561 + 2403763488\sqrt{5}$

In terms of PSLQ (Maple 9.5) we ask

With a working precision of 28 decimals

`PSLQ(v, sqrt(5), 1);`

It returns : `[0, 98209, -219602]`

This is false.

If I ask : `PSLQ(1, sqrt(5), v);`

returns

`[-1791659574, -4006272456, 1]`

This is false again and in addition the answer varies with the input being sorted or not???!!

What about Mathematica ?

It hangs.

What about Pari-Gp?

Same answer as Maple.

If I increase the precision to 100 decimals then it finds it properly.

In fact $v = 10749957121.99999999990697636$ this is very close to an integer and the 3 programs were trapped by this local minimum.

There are other known bad examples like that, one of them is the Ramanujan number : $\exp(\pi\sqrt{163}) = 262537412640768744$ almost exactly.

But this is not the worst.

Actually that example reformulated would be that

$$\frac{\log(262537412640768744)}{\sqrt{163}} - \pi = 0$$

Here the numerator is 18 digits long and the approximation to Pi is 31 digits.

The ratio is 31/18, not bad but look at this one :

$$\frac{\log(11614094642430242607991748403072229)}{163} - \log(\phi) = 0$$

In this case the ratio is 2.
The length of the big integer is 35 and
the number of exact digits is 71.

This is a trick : the big integer is the
163'rd Lucas number.

$$\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n = L_n$$

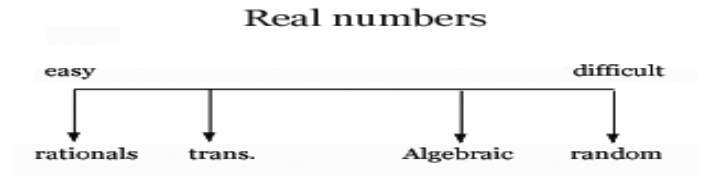
But what is the problem?

By increasing the precision, let's take
the double of the precision of the worst
near integer number then the problem is
solved isn't?

Not so fast.

This is true for most of the classical
examples yes. Like the example I gave
earlier.

But isn't true that most real numbers are
actually transcendental numbers ?



Rationals are very easy to compute.

Some transcendental numbers are very
easy to generate too like the

Champernowne number

0.123456789101112131415...

(that number is a nightmare to expand
into a continued fraction, don't try this
at home).

The Liouville classic example :
 0.1100010000000000000000001000...

Is even worst, even closer to rationals

Or even your favorite version of a Thue-Morse sequence coded into a real number.

The Kolakoski constant.

These numbers can be computed to zillion of digits with an ordinary PC and a big hard disk.

What is the maximum number of decimals humans can compute?

Well, the SETI project made 10^{21} flops. One physician calculated that if all atoms of the entire universe would be had been used since the beginning we would be at 10^{121} only.

For all practical purposes :

All numbers are rational.

Pisot sequences

Consider this recurrence :

$$a_{n+2} = \left[\frac{a_{n+1}^2}{a_n} \right]$$

With $a(1) = 6$ and $a(2) = 31$, let's call it $T(6,31)$ then by using GFUN we can find easily that $T(6,31)$ when expanded into a series in x satisfies :

$$\frac{x+1}{x^{23}+x^5+x^3-x^2-5x+1}$$

Numerator : initial conditions.

Denominator : recurrence.

In other words $T(6,31)$ is recurrent and the degree is 23. This is false because David Boyd found recently that $T(6,31)$ is in fact a recurrence of degree 6852224 at least.

After looking at a few cases, David Boyd conjectured that : $T(a, a^2 - a + 1)$ is recurrent and the recurrence is of the form :

$$a + x - x^2 - x^4 - x^{-e2} - x^{-e3} \dots - x^{-ea} \dots$$

Where $e(k+1) \approx e(k)^2$

In other words, it took 14 days of computation to find that the degree is 6852224, see David Boyd (Un. British Columbia) papers.

Note : I did put $a(0) = 1$ for simplicity, which simplifies the recurrence.

What we know is that most of these recurrences are chaotic and very few are rational polynomials. Note : Mandelbrot fractal is generated with a similar rec.

Since we suspect that $a(n)$ is a linear recurrence then by constructing a power series evaluated at $x \ll 1$ we can use continued fractions to detect anomalies. This is the only tool we have.

In that case : $a(n) \approx (5.15494091)^n$.

Fibonacci sequence or 23'rd degree recurrence can be done simply.

Of course it is trivial to linearize a recurrence like Fibonacci.

First step : The definition is recursive but the computation is not.

Second step : arrays are not needed.

Third step : we can use smart procedures to linearize, see Maple reference with ``option remember``.

Fourth step : a puzzle.

Compute the Fibonacci sequence

(1,1,2,3,5,8,13,21,34,55...)

by using the smallest amount of memory.

Answer : by using additions.

A = 1

B = 1

do loop

 B ← B + A

 A ← B - A

 print B

end loop.

Only 2 memory words.

Other steps are :

Binary power method for simple recurrences like Fibonacci are not too complicated to implement since $F(n)$ has many properties...but not the case with $T(6,31)$.

Zeolites

Zeolites are part of every day life.
They are porous minerals.
They are found in nature and are made
by humans since recently.

(taken from a faq site about them).

<http://www.zeolyst.com/html/faq.html>

*Zeolites are three-dimensional,
microporous, crystalline solids with well-
defined structures that contain
aluminum, silicon, and oxygen in their
regular framework; cations and water
are located in the pores.*

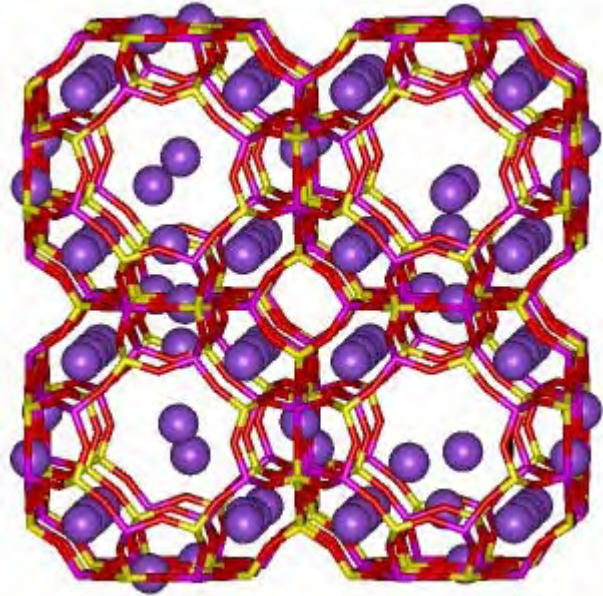
Zeolites means *boiling rock*,
Aluminosilicates minerals.

What is interesting for us is the
crystalline structure and how we can

count the number of atoms in a
structure.

They are very useful materials.

What they do is to count how many atoms there are from a central cell.

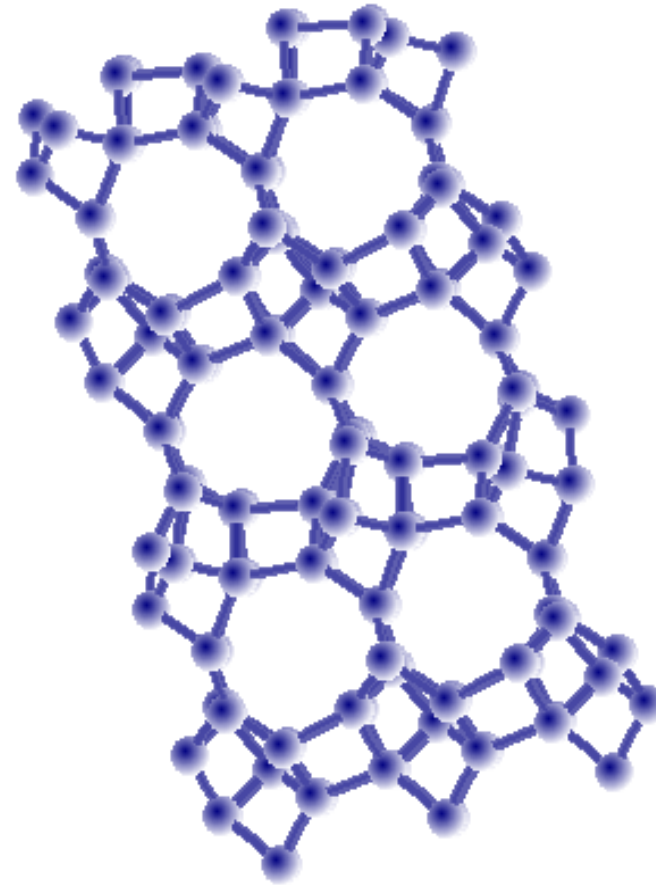
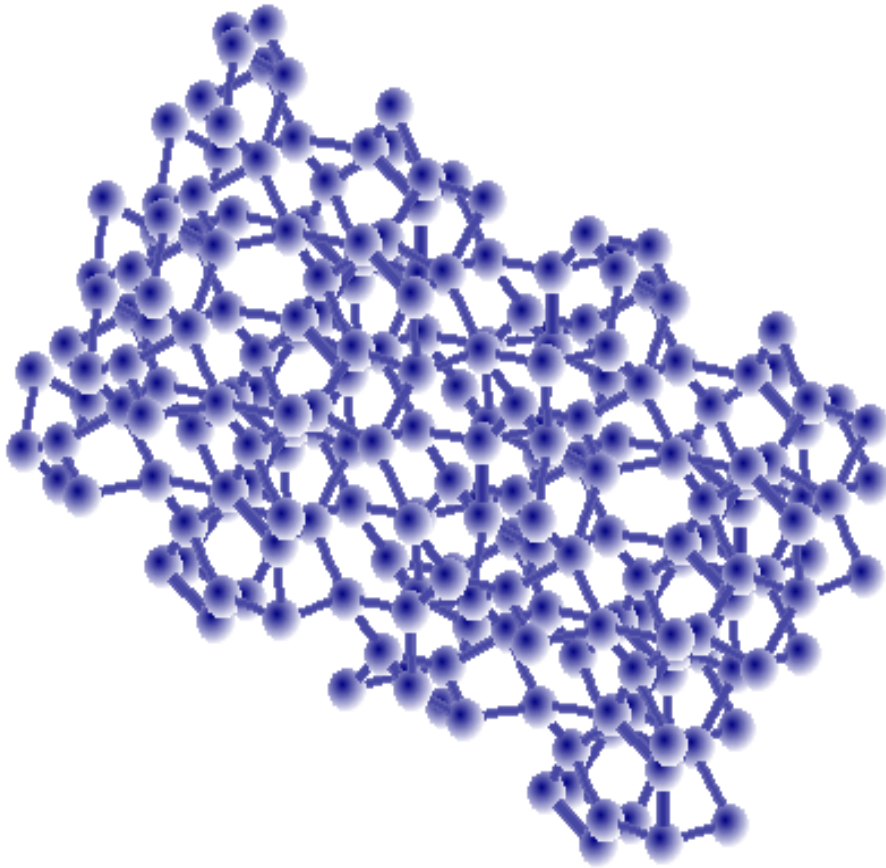


(this is what replaced phosphates in soaps).

This zeolite is : Sodium Zeolite A, used as a water softener in detergent powder. Because it is porous it can be used as a filter, there are many industrial applications of these minerals.

Detergent powders.
Can replace CFC's in aerosols, etc.

Here is another one (more complex than NaCl) :



The same structure viewed at a different angle.

To study the structure we choose a central point and count the number of atoms touching that point : also called the Coordination Sequence (CS).

In this case the count is :

Sequence A008114 of the O.E.I.S.

1,4,10,20,34,58,82,102,136,176,
220,266,306,362,428,484,550,
626,694,778,868,942,1042,1146,
1230,1350,1468,1554,1684,1822,
1926,2072,2214,2322,2494,2654,
2764,2950,3126,3246,3450,3638,
3762,3984,4188,4322,4566,4778, ...

As we know, a regular 3D structure would normally have a simple formula.

For example : the number of points on surface of octahedron is (also called the coordination sequence for cubic lattice).

1,6,18,38,66,102,146,198,
258,326,402,486,578,678,
786,902,1026,1158,1298,
1446,1602,1766,1938,2118,
2306,2502,2706,2918,3138,
3366,3602,3846,4098,4358,
4626,4902,5186, ...

This is simple : by using GFUN (Maple) or GeneratingFunctions (Mathematica),

In a few milli-seconds it finds :
 $a(n) = 4n^2 + 2, n > 0.$

The Generating Function is
 $((1+x)/(1-x))^3.$

Now we expect zeolites to be the same isn't? not quite.

In the case of the octahedron if we use finite difference we can reconstruct the g.f. by using Monmort formula, i.e. : Δ^2

But when the differences, GFUN and even home made specialized programs for cracking a sequence do not work then we must admit that even if the structure is relatively simple :

There are no known formulas for the number of atoms surrounding a central one. This problem is still not well understood.

GFUN is a well known package in Maple that is specialized in manipulating series, sequences and rational polynomials. It was first made by Plouffe-Bergeron, greatly enhanced by Salvy-Zimmermann and is now an essential part of the OEIS

as a standard tool for cracking a sequence.

Now if we go back to a very simple structure : NaCl (common salt).

If we compute the potential at 0 using the rule for coordinates i, j, k then :

- 1) $i, j, k \neq 0$
- 2) The sign being $(-1)^{i+j+k}$
- 3) the charges are $+/- 1/r$

The TOTAL electrostatic potential at the origin due to all charges is hence

$$M_3 = \int_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \frac{(-1)^{(i+j+k)}}{\sqrt{(i^2 + j^2 + k^2)}}$$

$M_3 = -1.74756459463318219063621\dots$

That constant is called the Madelung constant for NaCl and despite many formulas defining it we are still unable to say IF the constant can be expressed with known others like $\log(2)$, $\sqrt{3}$ or π .

Now there are very beautiful formulas for that constant :

$$M = -2\pi + \frac{\Gamma(\frac{1}{8})\Gamma(\frac{3}{8})\sqrt{2}}{\pi^{3/2}} + 2 \sum_{m, n, p=-\infty}^{\infty} \frac{(-1)^{m+n+p}}{\sqrt{m^2+n^2+p^2} [\exp(4\pi\sqrt{m^2+n^2+p^2}) + 1]}$$

$$\sqrt{2} - \pi + 2 \sum_{m, n, p=-\infty}^{\infty} \frac{(-1)^{m+n+p}}{\sqrt{m^2+n^2+p^2} [\exp(2\pi\sqrt{m^2+n^2+p^2}) + 1]}$$

$$-\frac{1}{4} - \frac{\ln 2}{2\pi} - \frac{2\pi}{3} + \frac{1}{\sqrt{2}} - 2 \sum_{m, n, p=-\infty}^{\infty} \frac{(-1)^{m+n+p}}{\sqrt{m^2+n^2+p^2} (\exp(4\pi\sqrt{m^2+n^2+p^2}) - 1)}$$

$$M = -\frac{1}{8} - \frac{\ln 2}{4\pi} - \frac{4\pi}{3} + \frac{1}{2\sqrt{2}} + \frac{\Gamma(\frac{1}{8})\Gamma(\frac{3}{8})}{\pi^{3/2}\sqrt{2}} - 2 \sum_{m, n, p=-\infty}^{\infty} \frac{(-1)^{m+n+p}}{\sqrt{m^2+n^2+p^2} [\exp(8\pi\sqrt{m^2+n^2+p^2}) - 1]}$$

This is very interesting but the question is : What IS Madelung constant???

Because we know that for example gamma = potential at 0 of charges placed at 1,2,3,4,....

At least 4 serious attempts were made using PSLQ, LLL and the like programs and so far even with more than 1000 digits known : nothing was found.

Nature is more seriously complex than we thought maybe...?

What can we do then?

About Pisot sequences, only small recurrences and rational polynomials can be found using GFUN, up to ~ 1000 terms.

Recurrences can be detected with ~ 100 terms.

Ordinary high-precision computations for other detections.

Zeolites : GFUN is well suited for simple cases since only finite differences are needed. But maybe Grobner basis for more?

Madelung constant : More PSLQ-LLL like programs, massive attacks.

Clues : terms with $1/(\exp(x) - 1)$, Epstein series, mixed constants maybe?

PSLQ-LLL limits.

-Degree of algebraic can be < 120 but no more.

A recent finding used 10000 digits and a special version of MPFUN of Bailey with parallel computation in order to achieve results.

- Some weaknesses when near an integer!

- Input sorted or not produce different outputs.

- Transcendental numbers could be very close to integers or rationals.

References

- 1) A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász, *Factoring Polynomials with Rational Coefficients*, Math. Ann. 261 (1982).
- 2) Web reference :
<http://www.dice.ucl.ac.be/~fkoeune/LLL.html>
- 3) H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1995 (Second edition).
- 4) H. R. P. Ferguson and R. W. Forcade, Generalization of the Euclidean Algorithm for Real Numbers to All Dimensions Higher Than Two, *Bulletin of the American Mathematical Society*, 1 (1979), p. 912 - 914.
- 5) J. Hastad, B. Just, J. C. Lagarias and C. P. Schnorr, "Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers," *SIAM Journal on Computing*, vol. 18 (1988), p. 859 - 881.

Terms to be searched on the internet:

Algebraically Independent
 Transcendental Number
 Integer Relation
 LLL algorithm
 PSLQ algorithm , HJLS, PSOS
 Lattice Reduction
 Gauss 60 degree algorithm
 Ferguson-Forcade Algorithm
 Lenstra-Lenstra-Lovasz
 Bailey-Borwein-Plouffe algorithm
 Plouffe algorithm for Pi in decimal and binary
 Schanuel's Conjecture
 Pisot sequences, David Boyd
 Zeolites

Computer Algebra Systems

-Maple : IntegerRelations, GFUN package

-Mathematica : see the GeneratingFunctions package, Recognize, Mallinger

-Pari-GP : lindep, algdep.