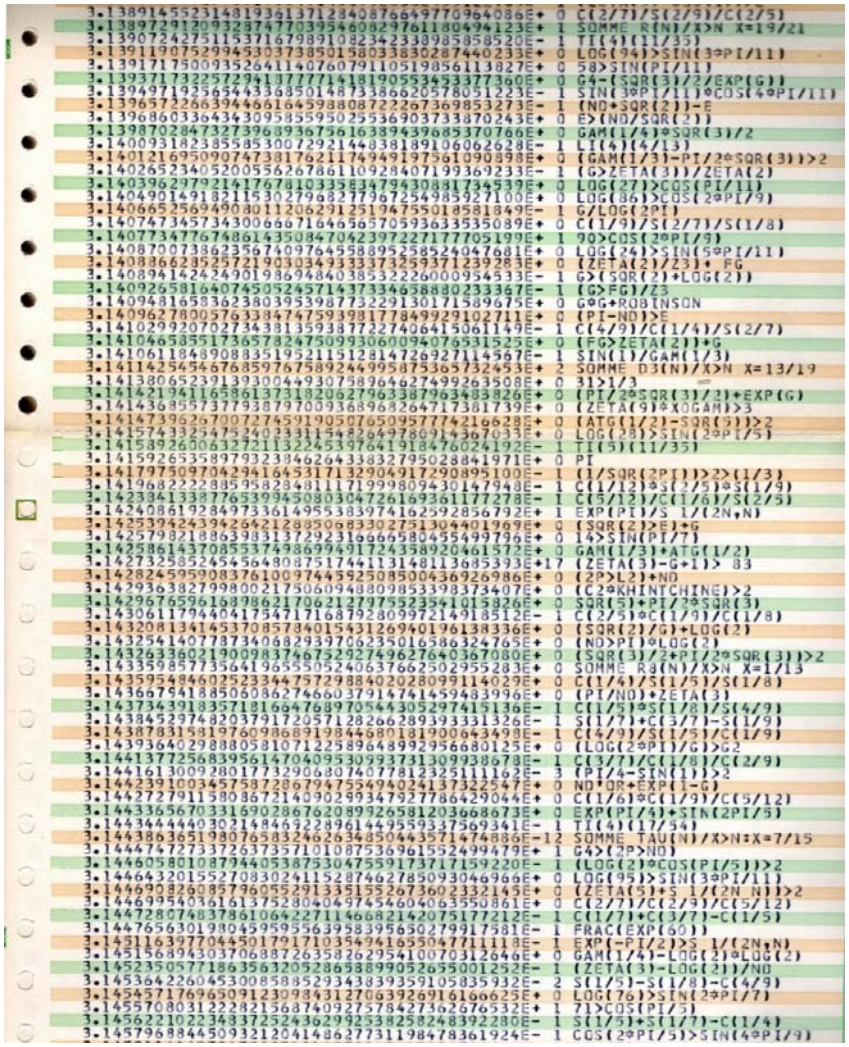


Computer output from 1988 on pyjama paper

Big Data in Pure Mathematics 2022

Simon Plouffe

The Inverter
from 1986 to 2022

By Simon Plouffe

- The talk will give details on how the Inverter works, how it began and the present situation. Some details of the algorithms used will be explained.
- Some results will be presented from which some are known.

Synopsis

The inverter
Data from the OEIS
Programs
Size, portability and
cpu time.

The inverter

a bit of history

It began in April 1986, I was collecting mathematical constants on 5.25 inches diskette on my Apple IIc. Using Appleworks, then later I used Hypercard for a while on my Mac Plus, eventually bought a hard disk (very expensive) of 60 megabytes to fill it in half with numbers.

The inverter

a bit of history

This is what it looked
like at the time on Hypercard

The inverter

a bit of history

Then Hypercard became too small for any storage of big tables.
At the time there were 2 parts : The Integers and the real constants.
It took a while but I managed to make one big database that contains the 2 worlds.

Then 2 things happened in 1990: an access to a Unix system at Uqam (Université du Québec à Montréal) and my collaboration with Neil Sloane of the Bell Labs.

This is where I decided it was time I do a master in mathematics.

The inverter

a bit of history

Most of my time was used for the EIS (Encyclopedia of Integer Sequences), the book was finally published in 1995 and soon after that the publisher agreed that an online version was not a threat to possible sales in libraries. The success of the online EIS (now OEIS) is quite impressive : There are 8000 contributors, 10033 citations and 19427 references in Wikipedia (english and french).
As of May 16 2022 there are 353700 sequences in the database, 10000 to 15000 are added each year. In this catalog there are 10821 mathematical constants already.

The inverter

a bit of history

In 1993, I began to collect large amount of mathematical constants, first programs where written. This is where I began to use the LLL (Integer relations algorithm). They were collected from the M.T.A.C tables of Mathematics of Computation micro-fiche.

The version I had was the one on the PARI-GP program called <lindep>. Very efficient and rapid. The big advantage was that I could interface the routine with Maple.

This is easy to do : you make a unix script that takes an input from Maple, sends a command to pari-gp that return the answer properly formatted.

The inverter

a bit of history

The integer relations algorithm has many versions.

In short it can solve the exact inverse problem of having a polynomial and finding a root. From a real constant (up to a certain precision) it can find the polynomial from which it comes from. The precision needed is roughly :

$$\sqrt{Number\ of\ decimal\ digits} = degree\ of\ the\ polynomial$$

In practice, up to 1000 digits at the time could be used, today the maximum number of digits is ~5000.

The inverter

a bit of history

You can read about it on Wikipedia :
https://en.wikipedia.org/wiki/Integer_relation_algorithm

Here is the definition :
An **integer relation** between a set of real numbers x_1, x_2, ..., x_n is a set of integers a_1, a_2, ..., a_n, not all 0 such that

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0.$$

The inverter

a bit of history

An **integer relation algorithm** is an algorithm for finding integer relations. Specifically, given a set of real numbers known to a given precision, an integer relation algorithm will either find an integer relation between them, or will determine that no integer relation exists with coefficients whose magnitudes are less than a certain upper bound.

The inverter

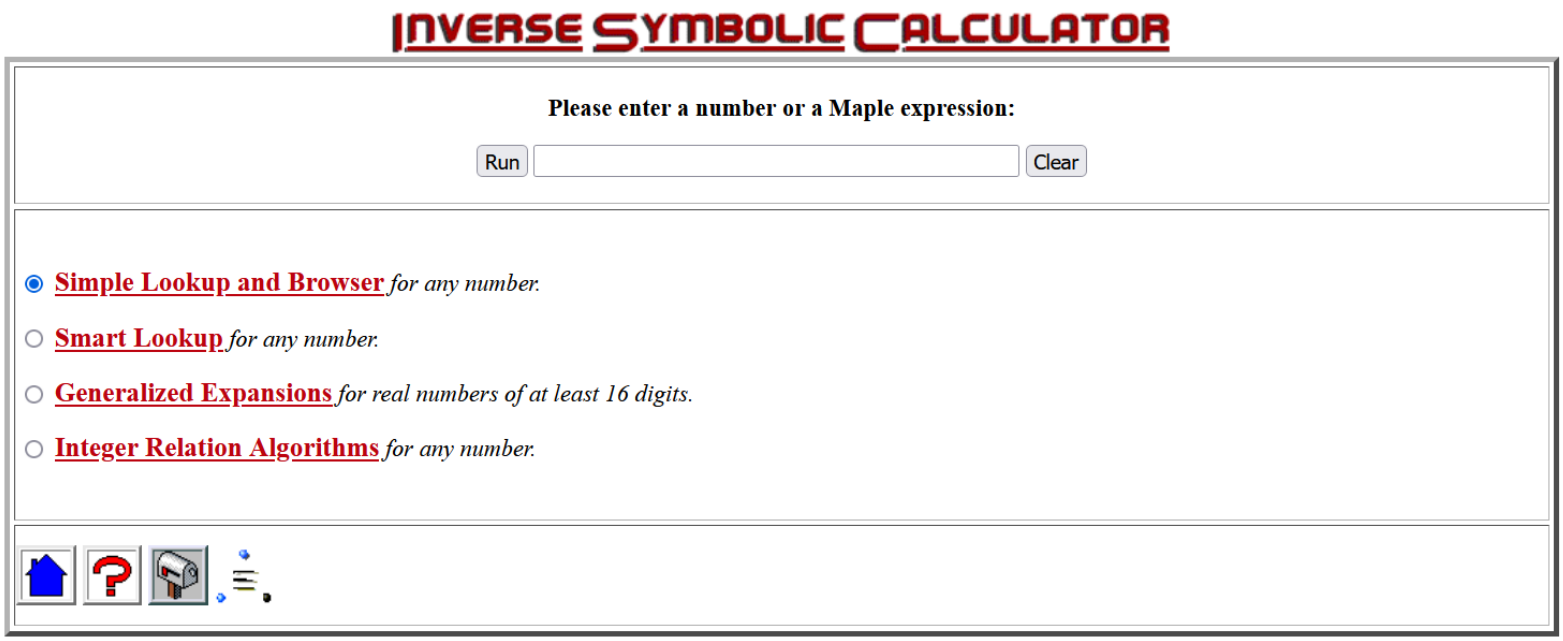
a bit of history

This is what found my formula for π in 1995.

$$\pi = \sum_{k=0}^{\infty} \left[\frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right]$$

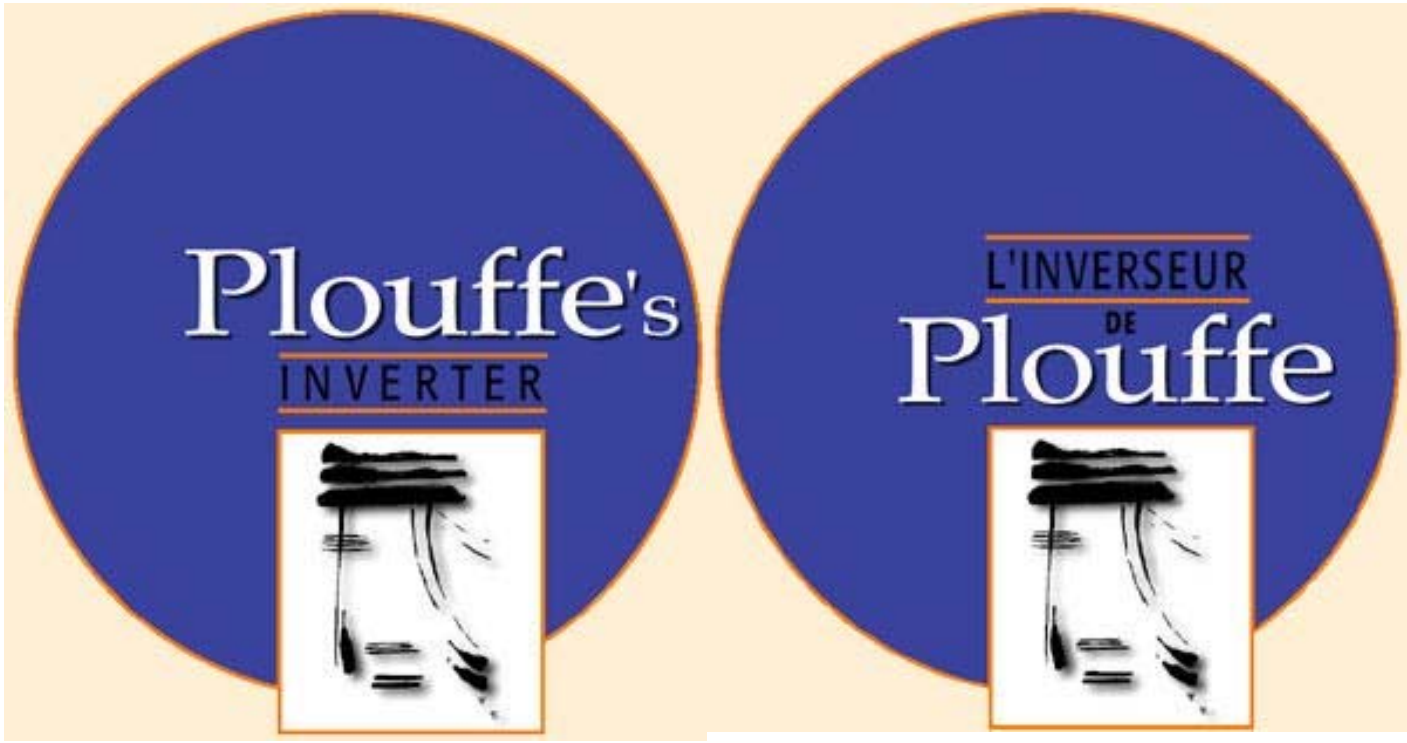
A main ingredient of the Inverse Symbolic Calculator that opened on July 18, 1995.

That same ISC is still in operation here :
http://wayback.cecm.sfu.ca/projects/ISC/ISCmain.html
it contains 54 million constants.



The ISC opened in Vancouver on July 18, 1995 and is still in operation after almost 27 years. !

The Inverter in 1998 (up to 2011) at UQAM



Design by Renée Othot, Pi logo by Yves Chiricota (with permission)

The Structure of the Inverter

There is no structure, no index. The Inverter is a set of 9000 tables from 1000 to 9999 with no decimal point. The real numbers are naturally ordered.

Here is a sample output from the old version (algebraic entries).

```
1.000000000000000000000000033655803748315958521499648026848359612e0 a001 2504730781961/10610209857723*(1/2+1/2*5^(1/2))^3
1.000000607323283127668981735654042398813845595946023834829615844e1 a002 1/7*(13+7^(2/3)*12^(3/4))*7^(1/3)
1.000000001055626531498019979434444231768462746634735819594218467e-1 a003 2*cos(1/18*Pi)-cos(2/9*Pi)-cos(11/24*Pi)-cos(2/27*Pi)
1.000000000000000000000000000000000000000203259905113355101968490698 25e0 a004 (1/2+sqrt(5)/2)^83/Fibonacci(1)/Lucas(83)
1.000002614946700871297147550944904662028044167410212290084342459e23 a005 (1/sin(22/191*Pi))^51
1.000015645779046667729154723415315713424456112660258802343846351e18 a006 ((1/2+1/2*5^(1/2))^87+(-
1/2*5^(1/2)+1/2)^87)*fibonacci(218/19)/Lucas(9)/sqrt(5)
1.000000835096713961566373109143293676943655683702340995442991156e-1 a007 Real Root Of -369*x^4+373*x^3-989*x^2+987*x+109
1.000009259603800094168490831478943636021186572276495810122022991e-1 a008 Real Root of x^4+20*x^2-112*x+11
2.838541822761633051368050605171122071267889746762771794459861652e0 a009 9/(13^(1/2)+12^(3/4))^(1/2)
1.000000154953758625804884675278917791176598916601460893578357498e1 a010 (1/5*cos(37/120*Pi)-5/6)/(5/6*cos(37/120*Pi)-2/5)
1.000001232234080033322997540518611368672051604867643646689475870e-1 a011 (1/12*(2-2^(1/2))^(1/2)-5/6)/(2*(2+2^(1/2))^(1/2)+4)
1.000001183974399355846415232612483030241341666961769435445322768e0 a012 (4/5*cos(19/60*Pi)-5/6)/(5/6*cos(13/60*Pi)-1/4)
1.000000766783997513832641896050782067210834339837489422329400756e-1 a013 (2/3*tan(13/60*Pi)-1/3)/(4/5*tan(19/60*Pi)+5/6)
1.000015528100075709292697479256740555241377390539850739262476991 64e2 a014 361/2-36*5^(1/2)
1.005051880700988039258313965880795892267481304863310870886894848e0 a015 Real Root of 1+x^17-x^16-x^15+x^14-x^11+x^10-x^8-x^7-x^6+x^5-
x^4-x^2-x
1.000000004502970722266085525705148215308064375023602710905401526e2 a016 1/7*(74+346430*2^(1/2))^(1/2)
1.000000001200185568869925301084700029362740547733475671366117958e0 a017 81/59-3/17*89^(1/6)
1.000000005602684438703446331733540079856595861745317342076722784e1 a018 -86/45+65/29*86^(3/8)
1.000000019143670052409176795932737219533259582324571750389280152e8 a019 181160923^(31/32)
1.000000333166472541693195159388481580225590067365220582451174030e1 a020 Real Root of x^6+x^3+1 = 1001003
```

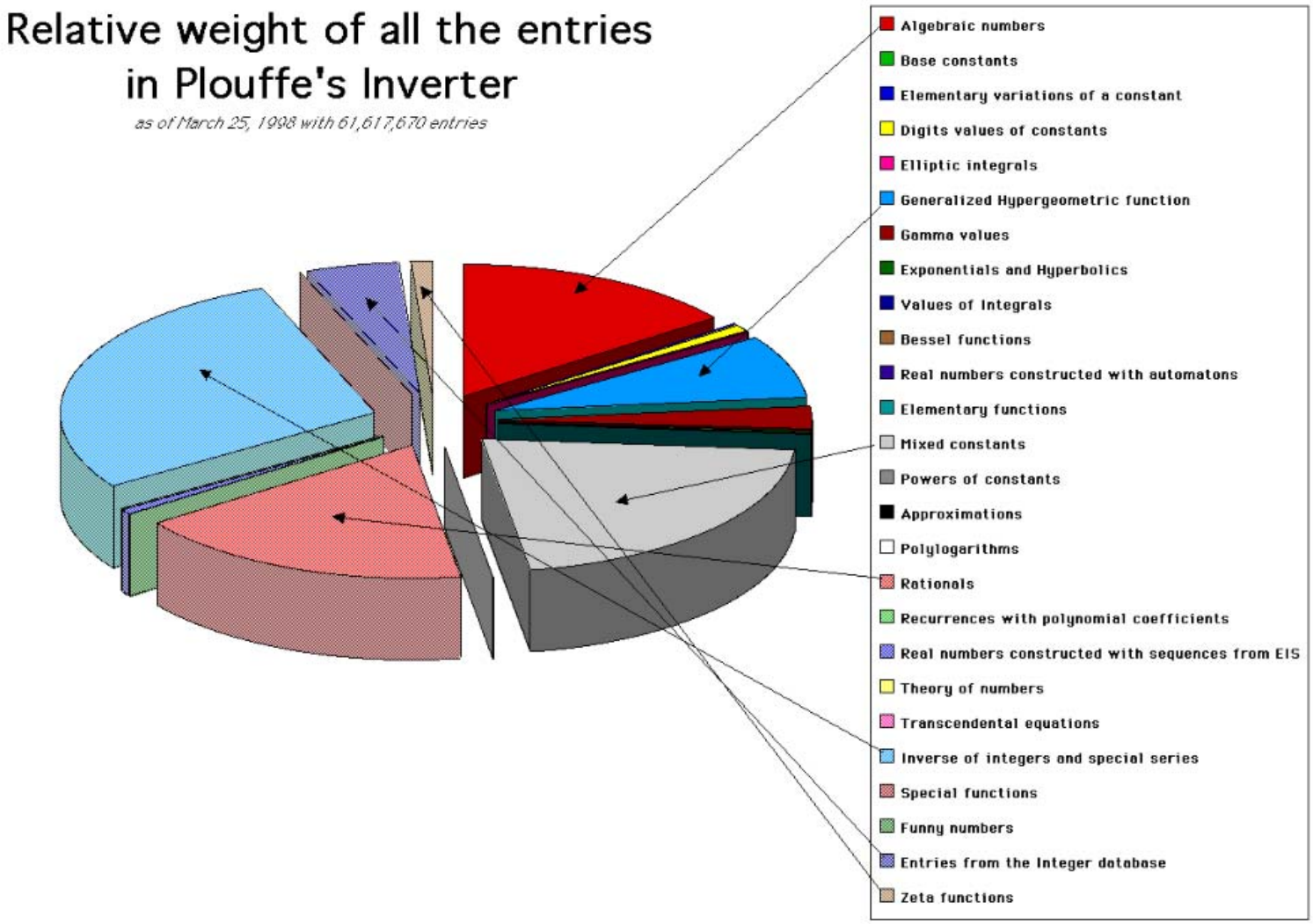
The Structure of the Inverter

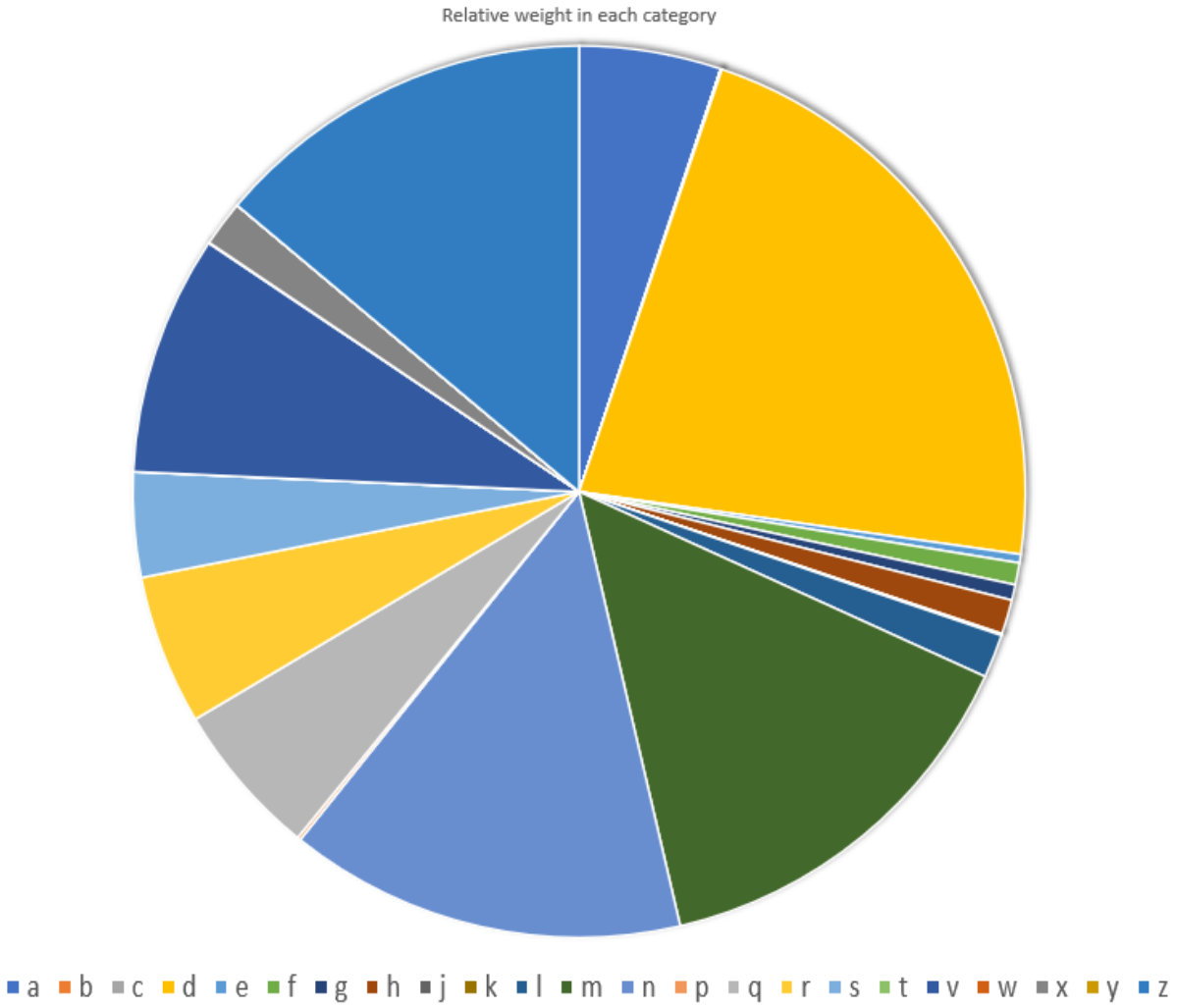
There are 9000 tables (final version) and 571 different tables (source files).

In size :
86464978693 mathematical constants
7669 gigabytes or 7.669 TB
Average entry : about 90 characters

From the OEIS database (2 categories)
1850219199 entries from sequences to real
approx. 3.64779582 billion entries (big integers)
Total : 5.5 billion entries from the OEIS.

As of 1998 on Plouffe's inverter





As of today (May 2022)

a	Algebraic
b	Base constants
c	Constructed reals
d	Digits of reals
e	Exponential
f	Hypergeometric
g	Gamma functions
h	Hyperbolic
j	Bessel, special numbers
k	Constructed reals
l	Logs
m	Mixed constants
i	Integers
p	Polylogs
q	Rationals
r	Recurrences
s	Real to sequences
t	Number theory functions
v	Series
w	Transcendental eq.
x	Integers
y	Big integers
z	Zeta function

The inverter

the engine

The engine is simply the ‘look’ command.
It is part of Unix since a long time. It can search a sorted file very rapidly by using a dynamic binary search.

On a Windows/cygwin64 installation the limit is size for a file is more than 100 gigabytes. In plain English it can find the occurrence of a string of characters from any sorted text file in a fraction of a second. This is why there is no need for an index for my 9000 files.

The inverter

the engine

The program that drives all this is a Maple program of about 8000 lines, there is an interface for Pari-GP and Mathematica as well as ‘calls’ to the Unix system for the use of the ‘look’ command.

There are 5 parts.
1- pismart() : it tries 470 elementary combinations from K.
2- pidev() : Uses generalized expansion for a real K.
3- pilll() : Uses the LLL algorithm of Pari-GP.
4- pialg() : Test if the constant K is algebraic of small degree.
5- pil() : The simple lookup (with a parameter for the length).

The inverter

How many constants

A parenthesis about the pismart program.

The reach of possible combinations with all the variants on 1 number is 1.5×10^{15}. That test takes several minutes.

The inverter

the engine

The depth of search depends on the number of digits available. For a complete test, I need 41 digits, most of the real constants are with a standard 41 digits.
I choose 41 digits for several reasons.
1- It is the precision used in Knuth books (TAOCP) for constants.
2- It is the needed precision to detect exotic numbers like

$$e^{\pi\sqrt{163}} = 0.26253741264076874399999999999992500725972\ldots$$

$$\sum_{n=1}^{\infty} \frac{n^3}{e^{\pi n 2/13} - 1} = 119.00000000000000000000000000009593745\ldots$$

The inverter

The decimal point

My first Inverter was made of separate files with 41 digits and exponent. In 1995 I made a strange decision to cut the digits at 16 with no decimal point for practical reasons. For lookups I just need the string of digits.

Later in 2004, I began to extend (from 200 million to approx. 1 billion constants) with a format of 64 digits (with the exponent).

In 2016 I came back with 41 digits and no exponent to finally arrive at a variable length entry with digits from 11 to 100+ digits. I still have the original files separately but the main table now is definitively with that format.

The inverter

the engine

The idea behind was to be able to include integers, big integers. There are many reasons for that.

Look at n!, what is the use of having large values of n! (truncated to let's say 64 digits) when n is large ?

- Because these entries are approximate values of a truncated Stirling expression for example.
- The other reason is to include the new values that are reverse of constants.

The inverter

reverse of constants ?

Question : what is the limit point of 5^n if we read digits from the right ? There are 2 accumulation points.
$5213023304311311\ldots$ and $526567957879699\ldots$

The first acc. point contains only 0,1,2,3,4 (apart from 5).
The second acc. point contains only 5,6,7,8,9 (apart from 2).
Why ? (I do not know the answer). See sequence : A158625.
a(n) = [5, 2, 1, 3, 0, 2, 3, 3, 0, 4, 3, 1, 1, 3, 1, 1, 2, 4, 2, 1, 0, 3, 1 ,…]

What about 2^n or n! ?
The reverse lower limit value of n! or 2^n
is : 2, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1 , …
only zeros and ones (apart from th initial 2). (see sequence : A145679
Is also the upper limit value of n! when we read backward.
8, 8, 8, 9, 8, 9, 9, 8, 9, 8, 9, 8, 8, 9, 9, 8, 9, 8, 9 = A023415
Note : the sum of the 2 sequences is : 10,9,9,9,9,9,9,9,9,9, …
and I still don't know why. The sum of the 2 sequences is 11.

The Inverter

Details of programs

The pismart() program only does 470 variations with 1 constant k like

$$\frac{k}{2}, \sqrt{k}, \frac{2k}{3}, \sin(k), \arctan(k), \exp(k), \exp(\pi k), etc.$$

Of course when possible.

The pismart2() program makes 15 variations of the first list, for a total of 7050 variations. That one is reserved for <tough cookies> .

The Inverter

Details of programs

The pilll() program tries to find a simple combination of known constants, there are 126 tests.

first test (table 001).

$$\pi, e, \gamma, \frac{1}{\pi}, \frac{1}{\gamma}, 1, \frac{1}{2}+\frac{\sqrt{5}}{2}, \sqrt{2}, \sqrt{3}, \frac{1}{\mathrm{Ei}(1)}, \mathrm{Ei}(1), W(1), \frac{1}{Catalan},$$

$$Catalan, \zeta(3), \frac{\pi^2}{6}, \frac{\pi^4}{90}, \zeta(5), \ln(2), \ln(3), e^{-1}, \ln(5), \frac{1}{\ln(5)},$$

$$\frac{1}{\arctan\left(\frac{1}{2}\right)}, \arctan\left(\frac{1}{2}\right), \frac{1}{\ln(2)}, \sqrt{\pi}, \frac{2\pi\sqrt{3}}{3\,\Gamma\left(\frac{2}{3}\right)}, \frac{\pi\sqrt{2}}{\Gamma\left(\frac{3}{4}\right)}, \ln(\pi),$$

$$e^{\pi}, \sin(1), \cos(1)$$

The Inverter

Details of programs

The pilll() program tries to find a simple combination of known constants, there are 126 tests.

3rd test (table 003).

$$\frac{\pi^2}{2}, \Psi\left(1, \frac{1}{3}\right), \pi^2 + 8\,Catalan, \Psi\left(1, \frac{1}{5}\right), \Psi\left(1, \frac{2}{5}\right), \Psi\left(1, \frac{3}{5}\right), \Psi\left(1, \frac{1}{7}\right),$$
$$\Psi\left(1, \frac{2}{7}\right), \Psi\left(1, \frac{3}{7}\right), \Psi\left(1, \frac{4}{7}\right), \Psi\left(1, \frac{5}{7}\right), \Psi\left(1, \frac{1}{8}\right), \Psi\left(1, \frac{3}{8}\right), \Psi\left(1,$$
$$\frac{1}{9}\right), \Psi\left(1, \frac{2}{9}\right), \Psi\left(1, \frac{4}{9}\right), \Psi\left(1, \frac{5}{9}\right), \Psi\left(1, \frac{1}{11}\right), \Psi\left(1, \frac{2}{11}\right), \Psi\left(1, \frac{3}{11}\right),$$
$$\Psi\left(1, \frac{4}{11}\right), \Psi\left(1, \frac{5}{11}\right), \Psi\left(1, \frac{6}{11}\right), \Psi\left(1, \frac{7}{11}\right), \Psi\left(1, \frac{8}{11}\right), \Psi\left(1, \frac{9}{11}\right),$$
$$\Psi\left(1, \frac{1}{12}\right)$$

The Inverter
Details of programs

This is the 126'th table.

$$\frac{\pi}{2}, 1, \frac{\pi\sqrt{2}}{2}, \pi\cos\left(\frac{\pi}{5}\right), \frac{\pi\sqrt{3}}{2}, \pi\cos\left(\frac{\pi}{7}\right), \pi\cos\left(\frac{2\pi}{7}\right), \pi\cos\left(\frac{\pi}{8}\right),$$
$$\pi\cos\left(\frac{3\pi}{8}\right), \pi\cos\left(\frac{\pi}{9}\right), \pi\cos\left(\frac{2\pi}{9}\right), \pi\cos\left(\frac{\pi}{10}\right), \pi\cos\left(\frac{3\pi}{10}\right),$$
$$\pi\cos\left(\frac{\pi}{11}\right), \pi\cos\left(\frac{2\pi}{11}\right), \pi\cos\left(\frac{3\pi}{11}\right), \pi\cos\left(\frac{4\pi}{11}\right), \pi\cos\left(\frac{\pi}{12}\right),$$
$$\pi\cos\left(\frac{\pi}{13}\right), \pi\cos\left(\frac{2\pi}{13}\right), \pi\cos\left(\frac{3\pi}{13}\right), \pi\cos\left(\frac{4\pi}{13}\right), \pi\cos\left(\frac{5\pi}{13}\right),$$
$$\pi\cos\left(\frac{\pi}{14}\right), \pi\cos\left(\frac{3\pi}{14}\right), \pi\cos\left(\frac{5\pi}{14}\right), \pi\cos\left(\frac{\pi}{15}\right), \pi\cos\left(\frac{2\pi}{15}\right),$$
$$\pi\cos\left(\frac{\pi}{16}\right), \pi\cos\left(\frac{3\pi}{16}\right), \pi\cos\left(\frac{5\pi}{16}\right)$$

The Inverter
Details of programs

The pidev program : it expands a real constant in all known bases and algorithms. There are more than 1000 variants. Here is a few ones.

Continued fraction: $y_n = \left[\frac{1}{x_n}\right]$, $x_{n+1} = \left\{\frac{1}{x_{n+1}}\right\}$ will give

$$x = a_{0+}\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{a_{3+\cdots}}}}$$

Egyptian fraction: $y_n = \left[\frac{1}{x_n}\right]$, $x_{n+1} = x_n - \left\{\frac{1}{y_n}\right\}$ will give

$$x = \frac{1}{a_0}+\frac{1}{a_1}+\frac{1}{a_2}+\frac{1}{a_3}+\frac{1}{a_4}\ldots$$

Egyptian product or Engel expansion: $y_n = \left[\frac{1}{x_n}\right]$, , $x_{n+1} = \{x_n y_n\}$ will give

$$x = \frac{1}{a_0}+\frac{1}{a_0a_1}+\frac{1}{a_0a_1a_2}+\frac{1}{a_0a_1a_2a_3}+\frac{1}{a_0a_1a_2a_3a_4}\ldots$$

The inverter
Details of programs

Base k (integer) algorithm: $y_n = [kx_n]$, $x_{n+1} = \{kx_n\}$

Infinite product ($x > 1$): $y_n = 1 + \left[\frac{1}{x_n - 1}\right]$, $x_{n+1} = x_n \frac{y_n}{y_n + 1}$ will give

$$x = \left(1 + \frac{1}{a_0}\right) \cdot \left(1 + \frac{1}{a_1}\right) \cdot \left(1 + \frac{1}{a_2}\right) \cdot \left(1 + \frac{1}{a_3}\right) \cdot \left(1 + \frac{1}{a_4}\right) \ldots$$

Infinite product ($x < 1$): $y_n = 1 + \left[\frac{1}{|x_n - 1|}\right]$, $x_{n+1} = x_n \frac{y_n}{y_n - 1}$ will give

$$x = \left(1 - \frac{1}{a_0}\right) \cdot \left(1 - \frac{1}{a_1}\right) \cdot \left(1 - \frac{1}{a_2}\right) \cdot \left(1 - \frac{1}{a_3}\right) \cdot \left(1 - \frac{1}{a_4}\right) \ldots$$

Alternated Egyptian product or Pierce expansion: $y_n = \left[\frac{1}{x_n}\right]$, $x_{n+1} = 1 - \{x_n y_n\}$ will give

The inverter
Details of programs

$$\frac{1}{24} = [22, 18, 22, 18, 22, 18, 22, 18, 22, 18, 22, 18, 22, 18, \ldots.]$$

is the expansion if I use the base :

$$\sum_{n=1}^{\infty} \frac{n^3}{(e^{2\pi n} - 1)}$$

What else can be found if I test with the 10821 math constants of the OEIS database ? (I tried it, of course)…

Once you get an expansion there are some possibilities.
Use Gfun (maple) to try to crack the sequence.

The Inverter

The next step is to speedup the lookups. For that, I can multiply the sources (5 disks) and do a lookup in parallel.
I consider using SSD or even M.2 very fast disks. The problem is mainly $$.

Portability. There are several problems.

1- The database is 7.8 TB, most people are afraid of that (TB).
2- To maintain such a database on the web you need at the least a fiber line. (if it is available near you).
3- CPU time : to make one complete test takes several minutes.

any suggestion ?
Thank you for your attention, merci de votre attention.

References :

- Stoutemyer David, How to hunt wild constants : https://arxiv.org/abs/2103.16720
- Plouffe Simon, first tables of the Inverter : https://arxiv.org/abs/2103.16720
- Plouffe Simon, Presentation of the Inverter : https://vixra.org/abs/1409.0151

- This talk is available at : http://plouffe.fr/BIG_DATA_and_MATHEMATICS_Boston_University/

- Interested in big tables ?
- a- Send me a mail at simon.plouffe_hat_gmail.com
- b- Tell me which version you want : the small, the medium or the BIG one. The small one is 368 gigabytes, the medium is 868 gigabytes and the BIG one is 3+ terabytes (all compressed with gzip).
- c- Send a hard disk (internal or external) of at least 1 TB with your address.
- d- I only charge for transport. (a few dollars).
- e- Wait for the data.
- f- Preferably : you need Unix and Maple installed (cygwin64 or Linux).