

# Analytic Solutions to Polynomial Equations

## From Quadratic Formula to Hilbert's 12th Problem

Joe Ferrara

Introductory Number Theory Talk

# Outline

- Number Theory Background
- My Work
  - Joseph Ferrara, **A p-adic Stark conjecture in the rank one setting**, Acta Arithmetica **193** (2020), 369-417.
  - Joseph Ferrara, **Stark's Conjectures for p-adic L-functions**, PhD thesis, UC Santa Cruz (2018).
- Applications

# Algebraic Number Theory

Algebraic number theory is about *understanding* the solutions of polynomial equations with integer coefficients.

A **polynomial** is a function of the form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0.$$

- The  $a_i$ 's are the **coefficients** of  $f$ .
- $n$  is the **degree** of  $f$ .
- The solutions to the equation

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 = 0$$

are called the **roots** of  $f$ .

# Polynomial Examples

**Polynomial**

**Roots**

$$x^2 - 2$$

$$\sqrt{2}, -\sqrt{2}$$

$$x^2 - x - 1$$

$$\frac{1 + \sqrt{2}}{2}, \frac{1 - \sqrt{2}}{2}$$

$$x^2 + 1$$

$$\sqrt{-1}, -\sqrt{-1}$$

$$2x^3 - 6x + 2$$

$$-2\sqrt[3]{1 + \sqrt{-3}}, \sqrt[3]{1 + \sqrt{-3}} + \sqrt[3]{1 - \sqrt{-3}}, \\ -\sqrt[3]{1 + \sqrt{-3}} + \sqrt[3]{1 - \sqrt{-3}}$$

$$x^4 - 6x^2 + 7$$

$$\sqrt{3 + \sqrt{2}}, -\sqrt{3 + \sqrt{2}}, \sqrt{3 - \sqrt{2}}, -\sqrt{3 - \sqrt{2}}$$

# Roots of Polynomials Always Exist

## Fundamental Theorem of Arithmetic

*A degree  $n$  polynomial has  $n$  roots (counting repetitions) in the complex numbers. In other words, any polynomial can be factored into linear factors over the complex numbers.*

A **complex number** is a number of the form  $a + b\sqrt{-1}$  where  $a$  and  $b$  are real numbers.

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-1})(x - \alpha_n)$$

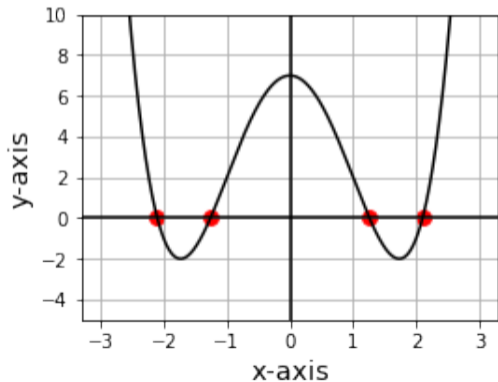
$\alpha_1, \alpha_2, \dots, \alpha_n$  are the roots of  $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$

**Question:** If every polynomial has solutions over the complex numbers, then what is the point of algebraic number theory?

# Understanding Solutions to Polynomials

**Answer:** You want to *understand* the solutions to polynomials, not just know that they exist and be able to write them down.

$$y = x^4 - 6x^2 + 7$$



Roots Formula

$$-\sqrt{3 + \sqrt{2}}$$

$$-\sqrt{3 - \sqrt{2}}$$

$$\sqrt{3 - \sqrt{2}}$$

$$\sqrt{3 + \sqrt{2}}$$

Roots Decimal

-2.10100...

-1.25928...

1.25928...

2.10100...

## General Formulas

- A formula for  $x$  in terms of the coefficients of the polynomial constitutes a *good* understanding of the solutions.

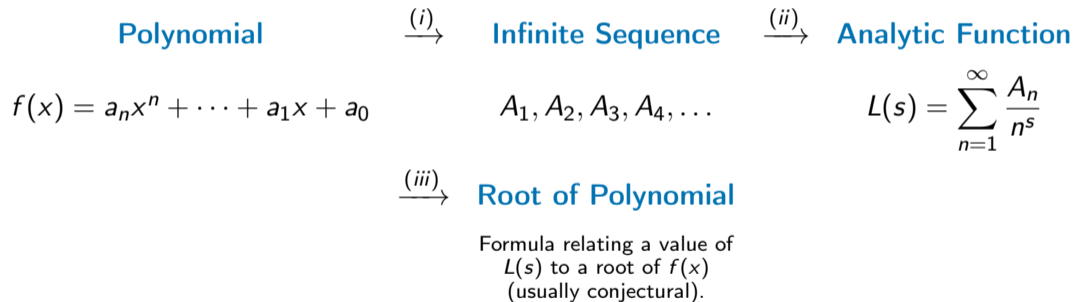
- **Quadratic Formula:** The solutions to  $ax^2 + bx + c = 0$  are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

- **Cubic and Quartic Formulas:** Due to Cardano and Ferrari in 1545.
- **Galois 1832:** There is no general formula for the roots of a degree 5 or higher polynomial in terms of the coefficients using only the operations of  $+$ ,  $-$ ,  $\cdot$ ,  $/$  and taking radicals  $\sqrt{\phantom{x}}$ ,  $\sqrt[2]{\phantom{x}}$ ,  $\sqrt[3]{\phantom{x}}$ ,  $\dots$ ,  $\sqrt[n]{\phantom{x}}$ .
  - Further, Galois gave conditions for when a given polynomial has a formula for its roots in terms of its coefficients.
  - Example:  $x^5 - x + 1 = 0$  has no formula for  $x$ .

## Hilbert's 12th Problem (1900)

For certain categories of polynomials of any degree, find analytic functions that generate the roots of the polynomials.



The analytic functions are known as *L-functions*.



# Examples

**Polynomial**  $\xrightarrow{(i)}$  **Infinite Sequence**

$$x^2 - 4x + 1 \quad 1, -1, 0, 1, -1, 0, 1, -1, 0, 1, -1, 0, 1, -1, 0 \dots$$

$$x^2 - 2x - 1 \quad 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, \dots$$

$\xrightarrow{(ii)}$  **Analytic Function**

$$L(s) = 1 - \frac{1}{2^s} + \frac{1}{4^s} - \frac{1}{5^s} + \frac{1}{7^s} - \frac{1}{8^s} + \frac{1}{10^s} - \dots$$

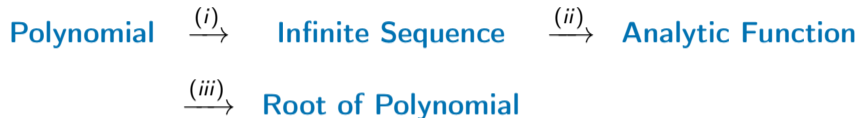
$$L(s) = 1 - \frac{1}{3^s} - \frac{1}{5^s} + \frac{1}{7^s} + \frac{1}{9^s} - \frac{1}{11^s} - \frac{1}{13^s} + \dots$$

$\xrightarrow{(iii)}$  **Root of Polynomial**

$$L(1) = \frac{2 \log(2+\sqrt{3})}{\sqrt{3}}$$

$$L(1) = \frac{\log(1+\sqrt{2})}{\sqrt{2}}$$

# My Work



- For two families of polynomials, I introduced a new set of analytic functions ( $p$ -adic  $L$ -functions).
- For these new analytic functions, I introduced new formulas relating them to roots of polynomials.

# Applications

## Theoretical

- Using  $p$ -adic  $L$ -functions could allow progress on formulas for roots of polynomials that are intractable using classical  $L$ -functions.

## Applied

- New computations of solutions of polynomials.
- New computations of  $p$ -adic  $L$ -functions.
- Numeric representations of new number systems.

## Polynomial Type I (Real Quadratic Fields)

$$\begin{aligned} & x^{20} - 9354219385645x^{19} + \\ & 419796153298286902440x^{18} - 3745796149156532581333733040x^{17} + \\ & +5000414300062302240625744515820070x^{16} - 213628920695837567658122371250985871454x^{15} + \\ & -267441832620194005366570616914604586140x^{14} - 712620126301587970438319446258488482420x^{13} + \\ & -732577810014457814740548349712979253855x^{12} - 981748526337760261014456925673859400785x^{11} + \\ & -922990144478224187891067776508936658344x^{10} + \\ & -981748526337760261014456925673859400785x^9 - 732577810014457814740548349712979253855x^8 + \\ & -712620126301587970438319446258488482420x^7 - 267441832620194005366570616914604586140x^6 + \\ & -213628920695837567658122371250985871454x^5 + 5000414300062302240625744515820070x^4 + \\ & -3745796149156532581333733040x^3 + 419796153298286902440x^2 + \\ & -9354219385645x + 1 \end{aligned}$$

**Note:** Coefficients are symmetric around degree 10 term.

## Polynomials Type II (Imaginary Quadratic Fields)


$$\begin{aligned} & x^{15} - 832535x^{14} + 65231675x^{13} - 5650639400x^{12} + \\ & + 15533478425x^{11} - 39376942640x^{10} - 212804236525x^9 - 380541320125x^8 + \\ & - 2607229594750x^7 - 2183192838625x^6 + 3771011381950x^5 - 1207366794625x^4 + \\ & + 99067277500x^3 - 221569375x^2 + 466875x - 125 \end{aligned}$$

**Note:** All coefficients are divisible by 5.

$$x^9 - 306x^8 - 1143x^7 - 71640x^6 + 60156x^5 + 117180x^4 + 25704x^3 - 7371x^2 + 5022x - 27$$

**Note:** All coefficients are divisible by 3.

# Online Number Theory Databases

 Feedback · Hide Menu

## LMFDB - The L-functions and Modular Forms Database

**Introduction**

Overview Random  
Universe Knowledge

**L-functions**

Rational All

**Modular forms**

Classical Maass  
Hilbert Bianchi

**Varieties**

Elliptic curves over  $\mathbb{Q}$   
Elliptic curves over  $\mathbb{Q}(\alpha)$   
Genus 2 curves over  $\mathbb{Q}$   
Higher genus families  
Abelian varieties over  $\mathbb{F}_q$

**Fields**

Number fields  
 $p$ -adic fields

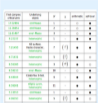
**Representations**

Dirichlet characters  
Artin representations

**Groups**

Galois groups  
Sato-Tate groups


**A database**



The LMFDB is an extensive database of mathematical objects arising in Number Theory.


Sample lists: L-functions, Elliptic curves, Tables of zeros, Number fields

**Hall of fame**



Riemann zeta function  
Ramanujan  $\Delta$  function and its L-function  
C277 and its L-function  
Gauss elliptic curve and its L-function  
Grand Canyon L-function

**Search and browse**

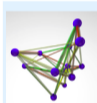


Search for objects with specific properties, or browse categories.

Browse: L-functions, Modular forms, Elliptic curves, Number fields

See a random object from the database


**Visualize data**



Explore individual plots or view distributions of various objects.

Examples:  $GL_4$  Level one Maass forms, Isogeny graph of elliptic curve 102.c

**Explore and learn**



The LMFDB makes visible the connections predicted by the Langlands program. Knows offer background information when you need it.

LMFDB universe Knowledge

**Code and open software**

```
SageMath s.conductor().factor()
N = 2 - 117223
SageMath s.discriminant().factor()
Delta = 2^4 - 117223
SageMath s._L_invariants().factor()
f = 2^0 - 3^1 - 7^1 - 181^1
Evol(E) = Z
```

Download the data, download the code, or see how the data was generated.

GitHub SageMath Pari/GP  
Magma Python

- Website: *lmfdb.org*.
  - Supported by NSF and others.
  - Maintained by mathematicians.
- I used the database in my research.
- My research will contribute to the database.

This project is supported by grants from the US National Science Foundation, the UK Engineering and Physical Sciences Research Council, and the Simons Foundation.

# Online Number Theory Databases

LMFDB Feedback · Hide Menu

Home → L-functions → Rational → 2 → 68 → 68.67 → c0-0 → 0

## L-function 2-68-68.67-c0-0-0

Normalization: arithmetic analytic

**Introduction**

Overview Random  
Universe Knowledge

**L-functions**

Rational All

**Modular forms**

Classical Maass  
Hilbert Bianchi

**Varieties**

Elliptic curves over  $\mathbb{Q}$   
Elliptic curves over  $\mathbb{Q}(\alpha)$   
Genus 2 curves over  $\mathbb{Q}$   
Higher genus families  
Abelian varieties over  $\mathbb{F}_q$

**Fields**

Number fields  
 $p$ -adic fields

**Representations**

Dirichlet characters  
Artin representations

**Groups**

Galois groups  
Sato-Tate groups

**Dirichlet series**

$$L(s) = 1 - 2^{-s} + 4^{-s} - 8^{-s} - 9^{-s} - 2 \cdot 13^{-s} + 16^{-s} + 17^{-s} + 18^{-s} + 25^{-s} + 2 \cdot 26^{-s} - 32^{-s} - 34^{-s} - 36^{-s} - 49^{-s} - 50^{-s} - 2 \cdot 52^{-s} + 2 \cdot 53^{-s} + 64^{-s} + 68^{-s} + 72^{-s} + 81^{-s} - 2 \cdot 89^{-s} + 98^{-s} + 100^{-s} - 2 \cdot 101^{-s} + 2 \cdot 104^{-s} - 2 \cdot 106^{-s} + \dots$$

**Functional equation**

$$\Lambda(s) = 68^{s/2} \Gamma_{\mathbb{C}}(s) L(s) = \Lambda(1-s)$$

**Invariants**

**Degree:** 2  
**Conductor:** 68 =  $2^3 \cdot 17$   
**Sign:** 1  
**Analytic conductor:** 0.0339364  
**Root analytic conductor:** 0.184218  
**Motivic weight:** 0  
**Rational:** yes  
**Arithmetic:** yes  
**Character:**  $\chi_{68}(67, -)$   
**Primitive:** yes  
**Self-dual:** yes  
**Analytic rank:** 0  
**Selberg data:** (2, 68, ( : 0), 1)

**Particular Values**

$L(\frac{1}{2}) \approx 0.3327885867$   
 $L(1)$  not available

**Properties**

**Label:** 2-68-68.67-c0-0-0  
**Degree:** 2  
**Conductor:** 68  
**Sign:** 1  
**Analytic cond.:** 0.0339364  
**Root an. cond.:** 0.184218  
**Motivic weight:** 0  
**Arithmetic:** yes  
**Rational:** yes  
**Primitive:** yes  
**Self-dual:** yes  
**Analytic rank:** 0

**Origins**

Artin representation 2.68.4t3.a  
Artin representation 2.68.4t3.a.a  
Modular form 68.1.d.a  
Modular form 68.1.d.a.67.1

**Downloads**

Euler factors to text  
Zeros to text  
Dirichlet coefficients to text

**Learn more about**

Completeness of the data  
L-function labels  
Source of the data  
Reliability of the data

- This shows the page for the classical  $L$ -function associated to the high degree polynomial from two slides ago.
- I found solutions to this polynomial using a  $p$ -adic  $L$ -functions.
- $p$ -adic  $L$ -functions are not in database yet.

# Sage - Python Based Number Theory Libraries



The screenshot shows the SageMath website homepage. At the top left is the SageMath logo, which consists of a purple cube with white lines and the word "Sage" in white on a purple background. To the right of the logo are links for "RSS", "Blog", "Trac", "Wiki", "Questions?", "Sponsor", and "Donate". Below these are links for "Online: CoCalc", "SageCell", and "Download, Source Code". The version "v9.2 (2020-10-24)" is displayed, along with social media icons for Facebook and Twitter, and a "Language" dropdown menu. A navigation bar contains links for "Home", "Tour", "Help", "Library", "Download", "Development", and "Links". The main content area features a paragraph describing SageMath as a free open-source mathematics software system licensed under the GPL, built on top of many existing open-source packages like NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R, and many more. It also states the mission: "Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab." Below this are two highlighted boxes: one stating that since version 9.0 released in January 2020, SageMath is using Python 3, and another providing links to learn how to use SageMath in English, Spanish, French, and German.

**SageMath** is a free [open-source](#) mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: [NumPy](#), [SciPy](#), [matplotlib](#), [Sympy](#), [Maxima](#), [GAP](#), [FLINT](#), [R](#) and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

Mission: *Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.*

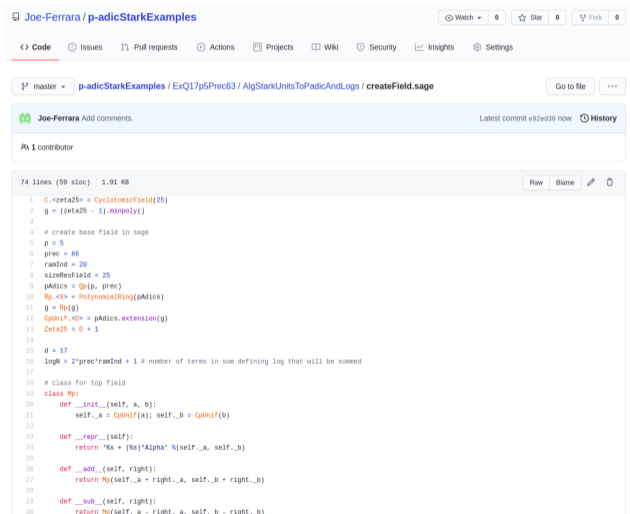
Since version 9.0 released in January 2020, SageMath is using **Python 3**.  
For more information, see the [Python 3 switch wiki](#).

Learn how to use SageMath:  
[Sage for Undergraduates](#) by Gregory Bard (Spanish: [Sage para Estudiantes de Pregrado](#))  
[Mathematical Computation with Sage](#) by Paul Zimmermann et al.  
(French: [Calcul mathématique avec Sage](#), German: [Rechnen mit Sage](#))

- Website: *sagemath.org*.
- Open-source mathematics software.
- Sage is written and interacted with via Python.
- Contains advanced number theory libraries that contribute to research.
- Contributed to and maintained by mathematicians.



# Number System Representation



The screenshot shows a GitHub repository for 'Joe-Ferrara / p-adicStarkExamples'. The file 'createField.sage' is open, displaying Sage code for creating a p-adic number system. The code defines a cyclotomic field, a polynomial, and a p-adic field extension, then defines a class 'Mp' for representing elements in the field.

```
1 C.<zeta25> = CyclotomicField(25)
2 g = (zeta25 - 1).minpoly()
3
4 # create base field in sage
5 p = 5
6 prec = 66
7 ramInd = 20
8 sizeResField = 25
9 pAdics = Qp(p, prec)
10 Rp.<X> = PolynomialRing(pAdics)
11 g = Rp(g)
12 CpUnif.<C> = pAdics.extension(g)
13 Zeta25 = C + 1
14
15 d = 17
16 logN = 2*prec*ramInd + 1 # number of terms in sum defining log that will be summed
17
18 # class for top field
19 class Mp:
20     def __init__(self, a, b):
21         self._a = CpUnif(a); self._b = CpUnif(b)
22
23     def __repr__(self):
24         return '%s + (%s)*Alpha' % (self._a, self._b)
25
26     def __add__(self, right):
27         return Mp(self._a + right._a, self._b + right._b)
28
29     def __sub__(self, right):
30         return Mp(self._a - right._a, self._b - right._b)
```

- I implemented a new number system in Sage.
- It is built on top of a currently implemented number system in Sage.
- The number system is used to represent  $p$ -adic  $L$ -function values and the solutions to polynomials.

## References

- Joseph Ferrara, **A p-adic Stark conjecture in the rank one setting**, Acta Arithmetica **193** (2020), 369-417.
- Joseph Ferrara, **Stark's Conjectures for p-adic L-functions**, PhD thesis, UC Santa Cruz (2018).

Thank You!