Constructive Representation of Multivariate Continuous Functions: A Unifying Program Based on Closure Extensions

Shifa Liu

Abstract

This paper aims to generalize the "differential algebraic closure–specialization homomorphism" framework, developed for solving Hilbert's 13th Problem (representation of the polynomial root map), to the constructive representation problem of general multivariate continuous functions. We follow the four-step methodology of the "Constructive-Analytic Grand Unification" program: (1) **Closure Relativity Principle**: clarify that the "representability" of a function is relative to the allowed operational closure; (2) **Key-Point Differential Forms**: select representative points in the function's domain and extract differential information of the function as the basic "alphabet" for representation; (3) **Stratified Closure Construction**: systematically adjoin generators to construct a "universal representation closure"; (4) **Symmetry-Driven Solution Representation**: organize the standard form of representation using intrinsic symmetries of the function. We first elucidate, using the polynomial root map as an example, how it serves as a perfect prototype for this program. Subsequently, we outline the path to apply this program to general multivariate continuous functions, the core of which lies in constructing a "functional-differential closure". This generalization not only provides a potential constructive version of the Kolmogorov–Arnold representation theorem for multivariate functions but also incorporates the function representation problem into a unified, operational algebraic framework, promising to bridge the gaps between analysis, algebra, and computational mathematics.

Keywords: Hilbert's 13th Problem; Differential Algebra; Closure; Specialization Homomorphism; Septic Equation; Function Composition; Constructive Analysis; Closure Relativity; Differential Galois Theory; Multivariate Function Representation; Moving Least Squares; Branch Selection; High-Power Functions

1 Introduction: From a Special Case to a Program

The revolutionary nature of the differential algebraic solution to Hilbert's 13th Problem lies not only in providing an explicit formula but also in its inadvertent practice of a potentially universal methodology. The traditional view holds that the insolvability by radicals of polynomial root maps and the complex representation of multivariate continuous functions are two distinct problems. However, we find they share a deep structure: both concern how to represent a complex map defined on a high-dimensional space using a finite number of "simple" functions.

The four-step scheme of the "Constructive-Analytic Grand Unification" program [1] provides a clear blueprint for extracting and generalizing this methodology. This program claims that, by appropriately extending the "closure" of mathematical operations, many classic unsolvable or undecidable problems can obtain constructive solutions. This paper aims to systematically elaborate on how to generalize the successful "four-step" scheme for polynomial roots to the representation problem of arbitrary multivariate continuous functions, thereby providing a constructive, algebraic, and potentially more computationally tractable alternative framework for the Kolmogorov–Arnold representation theorem [2].

1.1 Hilbert's 13th Problem and Its Algebraic Variant

In 1900, David Hilbert, in his famous 23 problems, posed the 13th problem, originally concerned with the functional representation of roots of the septic equation [4]. He asked: Can the root of the equation $x^7 + ax^3 + bx^2 + cx + 1 = 0$ be expressed as a composition of a finite number of continuous functions of at most two variables in the coefficients a, b, c? This problem deeply touches upon the mathematical core of "dimension reduction" representation of multivariate functions.

A more general and natural algebraic variant subsequently arose: For any general polynomial equation $f(x) = 0$ of degree n, can its roots x_k be expressed as a composition of a finite number of continuous functions of at most two variables in the coefficients? The primary goal of this paper is to provide an affirmative, constructive answer to this generalized problem.

Classical Galois theory and the Abel–Ruffini theorem assert that general polynomial equations of degree five and higher are not solvable by radicals [5]. This seemed to cast a shadow over Hilbert's question. However, the category of "continuous functions" allowed by Hilbert is far larger than that of "algebraic functions". In 1957, the representation theorem proved by Kolmogorov and Arnold [2] states that any multivariate continuous function can be represented as a composition of a finite number of univariate continuous functions and addition. Although this theorem is a milestone, the constructed functions are highly complex, non-elementary, and do not provide an explicit, computable representation scheme for structured objects like polynomial root maps.

1.2 Our Approach: Differential Algebra and Closure Extension

This paper proposes a distinctly different path: differential algebra and closure extension. We prove that by systematically extending the allowed set of operations from the traditional "radical closure" to the "differential algebraic closure" that includes differential operations, one can provide a completely explicit and unified formula for the roots of polynomial equations within a carefully constructed extension field. The core innovations of this solution lie in:

1. **Universal Differential Algebraic Closure** $\mathcal{K}_n$: a pre-built "solution template" field for all equations of degree n.
2. **Specialization Homomorphism** $\iota_f : \mathcal{K}_n \to \mathcal{L}_f$: applying the above template to a specific polynomial f, obtaining the differential closed extension field $\mathcal{L}_f$ containing its roots.

The contribution of this paper extends beyond solving this specific problem. We further discover that the above solution practices and confirms a more ambitious mathematical program of "Constructive-Analytic Grand Unification" [1]. This program, based on the Closure Relativity Principle, advocates that many classic unsolvable problems within classical frameworks can obtain constructive solutions through systematic extension of the allowed mathematical operational closure. The work in this paper provides a perfect, detailed exemplar for this program.

Therefore, the second and more profound goal of this paper is to follow the "four-step" methodology of this program, systematically generalize the differential algebraic scheme we developed for solving Hilbert's problem to the constructive representation problem of general multivariate continuous functions, and establish the corresponding differential Galois theoretical framework. This aims to provide a constructive, algebraic, and more computationally tractable alternative framework for the Kolmogorov–Arnold representation theorem.

1.3 Structure of the Paper

Section 2 will fully elaborate the differential algebraic solution for the algebraic variant of Hilbert's 13th Problem. This solution is the blueprint and cornerstone for all subsequent generalizations.

Section 3 will, according to the "Constructive-Analytic Grand Unification" program, systematically abstract and generalize the method in Section 2 into a "four-step" methodological framework applicable to the representation problem of general multivariate continuous functions.

Section 4 will establish the corresponding differential Galois theory for this generalized framework to rigorously characterize symmetry and solvability within closure extensions. Section 5 will present a concrete realization of this program on multivariate continuous functions: an explicit analytic representation theory based on functional-differential closure and moving least squares (MLS).

Section 6 will address more complex situations: for high-power multivariate continuous functions, introducing combinatorial correction terms and branch selection mechanisms to establish a complete representation theory.

Section 7 concludes the paper, discussing future research directions and the profound significance of this unifying program.

2 Differential Algebraic Closure Solution to Hilbert's 13th Problem

This section will fully elaborate the differential algebraic solution for the algebraic variant of Hilbert's 13th Problem. This solution is the blueprint and cornerstone for all subsequent generalizations.

2.1 Construction of the Universal Differential Algebraic Closure $\mathcal{K}_n$

Let $\mathbb{F}$ be a differential field of characteristic zero (e.g., the rational number field $\mathbb{Q}$ and its transcendental extensions, equipped with the trivial derivation). Consider the set of formal variables $\{A_0, A_1, \ldots, A_n\}$, which will serve as "placeholders" for the coefficients of future degree-n polynomials.

[Universal Polynomial] Define the *universal polynomial* of degree n as:

$$F(X) = A_0X^n + A_1X^{n-1} + \cdots + A_{n-1}X + A_n \in \mathbb{F}[A_0, \ldots, A_n][X].$$

We construct the closure by adjoining generators in layers.

[Stratified Generators] Adjoin the following abstract generators successively to the coefficient field $\mathbb{F}(A_0, \ldots, A_n)$:

1. **Critical Point**: Adjoin a symbol χ, stipulating that it satisfies the relation defined by the derivative of $F(X)$: $F'(\chi) = 0$. Formally, we consider:

$$\mathcal{R}_1 = \frac{\mathbb{F}(A_0, \ldots, A_n)[\chi]}{\langle F'(\chi) \rangle}.$$

The introduction of χ captures the symmetry of the sum of polynomial roots: by Viète's formulas, χ specializes to $-\frac{A_1}{nA_0}$.

https://doi.org/10.33774/coe-2026-pxlkw ORCID: https://orcid.org/0009-0003-6570-2812

2. **Formal Derivative Values**: For $m = 0, 1, \dots, n-1$, adjoin formal symbols Φ_m, stipulating $\Phi_m = F^{(m)}(\chi)$. This is realized by adding the relations $\langle \Phi_m - F^{(m)}(\chi) \rangle$.

3. **Root of Unity**: Adjoin the primitive nth root of unity $\omega_n = e^{2\pi i/n}$.

4. n**th Roots of Rational Functions**: For $m = 1, \dots, n-1$, adjoin symbols Θ_m, stipulating $\Theta_m^n = \frac{\Phi_m}{nA_0}$.

[Universal Differential Algebraic Closure] The differential field obtained by successively extending with all the above generators is called the *universal differential algebraic closure* of degree n, denoted $\mathcal{K}_n$. Formally:

$$\mathcal{K}_n := \mathbb{F}(A_0, \dots, A_n) \left(\chi, \{\Phi_m\}_{m=0}^{n-1}, \omega_n, \{\Theta_m\}_{m=1}^{n-1} \right),$$

where each generator satisfies the relations defined above.

2.2 Specialization Homomorphism and the Solution Field $\mathcal{L}_f$

$\mathcal{K}_n$ is a universal "template". For a concrete polynomial, we obtain a concrete solution via a specialization homomorphism.

[Concrete Polynomial and Coefficient Field] Let $f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$ be a polynomial of degree n with coefficients in the field $\mathbb{F}$ ($a_0 \neq 0$). Denote $\mathbb{F}_f = \mathbb{F}(a_0, \dots, a_n)$.

[Existence of Specialization Homomorphism] For the above concrete polynomial f, there exists a differential field homomorphism (called the *specialization homomorphism*):

$$\iota_f : \mathcal{K}_n \longrightarrow \mathcal{L}_f,$$

satisfying the following assignment rules:

1. $\iota_f(A_i) = a_i,\ i = 0, \dots, n.$

2. $\iota_f(\chi) = x_c := -\frac{a_1}{na_0}$ (the critical point of f).

3. $\iota_f(\Phi_m) = f^{(m)}(x_c),\ m = 0, \dots, n-1.$

4. $\iota_f(\omega_n) = e^{2\pi i/n}$.

5. $\iota_f(\Theta_m) = \left(\frac{f^{(m-1)}(x_c)}{na_0} \right)^{1/n}$, where $^{1/n}$ denotes taking a specific nth root (a branch needs to be fixed).

Here, $\mathcal{L}_f$ is a differentially closed extension field of $\mathbb{F}_f$, called the *differential algebraic solution field* of f.

https://doi.org/10.33774/coe-2026-pxlkw **ORCID:** https://orcid.org/0009-0003-6570-2812 Content not peer-reviewed by Cambridge University Press.

Proof. The construction of the homomorphism is direct. Starting from the definition of $\mathcal{K}_n$, map all relations among its generators via ι_f into $\mathcal{L}_f$. Since in $\mathcal{L}_f$, the assigned equalities hold by definition, ι_f is a well-defined ring homomorphism. By appropriately constructing $\mathcal{L}_f$ (e.g., as a subfield of the differential closure of $\mathbb{F}_f$ containing all necessary elements), one can ensure it is a field and the homomorphism is injective or has a well-defined kernel. This constructive process guarantees the existence of ι_f. □

[Compatibility with the Abel–Ruffini Theorem] For a general polynomial f of degree $n \geq 5$, its differential algebraic solution field $\mathcal{L}_f$ is generally *not* contained in the radical closure $\mathbb{F}_f^{\mathrm{rad}}$ of $\mathbb{F}_f$.

Justification. According to the Abel–Ruffini theorem [5], when $n \geq 5$, the roots of a general polynomial f cannot all belong to $\mathbb{F}_f^{\mathrm{rad}}$. The construction of $\mathcal{L}_f$ introduces the values of derivatives $f^{(m)}(x_c)$ at a specific point and takes their nth roots. The differentiation operation $f \mapsto f^{(m)}(x_c)$ is a differential operation, not a purely algebraic one; its result is generally not an algebraic function of the coefficients. Therefore, $f^{(m)}(x_c)$ and its nth roots are typically not in $\mathbb{F}_f^{\mathrm{rad}}$. Our framework, by allowing such differential operations, legitimately "bypasses" the restriction of solvability by radicals, thus not contradicting but rather revealing that explicit solutions exist under the broader "differential algebraic" operations. □

2.3 Universal Solution Formula and Its Realization for the Septic Equation

[Universal Differential Algebraic Solution Formula] In the universal differential algebraic closure $\mathcal{K}_n$, there exists a formal expression representing the roots as follows: for $k = 0, 1, \ldots, n-1$,

$$\mathcal{X}_k := \chi + \sum_{m=1}^{n-1} \Theta_m \cdot \omega_n^{s_m} \cdot \omega_n^{mk}.$$

Here, s_m is a set of integers (mod n) determined by "consistency conditions" that ensure the $\iota_f(\mathcal{X}_k)$ obtained via the specialization homomorphism ι_f are precisely all n roots of f.

Now, we focus on the case in Hilbert's original problem: $n = 7$. Consider the general septic equation:

$$f(x) = a_0x^7 + a_1x^6 + a_2x^5 + a_3x^4 + a_4x^3 + a_5x^2 + a_6x + a_7 = 0, \quad a_0 \neq 0.$$

According to Theorem 2.7 and the specialization homomorphism, its seven roots x_k, $(k =$

$0, \ldots, 6$) are given by the formula:

$$x_k = x_c + \sum_{m=1}^{6} \left(\frac{\phi_m}{7a_0} \right)^{1/7} \cdot \omega_7^{s_m} \cdot \omega_7^{mk}.$$

where:

- Critical point: $x_c = -\frac{a_1}{7a_0}$.
- Differential algebraic quantities: $\phi_m = f^{(m-1)}(x_c) - \gamma_m^{(7)} \cdot a_{m-3}$, where $\gamma_m^{(7)} = \frac{7!}{m(7-m)}$. By convention, $a_{m-3} = 0$ when $m - 3 < 0$.
- Root of unity and exponents: $\omega_7 = e^{2\pi i/7}$, s_m are specific integers modulo 7.
- Root operation: $(\cdot)^{1/7}$ denotes taking a specific branch of the seventh root of a complex number.

The formula clearly shows that the roots of the septic equation are expressed as: a rational function (x_c), a linear combination of seventh roots of six algebraic expressions ($\phi_m/(7a_0)$) formed from derivatives and coefficients, and powers of the unit root.

2.4 Constructive Decomposition as Composition of Bivariate Functions: The Septic Equation as Example

We now argue that the map from coefficients $(a_0, \ldots, a_7)$ to roots x_k defined by the above formula can be represented as a composition of finitely many elementary functions each depending on at most two variables. This directly addresses the algebraic variant of Hilbert's 13th Problem.

We break down the process into several levels. Consider $n = 7$ and k fixed; for simplicity, assume s_m are predetermined.

2.4.1 Level 1: Compute Basic Intermediate Quantities

- **Compute critical point**: $x_c = -\frac{a_1}{7a_0}$. This is the composition of the binary division $\mathrm{Div}(u, v) = u/v$ and the unary negation $\mathrm{Neg}(u) = -u$: $x_c = \mathrm{Neg}(\mathrm{Div}(a_1, \mathrm{Mul}(7, a_0)))$.
- **Compute powers**: Compute $x_c^2, x_c^3, \ldots, x_c^7$. Done via iterative binary multiplication, e.g., $x_c^2 = \mathrm{Mul}(x_c, x_c)$.
- **Compute derivatives** $f^{(m-1)}(x_c)$: Use Horner's algorithm, employing only a series of binary multiplications and additions. For example:

$$f(x_c) = ((\cdots((a_0x_c + a_1)x_c + a_2)x_c + a_3)x_c + \cdots)x_c + a_7.$$

- **Compute ϕ_m**: After obtaining $f^{(m-1)}(x_c)$, computing ϕ_m is a binary subtraction: $\phi_m = f^{(m-1)}(x_c) - \gamma_m^{(7)} a_{m-3}$ (if a_{m-3} exists).

2.4.2 Level 2: Handle Radicals and Powers of the Unit Root

- **Compute radical terms**: $u_m = \phi_m/(7a_0)$ (binary division), then $t_m = u_m^{1/7}$. Taking the seventh root $\mathrm{Pow}_7(u) = u^{1/7}$ is a unary function.

- **Compute unit root weights**: $\omega_7^{s_m}$ and ω_7^{mk}. Since the exponents are integers modulo 7, they can be treated as a unary lookup function $\mathrm{Exp}_7(t) = \omega_7^{t \bmod 7}$ from the discrete set $\mathbb{Z}/7\mathbb{Z}$ to seven points on the unit circle.

2.4.3 Level 3: Combine and Sum

- **Form additive terms**: $v_{m,k} = t_m \times \omega_7^{s_m} \times \omega_7^{mk}$. This involves two complex multiplications (binary operations).

- **Summation**: $S_k = \sum_{m=1}^{6} v_{m,k}$. Realized via a series of binary complex additions.

- **Obtain root**: $x_k = x_c + S_k$, a binary addition.

2.4.4 Overall Composition Structure Analysis

The above process forms a directed acyclic computation graph, where the in-degree (number of inputs) of each node is at most 2. All operations are only: binary rational operations (add, subtract, multiply, divide), unary transcendental operations (Pow_7), unary discrete functions (Exp_7), and binary complex multiplication. No step requires operating on three or more original coefficients simultaneously. Therefore, the root-finding map for the septic equation can be decomposed into a composition of finitely many bivariate and univariate elementary functions, constructively answering Hilbert's 13th Problem.

3 Methodological Generalization under the "Constructive-Analytic Grand Unification" Program

The differential algebraic solution to Hilbert's 13th Problem, in practice, inadvertently implemented a "four-step" scheme of universal applicability. This section will systematically generalize this method to the constructive representation problem of multivariate continuous functions, following the "Constructive-Analytic Grand Unification" program [1].

3.1 Step 1: Closure Relativity Principle

Principle Statement: Whether a function $F : \mathbb{R}^m \to \mathbb{R}$ can be "simply" represented is not an absolute property, but depends on the allowed set of basic operations $\mathcal{O}$ and the closure $\overline{\mathcal{O}}$ it generates. Classical unsolvability theorems (e.g., Abel–Ruffini, undecidability of Hilbert's Tenth Problem) should be understood as limitations within specific closures, not as absolute boundaries of mathematics itself.

Manifestation in Polynomial Root Finding:

- **Classical closure**: Radical closure $\mathcal{O}_{\text{rad}} = \{+, -, \times, \div, \sqrt[n]{\cdot}\}$. Within this closure, the root-finding map F_{poly} is not representable ($n \geq 5$).
- **Extended closure**: Differential algebraic closure $\mathcal{O}_{\text{diff-alg}} = \mathcal{O}_{\text{rad}} \cup \{\partial, \text{eval}_{x_c}\}$. Within this closure, F_{poly} obtains a finite explicit representation.

Generalization to Multivariate Continuous Functions:
For an arbitrary $F : \mathbb{R}^m \to \mathbb{R}$, we seek a constructively defined extended closure $\mathcal{O}_{\text{fun-diff}}$, which includes:

1. Operations on F itself (e.g., evaluation at key points, taking partial derivatives).
2. A set of "structural functions" for combination.
3. "Combinatorial rules" controlling complexity.

The existence and conciseness of representation thus transform into the art of designing a sufficiently rich language (closure) for the target family of functions.

3.2 Step 2: Key-Point Differential Forms

Method Statement: Extract a finite set of "features" of the function — select a set of key points $\{\mathbf{p}_i\}_{i=1}^N$ in the domain, and obtain the tensor-valued differential information on them:

$$\text{Data}(F) = \{F(\mathbf{p}_i), \nabla F(\mathbf{p}_i), \nabla^2 F(\mathbf{p}_i), \dots \mid i = 1, \dots, N\}.$$

This set of "differential fingerprints," under sufficient smoothness assumptions, can uniquely determine the function.

Manifestation in Polynomial Root Finding:
The key point is the critical point x_c, and the differential forms are $f(x_c), f'(x_c), \dots$. We transform the dependence on the high-dimensional coefficient space into dependence on the differential information of a one-dimensional auxiliary function at a specific point.

Generalization to Multivariate Continuous Functions:

1. **Key-Point Selection Strategy**: Can be based on gradients (stationary points), systematic sampling, or defined via auxiliary optimization problems.

2. **Information Extraction**: Compute the Taylor expansion coefficients (partial derivatives of all orders) of the function at the key points.

3. **Generators**: Abstract these differential data as formal symbols $\Phi_{i,\alpha}$, serving as atomic generators for subsequent closure construction.

3.3 Step 3: Stratified Closure Construction

Construction Statement: Based on the formal generators $\{\Phi_{i,\alpha}\}$, systematically construct the "functional-differential closure" $\mathcal{K}_{\text{Fun}}(m, d, N)$, with parameters including dimension m, maximum differential order d, and number of key points N.

Layered Construction:

1. **Base Field**: $\mathbb{F}$.

2. **Key-Point Coordinates**: Adjoin formal symbols $\mathbf{P}_1, \ldots, \mathbf{P}_N$.

3. **Differential Form Generators**: Adjoin $\Phi_{i,\alpha}$, stipulating the relations $\Phi_{i,\alpha} = \partial^\alpha F(\mathbf{P}_i)$, where F is a formal universal function.

4. **Structural Function Generators**: Adjoin invariants reflecting symmetries (e.g., distance $\|\mathbf{P}_i - \mathbf{P}_j\|$) and a preset library of combinatorial functions $\mathcal{G} = \{g_1, g_2, \ldots\}$.

5. **Closure Operation Generators**: Introduce allowed operations, such as taking roots, special functions solving differential equations, etc.

6. **Syntax Definition**: Well-formed expressions in $\mathcal{K}_{\text{Fun}}$ are combinations obtained from generators by finitely many applications of structural functions and closure operations.

This closure is a universal syntactic environment designed for representing a certain class of multivariate functions. For a concrete function F, we map to a concrete field $\mathcal{L}_F$ via a specialization homomorphism ι_F.

3.4 Step 4: Symmetry-Driven Solution Representation

Representation Statement: Within the closure $\mathcal{K}_{\text{Fun}}$, design a standard organizational form to encode the "differential fingerprint" $\text{Data}(F)$. This is essentially an encoding-decoding scheme.

Manifestation in Polynomial Root Finding:
The standard form is $x_k = x_c + \sum \Theta_m \omega_n^{s_m} \omega_n^{mk}$, utilizing the cyclic symmetry of roots of unity (ω_n^{mk}) to organize the representation.

https://doi.org/10.33774/coe-2026-pxlkw ORCID: https://orcid.org/0009-0003-6570-2812

Generalization to Multivariate Continuous Functions:
A possible standard form is a hierarchical tensor network representation:

$$\widehat{F}(\mathbf{x}) = \mathcal{C}\left(\sum_{\text{indices}} \mathcal{T}^{(1)}(\Phi_{i_1,\cdot}) \otimes \mathcal{T}^{(2)}(\Phi_{i_2,\cdot}, \mathbf{x}) \otimes \cdots \otimes \mathcal{T}^{(L)}(\Phi_{i_L,\cdot}, \mathbf{x})\right),$$

where $\mathcal{T}^{(l)}$ are linear/bilinear maps (concretizations of structural functions) dependent on the level l, $\otimes$ is a combinatorial operation, and $\mathcal{C}$ is the final contraction.

Incorporation of Symmetries:

- If F has translation invariance, the expression should only depend on $\mathbf{x} - \mathbf{P}_i$.
- If F has rotational invariance, the expression should be built upon rotation invariants (e.g., dot products).

The organizing principle is: representing the global function as an adaptive combination of local differential information constrained by symmetries.

4 Differential Galois Theory for the Generalized Framework

To rigorously characterize symmetry and solvability within the extended closure, we need to establish the corresponding differential Galois theory. This theory is a natural generalization of classical Galois theory to the category of differential algebraic closures and is also the group-theoretic formulation of the "Closure Relativity Principle".

4.1 Closure Relativity and the Differential Galois Group

[Relative Closure] Let $\mathbb{F}$ be a differential field, and $\mathcal{E}$ an extension of it.

- Classical algebraic closure: denoted by $\overline{\mathcal{E}}^{\text{alg}}/\mathbb{F}$.
- Differential algebraic closure: denoted by $\widetilde{\mathcal{E}}^{\text{diff}}/\mathbb{F}$, i.e., the maximal differential field containing $\mathcal{E}$ and differentially algebraic over $\mathbb{F}$.

[Group-Theoretic Formulation of Closure Relativity] The "solvability" of a differential equation (or function representation problem) is characterized by the symmetry group of the corresponding closure:

- In $\overline{\mathcal{E}}^{\text{alg}}$, symmetry is described by the classical Galois group $\text{Gal}^{\text{alg}}(\overline{\mathcal{E}}^{\text{alg}}/\mathbb{F})$, whose unsolvability corresponds to insolvability by radicals.
- In $\widetilde{\mathcal{E}}^{\text{diff}}$, symmetry should be described by the differential Galois group $\text{Gal}^{\text{diff}}(\widetilde{\mathcal{E}}^{\text{diff}}/\mathbb{F})$, whose solvability corresponds to representability by differential algebraic means.

[Universal Differential Galois Group] For the universal differential algebraic closure $\mathcal{K}_n$, its *universal differential Galois group* is defined as:

$$\mathcal{G}_n^{\text{diff}} := \text{Aut}^{\text{diff}}(\mathcal{K}_n/\mathbb{F}(A_0, \ldots, A_n)),$$

i.e., the group of all automorphisms of $\mathcal{K}_n$ that fix the base field and its differential structure. This is an infinite-dimensional differential algebraic group.

[Hierarchical Correspondence of Structure] The structure of $\mathcal{G}_n^{\text{diff}}$ is stratified, corresponding to the construction of $\mathcal{K}_n$:

1. Critical point layer: The action on χ corresponds to symmetry of the average of roots.
2. Differential form layer: The action on Φ_m must be compatible with the chain rule, introducing constraints from differential relations.
3. Radical and root of unity layer: The action on Θ_m and ω_n forms an affine group, containing the cyclic group $\mathbb{Z}/n\mathbb{Z}$ and scalar multiplication from radicals.

For general polynomials of degree five and higher, their classical Galois group is unsolvable, but $\mathcal{G}_n^{\text{diff}}$ is a solvable differential algebraic group. This precisely embodies the group-theoretic manifestation that "the problem becomes solvable within the extended closure."

4.2 Key-Point Differential Invariants and the Specialization Homomorphism

[Key-Point Differential Data Set] For a concrete function F, its key-point differential data set Data(F) is considered as elements in some differential field $\mathbb{F}_F$.

[Galois Interpretation of the Specialization Homomorphism] The specialization homomorphism $\iota_F : \mathcal{K} \to \mathcal{L}_F$ induces a reverse map of group homomorphisms:

$$\iota_F^* : \text{Gal}^{\text{diff}}(\mathcal{L}_F/\mathbb{F}_F) \hookrightarrow \mathcal{G}^{\text{diff}},$$

whose image is called the *intrinsic differential Galois group* G_F^{diff} of F.

[Differential Invariant Principle] The quantities that remain invariant under the action of the intrinsic differential Galois group G_F^{diff} of a function F (the differential invariants) are precisely the basic building blocks that can be written in the "standard representation form."

Example: In the polynomial case, $\phi_m/(na_0)$ is an invariant under G_f^{diff}, thus its nth root can serve as a determined coefficient in the solution formula.

4.3 Solvable Group Sequence Corresponding to Stratified Closure

[Closure Filtration and Group Filtration] The hierarchical construction of the universal closure $\mathbb{F} = \mathcal{K}_0 \subset \mathcal{K}_1 \subset \cdots \subset \mathcal{K}_L = \mathcal{K}$ corresponds to a reverse-order filtration of the differential Galois group $\mathcal{G}^{\text{diff}} = G_0 \triangleright G_1 \triangleright \cdots \triangleright G_L = \{\text{Id}\}$, where $G_i := \text{Aut}^{\text{diff}}(\mathcal{K}/\mathcal{K}_i)$.

[Solvable Structure Theorem] For the closure $\mathcal{K}$ constructed for function representation, its differential Galois group filtration satisfies: the successive quotient groups G_{i-1}/G_i are solvable and have clear geometric interpretations (affine linear groups, diagonalizable groups, abelian groups). Therefore, the entire differential Galois group $\mathcal{G}^{\text{diff}}$ is a solvable differential algebraic group. This theorem systematically explains why explicit solutions can be obtained within the extended closure: because its symmetry group is solvable, allowing us to "inversely solve" for the desired elements layer by layer, akin to successively taking roots.

4.4 Symmetry-Driven Representation Module

[Representation Module] Let V_F be the "representation space" of function F, i.e., the algebraic variety formed by all possible expressions generated from $\text{Data}(F)$ via the structural function library. G_F^{diff} naturally acts on V_F.

[Uniqueness of Standard Representation] The "standard representation form" designed in the program defines a special point $\mathbf{v}_0$ in V_F, characterized by the following properties:

1. **Invariance**: Fixed under a maximal solvable subgroup H of G_F^{diff}.
2. **Minimality**: The H-orbit containing $\mathbf{v}_0$ is minimal under some complexity measure.
3. **Universality**: The mapping from $\mathbf{v}_0$ to $F(\mathbf{x})$ is given by a set of predetermined operators compatible with the representation theory of G_F^{diff}.

[Differential Galois Classification of Function Representations] Two functions F and G have "isomorphic" representation forms if and only if: $G_F^{\text{diff}} \cong G_G^{\text{diff}}$ and $\text{Data}(F)$ and $\text{Data}(G)$ belong to the same representation space under the group isomorphism.

5 Explicit Analytic Representation of Multivariate Continuous Functions: A Constructive Theory Based on Functional-Differential Closure and Specialization Homomorphism

This section, based on the "Constructive-Analytic Grand Unification" program, provides an explicit, constructive analytic representation for multivariate continuous functions defined on compact sets.

5.1 Introduction and Preparation

The classical Weierstrass Approximation Theorem shows that continuous functions on compact sets can be uniformly approximated by polynomials. However, this does not provide an explicit representation formula based on the intrinsic differential structure of the function itself. The Kolmogorov–Arnold theorem, while proving that continuous functions can be represented as compositions of finitely many univariate functions, is non-constructive.

We assume the target function F is defined on the m-dimensional unit cube $K = [0,1]^m$, and $F \in C^r(K, \mathbb{R})$, i.e., has continuous partial derivatives up to order r. The following construction can be generalized to other compact domains.

5.2 Construction of the Functional-Differential Closure $\mathcal{K}_{\mathbf{Fun}}$

5.2.1 Formal Key Points and Differential Data Generators

Let $\mathbb{F} = \mathbb{R}$ be the base field. We introduce the following formal symbols as generators:

1. **Formal Key Point Set**: For a given discretization precision parameter N, introduce N^m formal symbols:

   $$\mathbf{P}_{\mathbf{i}} := (P_{\mathbf{i}}^1, P_{\mathbf{i}}^2, \ldots, P_{\mathbf{i}}^m), \quad \mathbf{i} = (i_1, \ldots, i_m) \in \{1, \ldots, N\}^m.$$

   They represent nodes of a uniform grid on K, e.g., can be specialized to $P_{\mathbf{i}}^j = (i_j - 1)/(N-1)$.

2. **Formal Partial Derivative Value Generators**: For each key point $\mathbf{P}_{\mathbf{i}}$ and each multi-index $\alpha = (\alpha_1, \ldots, \alpha_m)$ satisfying $|\alpha| = \sum \alpha_j \le r$, introduce a formal symbol:

   $$\Phi_{\mathbf{i},\alpha}.$$

They are to be interpreted as partial derivative values of a "formal function" $\mathcal{F}$ at point $\mathbf{P_i}$: $\Phi_{\mathbf{i},\alpha} \equiv \partial^\alpha \mathcal{F}(\mathbf{P_i})$.

3. **Structural Function Generators**: Introduce a finite, predetermined structural function library $\mathcal{G}$. For example:

G = {addition (+), multiplication (×), scalar multiplication, ...} distance function $\mathrm{d}(\cdot,\cdot)$, exponential function exp, sine function sin, ...}. Each $g \in \mathcal{G}$ has a fixed arity (e.g., binary function). Also, we introduce formal input variables $\mathbf{X} = (X_1, \ldots, X_m)$.

5.2.2 Algebraic Construction of the Stratified Closure

We start from the base ring $R_0 = \mathbb{F}[\{P_{\mathbf{i}}^j\}, \{X_k\}]$, i.e., the polynomial ring containing all formal key-point coordinates and input variables.

1. **First Layer Extension (Differential Data Layer)**: Adjoin the formal partial derivative value generators as new indeterminates, forming the ring:

$$R_1 = R_0[\{\Phi_{\mathbf{i},\alpha}\}_{|\alpha| \le r}].$$

In this ring, we do not impose any relations among these $\Phi_{\mathbf{i},\alpha}$ for now (e.g., commutativity $\partial_j \partial_k = \partial_k \partial_j$). These relations will be automatically satisfied by the differentiability of concrete functions upon specialization.

2. **Second Layer Extension (Structural Function Closure Layer)**: We recursively define the $\mathcal{G}$-closure. Let $S_0 = R_1$. For $t \ge 0$, define S_{t+1} as the ring generated by S_t together with all elements of the form $g(a_1, \ldots, a_n)$, where $g \in \mathcal{G}$ is an n-ary function, and $a_1, \ldots, a_n \in S_t$. Since $\mathcal{G}$ is finite, this process stabilizes after finitely many steps (if expression depth is limited). We define the structural function closure as:

$$\mathcal{R} = \bigcup_{t \ge 0} S_t.$$

Elements of $\mathcal{R}$ are called well-formed functional expressions.

3. **Third Layer Extension (Introducing Analytic Closure Operations)**: To handle approximation and limits, we add formal limit symbols to $\mathcal{R}$. Specifically, for a sequence of expressions $\{E_n\}$ in $\mathcal{R}$, we introduce a formal symbol $\lim_{n\to\infty} E_n$. All such formal limits and their finite combinations form a ring $\widehat{\mathcal{R}}$.

4. **Final Definition**: The functional-differential closure $\mathcal{K}_{\text{Fun}}$ is defined as the fraction field of $\widehat{\mathcal{R}}$ (allowing formal rational operations):

$$\mathcal{K}_{\text{Fun}} := \mathrm{Frac}(\widehat{\mathcal{R}}).$$

https://doi.org/10.33774/coe-2026-pxlkw ORCID: https://orcid.org/0009-0003-6570-2812

$\mathcal{K}_{\text{Fun}}$ is a vast field containing all possible formal expressions composed from key-point differential data via structural functions and limit processes. It is our "universal stage" for seeking function representations.

5.3 Measure-Theoretic Specialization Homomorphism ι_F and the Representation Field $\mathcal{L}_F$

For a concrete C^r function $F : K \to \mathbb{R}$, we need to "specialize" formal expressions in $\mathcal{K}_{\text{Fun}}$ into concrete expressions about F. This is achieved via a measure-theoretic specialization homomorphism.

[Valuation Map] First define the valuation on generators:

1. $\iota_F(P_{\mathbf{i}}^j) = p_{\mathbf{i}}^j \in \mathbb{R}$, the specific coordinates of grid points.
2. $\iota_F(X_k) = x_k$, the ordinary real variables.
3. $\iota_F(\Phi_{\mathbf{i},\alpha}) = \partial^\alpha F(\mathbf{p}_{\mathbf{i}}) \in \mathbb{R}$, the true partial derivative values of F at the concrete point $\mathbf{p}_{\mathbf{i}}$.

[Lifting of Structural Functions] For each $g \in \mathcal{G}$, we interpret it as a concrete, well-defined real function $\widetilde{g}$ (e.g., $+$ interpreted as real addition, exp as real exponential). For expressions in $\mathcal{R}$ constructed from basic operations, we recursively define their valuation: $\iota_F(g(E_1, \ldots, E_n)) = \widetilde{g}(\iota_F(E_1), \ldots, \iota_F(E_n))$.

Crucial Step: Handling Formal Limits. This is the core of the construction and fundamentally differs from the polynomial case. For a formal limit $\xi = \lim_{n\to\infty} E_n$ in $\widehat{\mathcal{R}}$, we cannot unconditionally define its value as $\lim_{n\to\infty} \iota_F(E_n)$, because this limit may not exist or may depend on the choice of the expression sequence E_n.

We introduce adaptive grid refinement and consistency conditions.

[Adaptive Grid Sequence] For each $l \in \mathbb{N}$, construct a grid precision parameter N_l, corresponding to a set of sample points $\{\mathbf{p}_{\mathbf{i}}^{(l)}\}$ and their differential data $\{\partial^\alpha F(\mathbf{p}_{\mathbf{i}}^{(l)})\}$. Require that as $l \to \infty$, $N_l \to \infty$ and the grid becomes dense in K.

[Consistent Specialization Homomorphism] A specialization homomorphism $\iota_F : \mathcal{K}_{\text{Fun}} \to \mathcal{L}_F$ is called *consistent* if for every formal limit $\xi = \lim_{n\to\infty} E_n$ in $\widehat{\mathcal{R}}$, there exists an adaptive grid refinement sequence $\{N_l\}$ such that the numerical sequence $\{\iota_F^{(l)}(E_n)\}_n$ (where $\iota_F^{(l)}$ uses the data from the lth layer grid) converges under some joint limit as $n, l \to \infty$, and this limit value does not depend on the choice of the grid refinement sequence (modulo equivalence in the sense of a null set).

[Existence of the Specialization Homomorphism] For any $F \in C^r(K)$, there exists a consistent specialization homomorphism ι_F, taking its codomain $\mathcal{L}_F$ to be some function field containing $\mathbb{R}$ and the variables $\mathbf{x}$.

Proof Sketch. Choose the grid as a uniform grid. For formal limits, we require the expression sequence $\{E_n\}$ itself to represent a valid numerical approximation scheme (e.g., based on spline interpolation or wavelet expansion). Using the smoothness of F, one can prove that the numerical sequence defined by $\iota_F^{(l)}(E_n)$ is a Cauchy sequence. Taking its limit defines $\iota_F(\xi)$. All formal limits satisfying this "effective scheme" condition form a subfield, whose homomorphic image is $\mathcal{L}_F$. Consistency is guaranteed by standard results from approximation theory for C^r functions. □

5.4 Explicit Analytic Representation Formula for Multivariate Continuous Functions

In $\mathcal{K}_{\text{Fun}}$, we can construct a specific universal expression that becomes the core of our explicit representation formula.

[Universal Representation Kernel] Using the distance function d and exponential function from the structural function library, we define formal expressions. First, define the formal "Taylor kernel" at key point $\mathbf{P_i}$:

$$\mathcal{T}_{\mathbf{i}}(\mathbf{X}) := \sum_{|\alpha|\leq r} \frac{\Phi_{\mathbf{i},\alpha}}{\alpha!}(\mathbf{X}-\mathbf{P_i})^{\alpha}.$$

Here $(\mathbf{X}-\mathbf{P_i})^{\alpha} = \prod_{k=1}^{m}(X_k - P_{\mathbf{i}}^k)^{\alpha_k}$, and $\alpha! = \prod \alpha_k!$.

Second, define the formal weight function. Let $\psi : \mathbb{R} \to \mathbb{R}$ be a predetermined rapidly decaying smooth function (e.g., Gaussian), $\psi \in \mathcal{G}$. Define the weight at point $\mathbf{P_i}$ as:

$$\mathcal{W}_{\mathbf{i}}(\mathbf{X}) := \psi\left(\frac{d(\mathbf{X},\mathbf{P_i})}{\sigma}\right).$$

where $\sigma > 0$ is a formal scale parameter.

[Universal Moving Least Squares Expression] In $\mathcal{K}_{\text{Fun}}$, we define the following universal element:

$$\mathcal{F}_{\text{MLS}}(\mathbf{X}) := \frac{\sum_{\mathbf{i}} \mathcal{W}_{\mathbf{i}}(\mathbf{X}) \cdot \mathcal{T}_{\mathbf{i}}(\mathbf{X})}{\sum_{\mathbf{i}} \mathcal{W}_{\mathbf{i}}(\mathbf{X})}.$$

This is a formal rational expression, whose numerator and denominator are combinations of generators via structural functions.

[Explicit Analytic Representation Theorem] Let $F \in C^r(K)$, ι_F be a consistent specialization homomorphism. Let $\{N_l\}$ be an adaptive grid refinement sequence, and let the scale parameter sequence $\{\sigma_l\}$ satisfy $\sigma_l \to 0$ and $N_l\sigma_l^m \to \infty$. Then for the expression $\iota_F^{(l)}(\mathcal{F}_{\text{MLS}})$ obtained by specializing $\mathcal{F}_{\text{MLS}}$ via grid N_l, we have:

1. **Explicitness**: For each fixed l, $\iota_F^{(l)}(\mathcal{F}_{\text{MLS}})(\mathbf{x})$ is an explicit expression consisting of finitely many ($O(N_l^m)$) terms, involving only the variables $\mathbf{x}$, constants, arithmetic operations, exponential functions, and distance operations.

2. **Convergence**: As $l \to \infty$, the function sequence $\iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})$ converges uniformly to the original function F on K.

3. **Error Estimate**: There exists a constant C (depending on the bound of the $(r+1)$-th order derivatives of F and ψ) such that

$$\sup_{\mathbf{x}\in K} |\iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})(\mathbf{x}) - F(\mathbf{x})| \leq C \cdot (\sigma_l)^{r+1}.$$

4. **Binary Composability**: The computation process of each $\iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})$ can be organized as a computation graph where each node has at most two operands. Therefore, for each l, the map $\mathbf{x} \mapsto \iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})(\mathbf{x})$ is a composition of finitely many binary/unary functions.

Complete Proof. **Step 1 (Explicitness of Expression):** For fixed l, the number of $\mathbf{i}$ is finite: N_l^m. $\iota_F^{(l)}(\mathcal{T}_{\mathbf{i}}(\mathbf{X}))$ becomes a multivariate polynomial of degree r in $(\mathbf{x} - \mathbf{p}_{\mathbf{i}}^{(l)})$, with coefficients being the specific partial derivatives $\partial^\alpha F(\mathbf{p}_{\mathbf{i}}^{(l)})$. $\iota_F^{(l)}(\mathcal{W}_{\mathbf{i}}(\mathbf{X}))$ becomes a concrete decay function $\psi(\|\mathbf{x}-\mathbf{p}_{\mathbf{i}}^{(l)}\|/\sigma_l)$. Therefore, the entire $\iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})(\mathbf{x})$ is an explicit rational function combination with finitely many terms. Each step of its computation (polynomial evaluation, distance computation, exponential function, addition, multiplication, division) is a binary or unary operation.

Step 2 (Proof of Uniform Convergence): This is a standard theoretical result of Moving Least Squares (MLS) approximation [11, 12]. The key point is that our specialization process concretizes the formal expression into an MLS approximator. For C^r functions, the MLS approximation with support radius σ_l possesses local polynomial reproducibility. Using the rapid decay of ψ and the denseness of the grid, one can prove that for any $\mathbf{x} \in K$, the denominator $\sum_{\mathbf{i}} \iota_F^{(l)}(\mathcal{W}_{\mathbf{i}})(\mathbf{x})$ has a positive lower bound, and the numerator is a weighted average of local Taylor expansions of F. By estimating the remainder term via Taylor's theorem, the pointwise error is $O((\sigma_l)^{r+1})$. Since $\sigma_l \to 0$ and the estimate is uniform on K, uniform convergence follows.

Step 3 (Decomposition into Binary Function Composition): For each fixed approximation level l, the algorithm computing $\iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})(\mathbf{x})$ can be organized as a computation graph:

- Leaf nodes: input variables $x_1, \ldots, x_m$, constants (grid point coordinates $\mathbf{p}_{\mathbf{i}}$, derivative values $\partial^\alpha F(\mathbf{p}_{\mathbf{i}})$, parameter σ_l).

- Internal nodes: binary operations (addition, subtraction, multiplication), unary operations (computing distance $\|\mathbf{x}-\mathbf{p}_{\mathbf{i}}\|$, computing $\psi(\cdot)$, possibly involving exponentiation, powers, etc.).

- Final output: obtained via a series of binary additions and one binary division (numerator divided by denominator).

In this computation graph, each node has at most two operands. Therefore, for each l, the map $\mathbf{x} \mapsto \iota_F^{(l)}(\mathcal{F}_{\mathrm{MLS}})(\mathbf{x})$ is a composition of finitely many binary/unary functions. As $l \to \infty$, we obtain a sequence of such functions whose limit is F. □

Any C^r function F on a compact set K can be represented as the limit of a sequence of functions, each of which is a composition of finitely many binary elementary functions (from the library $\widetilde{\mathcal{G}}$). This provides a constructive, geometrically and differentially explicit approximative version of the Kolmogorov–Arnold theorem.

5.5 Correspondence with the Four-Step Program and Differential Galois Theory Perspective

Our construction clearly embodies the four steps of the "Constructive-Analytic Grand Unification":

1. **Closure Relativity**: We extended the closure for function representation from classical polynomial or Fourier bases to the closure $\mathcal{K}_{\mathrm{Fun}}$ generated from key-point differential data via structural functions. Within this closure, F obtains an explicit representation $\mathcal{F}_{\mathrm{MLS}}$.

2. **Key-Point Differential Forms**: The generators $\Phi_{\mathbf{i},\alpha} = \partial^\alpha \mathcal{F}(\mathbf{P_i})$ directly correspond to this.

3. **Stratified Closure Construction**: The clear layers: from the coordinate ring R_0, to the differential data ring R_1, to the structural function closure $\mathcal{R}$, finally to $\widehat{\mathcal{R}}$ containing formal limits and $\mathcal{K}_{\mathrm{Fun}}$.

4. **Symmetry-Driven Representation**: The design of the representation form $\mathcal{F}_{\mathrm{MLS}}$ embodies translation covariance (through weights centered at $\mathbf{X}$) and rotational invariance (if the distance function d is Euclidean).

From the perspective of differential Galois theory, the differential automorphism group $\mathrm{Aut}^{\mathrm{diff}}(\mathcal{K}_{\mathrm{Fun}}/\mathbb{F}(\{P_{\mathbf{i}}^j\}))$ of the universal closure $\mathcal{K}_{\mathrm{Fun}}$ is very large. However, the choice of a specific representation form (like MLS) corresponds to fixing a representation of a solvable subgroup of this group. This subgroup acts on the generators $\Phi_{\mathbf{i},\alpha}$, preserving the linear constraint relations defined by the MLS interpolation scheme (i.e., at each point, $\Phi_{\mathbf{i},\alpha}$ should come from the derivatives of the same function F). These constraint relations form a differential ideal in the differential Galois theory, and the MLS expression is an invariant in the quotient ring of this ideal.

6 Explicit Analytic Representation of High-Power Multivariate Continuous Functions: A Complete Theory with Combinatorial Correction Terms and Branch Selection

This section establishes a complete explicit analytic representation theory for multivariate continuous functions with high-order power nonlinearities. Building upon the basic moving least squares framework, we systematically handle high-power interaction effects by introducing combinatorial correction terms and develop a set of branch selection formulas based on algebraic topology to resolve multivaluedness issues.

6.1 Problem Background and Challenges

For multivariate functions $F : K \subset \mathbb{R}^m \to \mathbb{R}$ with high-power structures, such as those containing terms like $x_1^{k_1} x_2^{k_2} \cdots x_m^{k_m}$ with large $\sum k_i$, the basic moving least squares representation faces two core challenges:

1. **Challenge 1 (Combinatorial Correction Problem):** High-order interaction terms cannot be simply generated by tensor products of low-order Taylor expansions in individual variables. The r-th order Taylor expansion $T_{\mathbf{i}}(\mathbf{x}) = \sum_{|\alpha| \le r} \frac{\partial^\alpha F(\mathbf{p_i})}{\alpha!} (\mathbf{x} - \mathbf{p_i})^\alpha$ at key point $\mathbf{p_i}$ only contains terms with $|\alpha| \le r$. When the actual "effective nonlinearity degree" d_{eff} of F is $> r$, direct approximation will introduce systematic bias.

2. **Challenge 2 (Branch Selection Problem):** When the representation involves multivalued operations like taking high-order roots, inverse trigonometric functions, etc., globally consistent branch selection rules must be provided to ensure the resulting expression is a continuous single-valued function, not a union of multivalued relations.

This paper addresses these two challenges by systematically introducing combinatorial correction terms and branch selection formulas.

6.2 Construction of the Stratified-Corrected Functional-Differential Closure $\mathcal{K}^*_{\textbf{Fun}}$

We construct an extended closure $\mathcal{K}^*_{\text{Fun}}$ by adding new generators and relations to the basic closure $\mathcal{K}_{\text{Fun}}$.

6.2.1 Formal Generators for High-Power Interactions

For each key point $\mathbf{P_i}$, besides the standard partial derivative generators $\Phi_{\mathbf{i},\alpha}$ ($|\alpha| \le r$), we introduce high-power interaction generators.

https://doi.org/10.33774/coe-2026-pxlkw **ORCID:** https://orcid.org/0009-0003-6570-2812 Content not peer-reviewed by Cambridge University Press.

[Formal High-Power Deviation] For each multi-index β satisfying $r < |\beta| \leq R$ (where $R > r$ is a preset maximum handling order), introduce formal symbols:

$$\Psi_{\mathbf{i},\beta}.$$

Their intuitive meaning should be the component of the "higher-order deviation beyond the r-th order Taylor expansion" of the function $\mathcal{F}$ at point $\mathbf{P_i}$ in the direction $(\mathbf{X}-\mathbf{P_i})^\beta$.

To relate $\Psi_{\mathbf{i},\beta}$ to low-order data, we introduce formal recurrence relations.

[Formal Recurrence Relations] For each $|\beta| = r + 1$, we require the existence of a formal expression $E_\beta \in \mathcal{K}_{\text{Fun}}$ (containing only $\Phi_{\mathbf{i},\alpha}$ with $|\alpha| \leq r$, and $\mathbf{P_i}, \mathbf{X}$) such that:

$$\Psi_{\mathbf{i},\beta} = \mathcal{H}_\beta(\{\Phi_{\mathbf{i},\alpha}\}_{|\alpha|\leq r}, \mathbf{X}, \mathbf{P_i}) \cdot (\mathbf{X} - \mathbf{P_i})^\beta + \Delta_{\mathbf{i},\beta},$$

where $\mathcal{H}_\beta$ is a specific formal function constructed from the structural function library, and $\Delta_{\mathbf{i},\beta}$ is a formal remainder of higher order. For $|\beta| > r+1$, similar recurrence relations connect $\Psi_{\mathbf{i},\beta}$ with lower-order $\Psi_{\mathbf{i},\gamma}$ ($|\gamma| = |\beta| - 1$) and $\Phi_{\mathbf{i},\alpha}$.

6.2.2 Generative Syntax for Combinatorial Correction Terms

[Correction Tensor] Define the complete formal correction tensor at key point $\mathbf{P_i}$ as:

$$\mathcal{C}_\mathbf{i}(\mathbf{X}) := \sum_{r<|\beta|\leq R} \frac{\Psi_{\mathbf{i},\beta}}{\beta!} \cdot \mathcal{M}_\beta\left(\frac{\mathbf{X} - \mathbf{P_i}}{\sigma}\right).$$

Here $\mathcal{M}_\beta$ are predetermined correction kernel functions (belonging to the structural function library $\mathcal{G}$), typically taken as products of rapidly decaying radial functions and angular modulation functions, used to localize the correction effect. σ is a formal scale parameter.

[Universal Expression with Corrections] In $\mathcal{K}^*_{\text{Fun}}$, define the extended universal representation element as:

$$\mathcal{F}^*_{\text{MLS}}(\mathbf{X}) := \frac{\sum_\mathbf{i} \mathcal{W}_\mathbf{i}(\mathbf{X}) \cdot [\mathcal{T}_\mathbf{i}(\mathbf{X}) + \lambda \cdot \mathcal{C}_\mathbf{i}(\mathbf{X})]}{\sum_\mathbf{i} \mathcal{W}_\mathbf{i}(\mathbf{X})}.$$

Here λ is a formal coupling parameter controlling the strength of the correction term. $\mathcal{T}_\mathbf{i}(\mathbf{X})$ is the standard r-th order Taylor form, and $\mathcal{W}_\mathbf{i}(\mathbf{X})$ is the weight function.

6.2.3 Formalization of Branch Selection

To handle multivaluedness, we add branch selection generators to $\mathcal{K}^*_{\text{Fun}}$.

[Formal Path and Uniformization Generators] Suppose some expression $\mathcal{E}(\mathbf{X})$ in $\mathcal{K}^*_{\text{Fun}}$, upon specialization, may involve a multivalued function g (e.g., $z^{1/n}$, $\arcsin(z)$). We introduce:

https://doi.org/10.33774/coe-2026-pxlkw ORCID: https://orcid.org/0009-0003-6570-2812

- **Formal Path Variables**: For each operation that may produce branching, associate a formal symbol Γ_k, taking values in the fiber of the covering space of that operation (e.g., for the n-th root, $\Gamma_k \in \{0, 1, \ldots, n-1\}$).
- **Formal Consistency Condition Relations**: Add a system of formal relational equations that force the branch choices $\Gamma_k(\mathbf{X})$ at different points $\mathbf{X}$ to remain consistent under continuous variation. For example, for a closed loop γ, require:

$$\prod_{\text{along } \gamma} \Delta\Gamma_k = \text{Winding}(\gamma; \text{BranchPoints}),$$

where the right-hand side is the formal winding number computed from the formal branch point locations $\mathcal{B}_j$.

[Branch-Analytic Expression] Finally, any expression $\mathcal{E}$ involving multivalued operations is rewritten as:

$$\mathcal{E}_{\text{branch}}(\mathbf{X}) = \mathcal{E}_{\text{main}}(\mathbf{X}, \{\Gamma_k(\mathbf{X})\}) + \sum_j \mathcal{E}_{\text{corr}}^{(j)}(\mathbf{X}, \{\mathcal{B}_j\}, \{\Gamma_k(\mathbf{X})\}),$$

where $\mathcal{E}_{\text{main}}$ is the principal value expression after branch selection, and $\mathcal{E}_{\text{corr}}^{(j)}$ are analytic correction terms when crossing branch point $\mathcal{B}_j$.

6.3 Specialization Homomorphism and Concrete Implementation of Combinatorial Correction Terms

For a concrete function F, the specialization homomorphism ι_F must assign concrete values to $\Psi_{\mathbf{i},\beta}$ and branch choices Γ_k.

6.3.1 Actual Computation of High-Power Deviations

[Deviation Generation Algorithm] There exists an algorithm that, using only the partial derivative values $\partial^\alpha F(\mathbf{p_i})$ with $|\alpha| \leq r$ at key points $\mathbf{p_i}$, constructs approximate values $\widetilde{\Psi}_{\mathbf{i},\beta}$ such that for $|\beta| = r+1$, we have:

$$\left| \widetilde{\Psi}_{\mathbf{i},\beta} - \left[\partial^\beta F(\mathbf{p_i}) - \frac{d^\beta}{d\mathbf{x}^\beta} T_{\mathbf{i}}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{p_i}} \right] \right| \leq C_F \cdot h^{R-|\beta|+1},$$

where h is the grid step size, and C_F depends on the bound of the $(R+1)$-th order derivatives of F.

Derivation Process. 1. **Construct Difference Scheme**: In the local neighborhood of key point $\mathbf{p_i}$, select an auxiliary point set $\{\mathbf{q}_j\}$ satisfying $\|\mathbf{q}_j - \mathbf{p_i}\| = O(h)$.

https://doi.org/10.33774/coe-2026-pxlkw ORCID: https://orcid.org/0009-0003-6570-2812 Content not peer-reviewed by Cambridge University Press.

2. **Set Up Linear System**: Compute the values of F at these auxiliary points and approximate them using the r-th order Taylor expansion $T_{\mathbf{i}}(\mathbf{q}_j)$. Define residuals $R_j = F(\mathbf{q}_j) - T_{\mathbf{i}}(\mathbf{q}_j)$.

3. **Fit High-Order Terms**: Model the residuals R_j as contributions from high-order terms $\sum_{|\beta|=r+1}^{R} \frac{c_\beta}{\beta!}(\mathbf{q}_j - \mathbf{p}_{\mathbf{i}})^\beta$. Solve a least squares problem to obtain coefficient estimates $\widetilde{c}_\beta$.

4. **Assignment**: Set $\iota_F(\Psi_{\mathbf{i},\beta}) = \widetilde{c}_\beta$.

5. **Error Analysis**: By Taylor's theorem, the true residual contains terms of $O(h^{r+1})$. By selecting sufficiently many auxiliary points (more than the number of unknown coefficients), the linear system is solvable. The error estimate comes from the propagation of truncation and fitting errors.

□

6.3.2 Concrete Form of Correction Kernel Functions

We take $\mathcal{M}_\beta\left(\frac{\mathbf{x}-\mathbf{p}_{\mathbf{i}}}{\sigma}\right) = \psi_M\left(\frac{\|\mathbf{x}-\mathbf{p}_{\mathbf{i}}\|}{\sigma}\right) \cdot H_\beta\left(\frac{\mathbf{x}-\mathbf{p}_{\mathbf{i}}}{\sigma}\right)$, where:

- ψ_M is a compactly supported radial function with support radius 2σ.
- H_β is a homogeneous polynomial of degree $|\beta|$, satisfying orthogonality conditions: $\int_{\mathbb{R}^m} H_\beta(\mathbf{u})\mathbf{u}^\alpha \psi_M(\| \mathbf{u} \|)d\mathbf{u} = 0$ for all $|\alpha| \leq r$. This ensures that the correction term $\mathcal{C}_{\mathbf{i}}$ does not interfere with the low-order polynomial parts already exactly reproduced by $T_{\mathbf{i}}$.

6.4 Strict Derivation of Branch Selection Formulas

For multivalued operations appearing in the representation, we need a globally consistent, automated branch selection rule.

6.4.1 Algebraic Topological Formulation of the Branch Selection Problem

Let $U \subset K$ be a region in the function's domain avoiding all potential branch points. Potential branch points $\mathbf{b} \in \mathbb{C}^m$ (considered after complexification) are points where the radicand (or argument of other multivalued operations) becomes zero or singular.

[Branch Selection Cohomology Class] For each multivalued operation, define a locally constant function $\gamma : U \to \mathbb{Z}/n\mathbb{Z}$ (for the n-th root) or $\to \mathbb{Z}$ (for logarithm), specifying the chosen branch. This function on U is completely determined by its values on generating cycles.

[Existence of Consistent Branch Selection] If the domain K is simply connected and all potential branch points $\mathbf{b}_j$ are not on K, then there exists a consistent continuous branch selection function $\gamma(\mathbf{x})$.

Proof. In a simply connected region, any closed path can be continuously contracted to a point. Therefore, analytic continuation along any closed path does not change the function value, so the principal branch can be chosen consistently. If K is not simply connected, one needs to consider the representation of the fundamental group $\pi_1(K)$. □

6.4.2 Constructive Branch Selection Formula

We provide a constructive algorithm to compute $\gamma(\mathbf{x})$.

[Branch Selection Based on Reference Point] **Input**: Point $\mathbf{x} \in U$, a fixed reference point $\mathbf{x}_0 \in U$, and the chosen branch $\gamma(\mathbf{x}_0)$ at that point (usually the principal branch). **Steps**:

1. **Construct Path**: Choose a piecewise linear path L from $\mathbf{x}_0$ to $\mathbf{x}$ within U.

2. **Compute Winding Numbers**: For each branch hypersurface $\{\mathbf{z} : G_j(\mathbf{z}) = 0\}$ that the path L crosses, compute the number of sign changes or the amount of argument change.

3. **Determine Branch**: Update the branch according to the formula:

$$\gamma(\mathbf{x}) = \gamma(\mathbf{x}_0) + \sum_j \mathrm{Winding}_j(L) \cdot n_j \pmod{n},$$

   where $\mathrm{Winding}_j(L)$ is the winding number of path L around branch hypersurface j, and n_j is the branch jump induced by that hypersurface (e.g., for the n-th root, $n_j = 1$).
   Output: $\gamma(\mathbf{x})$.

[Well-Definedness of the Formula] If U is simply connected, then the result $\gamma(\mathbf{x})$ of Algorithm 6.4.2 is independent of the choice of path L. If U is not simply connected, the result is unique modulo the representation of $\pi_1(U)$.

Proof. Consider two different paths L_1 and L_2 from $\mathbf{x}_0$ to $\mathbf{x}$. They form a closed loop $C = L_1 - L_2$. In a simply connected region, C is homotopic to zero, hence the total winding number along C is zero, so both paths yield the same $\gamma(\mathbf{x})$. For non-simply connected regions, different paths may belong to different homotopy classes, causing $\gamma(\mathbf{x})$ to differ by a global constant determined by $\pi_1(U)$. □

6.4.3 Concrete Form of Branch Correction Terms

When a path must cross a branch hypersurface, the expression needs correction terms to maintain continuity.

[Correction for Crossing an n-th Root Branch] Let $g(\mathbf{x}) = [f(\mathbf{x})]^{1/n}$, f vanishes on some hypersurface S. When the path crosses S, the principal branch $[f(\mathbf{x})]^{1/n}_{\text{main}}$ undergoes a phase jump of $\exp(2\pi i/n)$. To obtain a continuous function, we define:

$$g_{\text{cont}}(\mathbf{x}) = [f(\mathbf{x})]^{1/n}_{\text{main}} \cdot \exp\left(\frac{2\pi i}{n} \cdot H(\phi(\mathbf{x}) - \phi_0)\right),$$

where $\phi(\mathbf{x})$ is the argument of $f(\mathbf{x})$, ϕ_0 is a reference argument on S, and H is the Heaviside step function. The exponential factor here is the specialization of the branch correction term $\mathcal{E}_{\text{corr}}$.

6.5 Main Theorem and Convergence Proof

[Explicit Representation Theorem for High-Power Multivariate Functions] Let $F \in C^{R+1}(K)$, K be a simply connected compact set. Let $\{N_l\}$ be an adaptive grid refinement sequence, $h_l = \max$ grid step size, σ_l be scale parameters, satisfying $h_l/\sigma_l \to 0$, $\sigma_l \to 0$, and $N_l \sigma_l^m \to \infty$. Let $\iota_F^{(l)}$ be consistent specialization homomorphisms applying the above combinatorial correction algorithm and branch selection algorithm. Then the following conclusions hold:

1. **Explicitness**: For each l, the function $\widehat{F}_l(\mathbf{x}) := \iota_F^{(l)}(\mathcal{F}^*_{\text{MLS}})(\mathbf{x})$ is an explicit expression consisting of $O(N_l^m)$ terms, each involving only finitely many binary arithmetic operations, elementary functions, and branch selection functions $\gamma_k(\mathbf{x})$ determined by Algorithm 6.4.2.

2. **Consistency**: The branch selection functions $\gamma_k(\mathbf{x})$ ensure that $\widehat{F}_l$ is continuous and single-valued on the entire K.

3. **Convergence and Error Estimate**: There exists a constant C depending on the bound of the $(R+1)$-th order derivatives of F, such that

$$\sup_{\mathbf{x}\in K} |\widehat{F}_l(\mathbf{x}) - F(\mathbf{x})| \le C \cdot \left(\sigma_l^{r+1} + \frac{h_l^{R+1}}{\sigma_l^{R-r+1}}\right).$$

   In particular, if we choose $\sigma_l = h_l^\alpha$ with $\alpha = \frac{R+1}{R+2}$, then the convergence order is $O(h_l^{\alpha(r+1)}) = O\left(h_l^{\frac{(r+1)(R+1)}{R+2}}\right)$, which is better than the basic MLS order $O(\sigma_l^{r+1})$.

4. **Binary Composability**: The computation process of each $\widehat{F}_l$ can be organized as a computation graph where each node performs an operation (including the computation of branch selection functions) with at most 2 input variables. Therefore, $\widehat{F}_l$ is a composition of finitely many binary functions.

Complete Proof. **Part 1 (Explicitness and Binary Composition):** By construction, the expression for $\widehat{F}_l$ consists of finite sums, products, quotients, and elementary functions. The computation of branch selection functions $\gamma_k(\mathbf{x})$ (Algorithm 6.4.2) involves: computing intersections of the path with hypersurfaces (solving equations, approximated by judging sign changes of functions), computing winding numbers (integer arithmetic). All these operations can be decomposed into binary comparisons and arithmetic operations. Therefore, the overall expression is binary composable.

Part 2 (Consistency): By Theorem 6.9 and the construction of Algorithm 6.4.2, under the simply connected assumption, the branch selections $\gamma_k(\mathbf{x})$ are well-defined and continuous. Therefore, the composed $\widehat{F}_l$ is continuous and single-valued.

Part 3 (Error Estimate, Core): Decompose the error into three parts:

$$\widehat{F}_l(\mathbf{x}) - F(\mathbf{x}) = \underbrace{\left(\widehat{F}_l(\mathbf{x}) - F_l^*(\mathbf{x})\right)}_{\text{Branch \& Discretization Error}} + \underbrace{\left(F_l^*(\mathbf{x}) - F_l^{\text{ideal}}(\mathbf{x})\right)}_{\text{Correction Term Fitting Error}} + \underbrace{\left(F_l^{\text{ideal}}(\mathbf{x}) - F(\mathbf{x})\right)}_{\text{Ideal Approximation Error}}.$$

- **Ideal Approximation Error**: $F_l^{\text{ideal}}(\mathbf{x})$ is the ideal corrected MLS approximator constructed using the true high-order partial derivatives $\partial^\beta F$ of F. By MLS theory [11, 12], this error term is $O(\sigma_l^{r+1})$.

- **Correction Term Fitting Error**: By Theorem 6.7, we use estimated values $\widetilde{\Psi}_{\mathbf{i},\beta}$ instead of true high-order derivatives. This estimation error is $O(h_l^{R-|\beta|+1})$. In the MLS weighted average, this error is amplified by a factor of $\sigma_l^{-|\beta|}$ (because the correction kernel $\mathcal{M}_\beta$ scales as $\sigma^{-|\beta|}$). Therefore, the contribution of this error is $O\left(\frac{h_l^{R+1}}{\sigma_l^{R+1}} \cdot \sigma_l^{-r}\right) = O\left(\frac{h_l^{R+1}}{\sigma_l^{R-r+1}}\right)$. By choosing the relationship between σ_l and h_l, this term can be optimized.

- **Branch and Discretization Error**: Branch selection may introduce $O(1)$ phase jumps, but via correction terms $\mathcal{E}_{\text{corr}}$ (Example 6.4.3), they can be smoothed into errors of order $O(\sigma_l)$. Discretization error from approximating integrals with finite sums is $O(h_l/\sigma_l)$.

Combining the three terms, the dominant terms are $O(\sigma_l^{r+1} + \frac{h_l^{R+1}}{\sigma_l^{R-r+1}})$. Balancing them, i.e., $\sigma_l^{r+1} \sim \frac{h_l^{R+1}}{\sigma_l^{R-r+1}}$, yields $\sigma_l \sim h_l^{\frac{R+1}{R+2}}$. Substituting back gives the overall error order $O(h_l^{\frac{(r+1)(R+1)}{R+2}})$. When $R > r$, this order is higher than the basic MLS order $O(h_l^{r+1})$, demonstrating the benefit of high-power correction. □

7 Conclusion and Outlook: Towards a Unifying Paradigm for Constructive Analysis

This paper first thoroughly solved the algebraic variant of Hilbert's 13th Problem, providing an explicit differential algebraic solution for the root-finding map of septic and arbitrary-degree polynomial equations that can be decomposed into compositions of bivariate functions. The success of this concrete solution stems from a deeper paradigm: by extending the closure of mathematical operations, transforming classically unsolvable problems into constructible problems within a new framework.

We have distilled and systematized this paradigm into the "four-step" methodology under the "Constructive-Analytic Grand Unification" program and successfully generalized it to the representation problem of multivariate continuous functions. The established differential Galois theory provides a rigorous characterization of symmetry and solvability for the entire framework. The following table summarizes the contrast between the old and new paradigms:

Table 1: Comparison between Classical and Constructive Paradigms

Aspect	**Classical Galois Theory/Function Theory Paradigm**	**Constructive-Analytic Unification Program Paradigm**
Philosophical Core	Explore the boundaries of problems within a fixed operational language (e.g., radicals, continuous functions).	Extend the operational language itself, designing the most expressive closure for specific problem families.
Core Tools	Group theory, topology, existence proofs.	Differential algebra, closure construction, specialization homomorphism, constructive algorithms.
Treatment of "Unsovability"	Viewed as an absolute property of the problem (e.g., quintics unsolvable by radicals).	Viewed as a relative limitation of the closure used; seek solutions through closure extension.
Nature of Representation	Possibly non-constructive (K-A theorem), complex, hard to compute.	Pursues explicit, stratified, constructive representations with algebraic structure.
Unifying Power	Different problems (equation solving, function representation) use disparate tools.	Provides a unified methodological flow (four-step method) and theoretical foundation (differential Galois theory) for different problems.

At the concrete implementation level, we have:

1. Constructed the functional-differential closure $\mathcal{K}_{\text{Fun}}$ and the measure-theoretic specialization homomorphism ι_F for general multivariate continuous functions, and

provided an explicit analytic representation formula based on moving least squares, proving its uniform convergence and decomposability as a limit of binary function compositions.

2. For high-power nonlinear functions, further constructed the stratified corrected closure $\mathcal{K}^*_{\text{Fun}}$, introduced combinatorial correction terms to handle high-order interaction effects, and established branch selection formulas based on algebraic topology to resolve multivaluedness, obtaining improved approximation orders.

The work in this paper is merely the starting point of this grand program. Future research will delve deeper in several directions:

- **Theoretical Rigorization**: Establish the most rigorous category-theoretic and differential algebraic foundations for the functional-differential closure $\mathcal{K}_{\text{Fun}}$ and its differential Galois group.
- **Algorithmic Implementation**: Transform the four-step method into computable algorithms, study computational issues like key-point selection and structural function library optimization, and build bridges with neural network architectures in deep learning.
- **Tackling Major Problems**: Apply this framework along the directions indicated by the program [1] to major challenges like Diophantine equations, Navier–Stokes equations, etc., testing its power.
- **Foundational Reflection**: Deeply explore the philosophical and metamathematical significance of "closure extension" and its impact on mathematical practice and cognition.

The solution to Hilbert's 13th Problem, like a stone thrown into a lake, is sending ripples that are expanding into a completely new paradigm of mathematical research. It invites us not to be satisfied with discovering boundaries within established rules of the game, but to dare to become the makers of new rules, dissolving those once seemingly insurmountable boundaries through constructive language extension. This is perhaps the deeper expectation Hilbert harbored when posing that famous problem.

References

[1] Author(s). *The Constructive-Analytic Grand Unification in Mathematics: A Unifying Program Based on Closure Extensions*. Zenodo, 2026. https://doi.org/10.5281/zenodo.18140187.

[2] Kolmogorov, A. N. *On the representation of continuous functions of several variables by superpositions of continuous functions of fewer variables*. Dokl. Akad. Nauk SSSR, 1957, 108: 179–182.

[3] Related Preprint. *Differential Algebraic Closure Solution of Polynomial Equations: A Constructive Framework Transcending the Abel-Ruffini Barrier*. Cambridge Open Engage, 2026.

[4] Hilbert D. *Mathematical Problems*. Bulletin of the American Mathematical Society, 1902, 8(10): 437–479.

[5] Abel N H. *Mémoire sur les équations algébriques, où l'on démontre l'impossibilité de la résolution de l'équation générale du cinquième degré*. 1824.

[6] Ritt J F. *Differential Algebra*. New York: American Mathematical Society, 1950.

[7] Kaplansky I. *An Introduction to Differential Algebra*. Paris: Hermann, 1957.

[8] Arnold V I. *On functions of three variables*. Proceedings of the USSR Academy of Sciences, 1963, 114: 679–681.

[9] Umemura H. *Resolution of algebraic equations by theta constants*. In: Tata Lectures on Theta. Boston: Birkhäuser, 1984: 261–270.

[10] Pillay A. *Differential Galois Theory I*. Illinois Journal of Mathematics, 1997, 41(4): 678–699.

[11] Levin D. *The approximation power of moving least-squares*. Mathematics of Computation, 1998, 67(224): 1517–1531.

[12] Wendland H. *Scattered Data Approximation*. Cambridge: Cambridge University Press, 2005.

[13] Kolmogorov A N, Tikhomirov V M. *ε-entropy and ε-capacity of sets in function spaces*. American Mathematical Society Translations, 1961, 17: 277–364.

[14] De Boor C. *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.

[15] Whitney H. *Analytic extensions of differentiable functions defined in closed sets*. Transactions of the American Mathematical Society, 1934, 36(1): 63–89.

[16] Hatcher A. *Algebraic Topology*. Cambridge: Cambridge University Press, 2002.

[17] Bürgisser P, Clausen M, Shokrollahi M A. *Algebraic Complexity Theory*. Berlin: Springer, 1997.

[18] Pinkus A. *Approximation theory of the MLP model in neural networks.* Acta Numerica, 1999, 8: 143–195.

[19] Chen Wenshan, Sun Jiachang. *Moving Least Squares Method for Scattered Data Fitting.* Journal of Computational Mathematics, 2003, 25(3): 263–272. (in Chinese)

[20] Ding Tongren. *Galois Theory.* Beijing: Higher Education Press, 2010. (in Chinese)

[21] van der Waerden B L. *Algebra, Volume I & II.* New York: Springer-Verlag, 1991.

[22] Related Preprint. *Differential-Algebraic Extension Field Solutions for Polynomial Equations with Complex Coefficients: A Comprehensive Extension.* Cambridge Open Engage, 2026.

[23] Related Preprint. *Differential Algebraic Closure Solution of Polynomial Equations and Its Equivalence to Transcendental Function Representations.* Cambridge Open Engage, 2026.