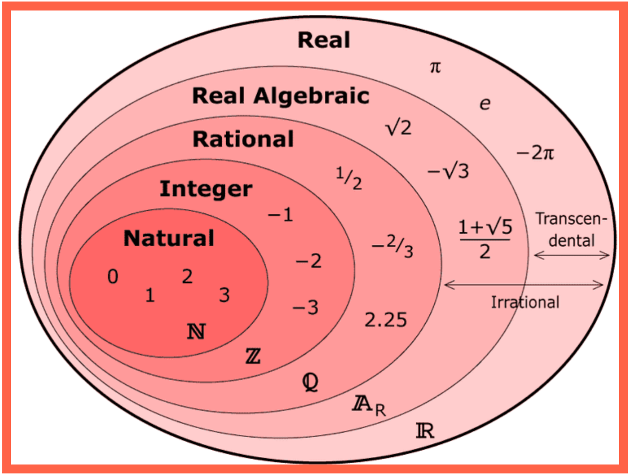
What is...algorithmic number theory?

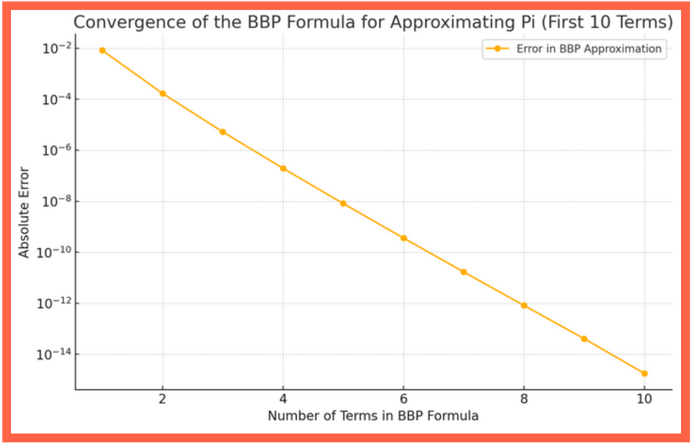
Or: Subfields of mathematics 30

Transcendental number checker



- **Transcendental number** = no relation $a_0 + a_1 x + \ldots + a_n x^n = 0$ for $a_i \in \mathbb{Q}$
- **Task** Determine whether a given number is transcendental
- **Problem** This is super hard; how can we check this even numerically?

Bailey–Borwein–Plouffe (BBP) formula



- The BBP approximation of π is

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

- How on earth can one find such a formula?

Euclid's famous algorithm

Compute the Euclidean algorithm step by step

$a = 1071; b = 462$

$1071 = q_0 \times 462 + r_0$
$q_0 = 2; r_0 = 147$

$462 = q_1 \times 147 + r_1$

$q_1 = 3; r_1 = 21$

$147 = q_2 \times 21 + r_2$

$q_2 = 7; r_2 = 0$
Since $r_2 = 0$ the algorithm is finished. Thus
GCD(1071, 462) = 21.

Restart

Compute the Euclidean algorithm step by step

$a = 4736; b = 462$

$4736 = q_0 \times 462 + r_0$
$q_0 = 10; r_0 = 116$

$462 = q_1 \times 116 + r_1$

$q_1 = 3; r_1 = 114$

$116 = q_2 \times 114 + r_2$

$q_2 = 1; r_2 = 2$

$114 = q_3 \times 2 + r_3$

$q_3 = 57; r_3 = 0$
Since $r_3 = 0$ the algorithm is finished. Thus
GCD(4736, 462) = 2.

Restart

- Bézout's identity $ax + by = d$ is a special case of $a_1x_1 + \ldots + a_nx_n = 0$ (integer relation)
- Euclid's algorithm finds d ridiculously fast (number of steps $\leq$ five times the number of base 10 digits)

Enter, the theorem

Lenstra–Lenstra–Lovász (LLL) lattice basis reduction algorithm can find integer relations in polynomial time

```
INPUT
    a lattice basis b1, b2, ..., bn in Z^m
    a parameter δ with 1/4 < δ < 1, most commonly δ = 3/4
PROCEDURE
    B* <- GramSchmidt({b1, ..., bn}) = {b1*, ..., bn*};  and do not normalize
    μi,j <- InnerProduct(bi, bj*)/InnerProduct(bj*, bj*);   using the most current values of bi
and bj*
    k <- 2;
    while k <= n do
        for j from k−1 to 1 do
            if |μk,j| > 1/2 then
                bk <- bk − ⌊μk,j⌉bj;
               Update B* and the related μi,j's as needed.
               (The naive method is to recompute B* whenever bi changes:
                B* <- GramSchmidt({b1, ..., bn}) = {b1*, ..., bn*})
            end if
        end for
        if InnerProduct(bk*, bk*) > (δ − μ^2 k,k−1) InnerProduct(bk−1*, bk−1*) then
            k <- k + 1;
        else
            Swap bk and  bk−1;
            Update B* and the related μi,j's as needed.
            k <- max(k−1, 2);
        end if
    end while
    return B the LLL reduced basis of {b1, ..., bn}
OUTPUT
    the reduced basis b1, b2, ..., bn in Z^m
```

- This has been improved upon several times (HJLS, PSOS, PSLQ, ...)
- For $\{1, x, x^2, ..., x^n\}$ $\Rightarrow$ a numeric check whether a number is transcendental
- PSLQ gave the BBP formula: see Analysis of PSLQ, an integer relation algorithm
- Algorithmic number theory answers similar questions!

Algorithms of the century

Home / Magazines / Computing in Science & Engineering / 2000.01

Computing in Science & Engineering

Guest Editors Introduction to the top 10 algorithms

Jan.-Feb. 2000, pp. 22-23, vol. 2
DOI Bookmark: 10.1109/MCISE.2000.814652

Authors
J. Dongarra, University of Tennessee
F. Sullivan

DOWNLOAD PDF | SHARE ARTICLE | GENERATE CITATION

Keywords
Linear Algebra, History, Eigenvalues And Eigenfunctions, Linear Programming, Large Scale Systems, Matrices, Art

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

- ▶ Above From the IEEE Computer Society Journal
- ▶ No such list can be perfect but integer relation made it on it should tell us something ☺

Thank you for your attention!

I hope that was of some help.