Geofinitism: Finite Symbolic Mechanics and the Foundations of Finite Analysis

Kevin R. Haylett, PhD
Manchester, United Kingdom

Version 1.0

June 2026

Contents

Abstract

This work presents a foundational re-examination of linear algebra from the perspective of Geofinitism, a measurement-first philosophical framework that treats all mathematical objects as finite symbolic compressions of underlying sequential processes. The central thesis is that matrices are not primary objects but compressed instruction sets for finite sequences of arithmetic operations. When executed, these operations generate genuine time series of calculations—functional symbolic trajectories whose geometry can be examined directly.

Within this view, eigenvalues and eigenvectors are reinterpreted as stabilized descriptors of these calculation trajectories. They are not pre-existing Platonic properties but invariants that emerge after sufficient sequential operations have been performed. The QR algorithm, applied to an LRC circuit state matrix, serves as a concrete demonstration: the unfolding of matrix transformations produces a finite trajectory whose diagonal entries stabilize at the eigenvalues after approximately ten iterations.

The framework extends naturally to algebraic unsolvability. The impossibility of solving general quintic equations by radicals is reframed as the inability of any finite sequence of permitted operations (additions, multiplications, and root extractions) to navigate the irreducible nonlinear dynamics of the required calculation trajectory. This geometric, process-based explanation complements classical Galois theory.

Finally, by applying delay embedding (Takens' method) to computational sequences, we demonstrate that the same phase-space techniques used to analyze physical systems can be applied to the calculation processes themselves. This closes a powerful methodological loop: physical systems and the computational procedures used to study them are treated as unified finite trajectories amenable to the same geometric tools. The work connects to broader developments in Finite Symbolic Mechanics and establishes linear algebra as a case study in the Geofinitist reconstruction of mathematics.

Chapter 1

The Unfolded Calculation Trajectory of Linear Systems: Eigenvalues and Eigenvectors as Descriptors of Finite Sequential Processes

1.1 Introduction and Motivation

Conventional treatments of linear algebra present matrices, eigenvalues, and eigenvectors as the primary objects of study. Eigenvalues are described as scaling factors and eigenvectors as preferred directions. In low-dimensional cases this narrative feels intuitive and geometric. However, as systems grow in size or complexity, the narrative collapses into analogy. The matrix formalism becomes a highly compressed shorthand that hides the actual work being performed.

In the framework of **Geofinitism**, we insist on remaining with what is finite, sequential, and directly traceable. This philosophical position, developed in the broader Geofinitism programme, rests on five core commitments:

1. **Geometric Container**: Every symbol occupies measurable volume and carries uncertainty.

2. **Approximations/Measurements**: All mathematical objects are finite, measurement-conditioned constructions.

3. **Dynamic Flow**: Processes and trajectories are primary; static objects are compressions.

4. **Useful Fiction**: Mathematical abstractions are valuable but must be recognized as fictions.

5. **Finite Reality**: Only finite, sequentially traceable constructions are admissible as foundational.

When matrix operations are carried out, they consist of long chains of elemental arithmetic steps—multiplications, additions, and subtractions—executed one after another in time. These steps form a genuine **time series of calculations**. The matrix notation is convenient, but it obscures this underlying sequential process. The concept of **Finite**

Process Unfolding (FPU) formalizes the recovery of this sequential structure from compressed symbolic forms.

This chapter develops the view that the **unfolded sequence of explicit calculations** is the primary object. Eigenvalues and eigenvectors are not fundamental entities that exist independently; they are **descriptors**—invariants and directional summaries—that characterise properties of that larger finite trajectory once it has been sufficiently executed. Within Geofinitism, this is understood as a process of **stabilization**: the emergence of stable, repeatable features from finite dynamical processes.

1.2 Matrices as Compressed Representations of Sequential Operations

A matrix equation such as $y = Ax$ is, at root, a compact instruction set. Expanding the multiplication for even a modest 2×2 matrix reveals four multiplications and two additions that must be performed in a definite order. For larger matrices the number of elemental operations grows rapidly, yet all remain strictly sequential and finite.

1.2.1 Formalizing the Calculation Trajectory

We define the following concepts to make the discussion precise:

Definition 1.1 (Calculation State). *A calculation state S_t at time step t is the tuple of all intermediate numerical values present at that moment in the computation:*

$$S_t = \big(v_1(t),\ v_2(t),\ \ldots,\ v_{n_t}(t)\big)$$

where n_t is the number of active values.

Definition 1.2 (Elemental Operation). *An elemental operation is a binary operation $o : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ drawn from the set $\{+, -, \times, \div\}$, applied to two existing values in the calculation state to produce a new value.*

Definition 1.3 (Calculation Trajectory). *A calculation trajectory is the finite sequence of states generated by applying a deterministic sequence of elemental operations:*

$$\Gamma = S_0 \xrightarrow{o_1} S_1 \xrightarrow{o_2} S_2 \xrightarrow{o_3} \cdots \xrightarrow{o_T} S_T$$

where S_0 is the initial state (containing the input values) and S_T is the final state (containing the results).

Definition 1.4 (Functional Symbolic Trajectory). *Following the framework of Functional Symbolic Trajectories, a functional symbolic trajectory is a finite symbolic movement that carries usable structure through context, under constraint, with uncertainty. For a symbol s, its trajectory is:*

$$\gamma_s = T(s;\ C,\ H_s,\ R,\ \alpha,\ \delta)$$

where C is context, H_s is history, R is reconstruction operator, α is resolution, and δ is uncertainty bound.

The compressed matrix view allows rapid manipulation and elegant theorems. It does not, however, make visible the actual time series of calculations that any physical or digital processor must execute. In Geofinitism we choose to keep the sequential nature explicit whenever deeper insight or verification is required.

1.3 The LRC Circuit: From Physical Model to State Matrix

Consider the series LRC circuit governed by Kirchhoff's voltage law. Choosing inductor current I_L and capacitor voltage V_C as states yields the linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$

where

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}, \qquad x = \begin{bmatrix} I_L \\ V_C \end{bmatrix}.$$

The matrix A is a convenient encapsulation of the coupled differential equations. To analyse stability, compute eigenvalues, or simulate the system, one must ultimately perform sequences of arithmetic operations on the numerical entries of A (or on discretised versions of the dynamics). The matrix itself does not perform these operations; it merely organises the instructions.

Within the Geofinitism framework, this physical system exemplifies the relationship between measurement, symbol generation, and computation. The state variables I_L and V_C are themselves **measured quantities** with finite resolution and uncertainty. The matrix A is a **symbolic compression** of the differential equations, and the eigenvalue computation is a **finite process** that generates stabilised descriptors from this compression.

1.4 The QR Algorithm as an Explicit Finite Time Series of Calculations

The standard method for extracting eigenvalues of a general matrix is the QR algorithm. Rather than solving the characteristic polynomial (which becomes impossible in radicals for degree ≥ 5), the algorithm generates a sequence of similar matrices through repeated, fully explicit transformations.

Start with $A_0 = A$. At each step k:

1. Compute the QR factorisation $A_k = Q_k R_k$ (itself a finite sequence of Householder or Givens operations).
2. Form the next matrix $A_{k+1} = R_k Q_k$.

Each iteration is a finite block of arithmetic. The entire process is therefore a discrete time series:

$$A_0 \to A_1 \to A_2 \to \cdots \to A_N.$$

All matrices in the sequence remain similar to the original A, so any invariants of similarity are preserved throughout the trajectory.

1.4.1 Algorithmic Specification

1.4.2 Concrete Unfolding — Overdamped LRC Example

Take $L = 1$, $C = 1$, $R = 3$ (overdamped case). The initial matrix is

$$A_0 = \begin{bmatrix} -3 & -1 \\ 1 & 0 \end{bmatrix}.$$

Algorithm 1 QR Algorithm for Eigenvalues

Input: Matrix $A \in \mathbb{R}^{n\times n}$, tolerance $\varepsilon > 0$, max iterations $N_{\max}$
Output: Approximate eigenvalues $\lambda_1, \dots, \lambda_n$
$A_0 \leftarrow A$
$k \leftarrow 0$
while $k < N_{\max}$ and $\max_{i>j} |(A_k)_{ij}| > \varepsilon$ **do**
    Compute QR decomposition: $A_k = Q_k R_k$ ▷ Using Householder reflections or Givens rotations
    Update: $A_{k+1} \leftarrow R_k Q_k$
    $k \leftarrow k + 1$
end while
Return: $\lambda_i \leftarrow (A_k)_{ii}$ for $i = 1, \dots, n$

The characteristic equation is

$$\det(A_0 - \lambda I) = \lambda^2 + 3\lambda + 1 = 0,$$

with eigenvalues

$$\lambda_1 = \frac{-3+\sqrt{5}}{2} \approx -0.38197, \qquad \lambda_2 = \frac{-3-\sqrt{5}}{2} \approx -2.61803.$$

After successive QR iterations the sequence evolves as follows (rounded for display):

$$A_1 \approx \begin{bmatrix} -2.700 & 1.900 \\ -0.100 & -0.300 \end{bmatrix},$$

$$A_5 \approx \begin{bmatrix} -2.61807 & 1.99996 \\ -0.00004 & -0.38193 \end{bmatrix},$$

$$A_{10} \approx \begin{bmatrix} -2.61803 & -0.000 \\ 0.000 & -0.38197 \end{bmatrix}.$$

By iteration 10 the sub-diagonal element has been driven to numerical zero. The diagonal entries have stabilised at the eigenvalues ≈ -2.618 and -0.382. These two numbers are **invariants** that remained constant for every matrix along the entire computational trajectory.

The process did not "discover" the eigenvalues by solving an equation in the abstract. It generated them as stable descriptors that become explicitly readable once the time series of matrix transformations has unfolded far enough.

1.5 Eigenvalues and Eigenvectors as Descriptors of the Calculation Trajectory

In this perspective:

- The **primary object** is the finite time series of explicit calculations (the unfolded trajectory in matrix space).
- **Eigenvalues** are the invariant scalars that characterise the whole trajectory. They are unchanged by every similarity transformation performed during the unfolding.

- **Eigenvectors** (or Schur vectors) describe the evolving coordinate system in which the trajectory simplifies. They indicate the directions along which the sequential operations progressively align the matrix.

1.5.1 Descriptor Terminology

We can distinguish three roles that eigenvalues play in the calculation trajectory:

Definition 1.5 (Invariant). *A quantity I is an invariant if $I(A_k) = I(A_0)$ for all k in the trajectory. Eigenvalues are invariants under similarity transformations.*

Definition 1.6 (Emergent Quantity). *A quantity E is emergent if it becomes visible only after a sufficient number of steps in the calculation trajectory. The diagonal entries of the QR iteration are emergent quantities that stabilise at the eigenvalues.*

Definition 1.7 (Descriptor). *A descriptor is a quantity that characterises the entire trajectory or its terminal behaviour. Eigenvalues function as descriptors in both the invariant and emergent senses.*

When the eigenvalues are complex, the trajectory in matrix space settles into 2×2 blocks rather than full diagonal form; the complex conjugate pair remains the invariant descriptor of that oscillatory computational behaviour.

This view restores geometric meaning even for large matrices: the eigenvalues and eigenvectors are not mysterious properties of an opaque object; they are measurable features of a long but finite sequence of arithmetic steps.

1.6 Embedding the Calculation Time Series in Phase Space

Because the unfolding is itself a time series, it can be treated as a dynamical signal. One may record the evolution of matrix entries, the norm of off-diagonal blocks, the trace, or any derived scalar quantity across iterations. This produces a new time series describing the **geometry of the computation itself**.

1.6.1 The Finite-Symbol Embedding Theorem

The application of delay embedding to finite symbolic sequences is justified by the Finite-Symbol Embedding Theorem (FSET), which extends Takens' classical theorem to discrete symbolic systems.

Theorem 1.1 (Finite-Symbol Embedding Theorem, Haylett 2025). *Let $\Gamma = \{s_t\}_{t=1}^{T}$ be a finite sequence of symbols from a finite alphabet S, generated by an underlying deterministic dynamical system $F : M \to M$ with measurement function $h : M \to S$. Under two non-degeneracy conditions:*

1. ***Observational Separation**: $h(x) \neq h(y)$ for distinct x, y in the relevant region of M,*

2. ***Finite Complexity**: The system has finite embedding dimension $m < \infty$,*

there exists an embedding dimension $d \leq 2m+1$ and delay τ such that the delay map

$$\Phi_{d,\tau}(s_t) = (s_t,\ s_{t-\tau},\ \ldots,\ s_{t-(d-1)\tau})$$

is injective up to resolution ε and geometrically stable under perturbations bounded by $\varepsilon/2$. The reconstructed phase portrait converges to the true attractor as $\varepsilon \to 0$.

Remark 1.1. *The FSET establishes that Takens' theorem is a limiting case of finite-symbol embedding:*

$$T = \lim_{\varepsilon \to 0} F_\varepsilon,$$

where T is the classical Takens embedding and F_ε is the finite-symbol embedding. This inverts the traditional proof hierarchy: finite representability is the ground; continuous smoothness is an idealisation.

1.6.2 Application to Computational Trajectories

Applying the FSET to a computational time series, we define the embedding:

Definition 1.8 (Computational Phase-Space Embedding)**.** *For a computational trajectory $\Gamma = (v_1, v_2, \ldots, v_T)$ of scalar values, the delay embedding with parameters d and τ is:*

$$\Phi_{d,\tau}(v_t) = (v_t,\ v_{t-\tau},\ \ldots,\ v_{t-(d-1)\tau}) \in \mathbb{R}^d.$$

The embedded trajectory $\{\Phi_{d,\tau}(v_t)\}_{t=1}^{T-(d-1)\tau}$ reconstructs the phase portrait of the computation.

Applying delay embedding (Takens' method) to this computational time series reconstructs a phase portrait of how the calculations unfold. In this higher-level phase space one can observe:

- The path taken by the sequence of matrices,
- The rate at which the trajectory approaches its canonical (Schur) form,
- Any transient or cyclic behaviour in the computational process.

The original eigenvalues appear in this portrait as invariants of the embedded attractor. Thus the same geometric tools used to analyse the physical LRC circuit can be applied directly to the calculation process that analyses the circuit.

This closes a powerful loop: physical systems and the computational procedures used to study them are both treated as finite trajectories amenable to the same phase-space techniques.

1.7 Finite Stability in Physical and Computational Domains

For the LRC circuit, finite stability requires that physical energy remains bounded (eigenvalues of A lie in the open left-half plane). In the unfolded view we add a second requirement: the **computational time series** that extracts those eigenvalues must itself remain well-behaved and convergent within a finite number of steps.

Definition 1.9 (Finite Stability). *A system is finitely stable if:*

1. *The physical trajectory remains within bounded measurement bounds.*

2. *The computational trajectory used to analyse it converges to its stabilised descriptors within a finite, predetermined number of steps.*

When parameters vary with time ($R(t)$, $C(t)$), the matrix $A(t)$ changes and a new finite unfolding trajectory must be generated at each instant. Stability of the overall system therefore encompasses both the physical trajectories and the computational trajectories that monitor them.

The visualisation below shows the classical eigenvalue locus in the complex plane as resistance varies—itself a higher-level descriptor of families of computational trajectories.

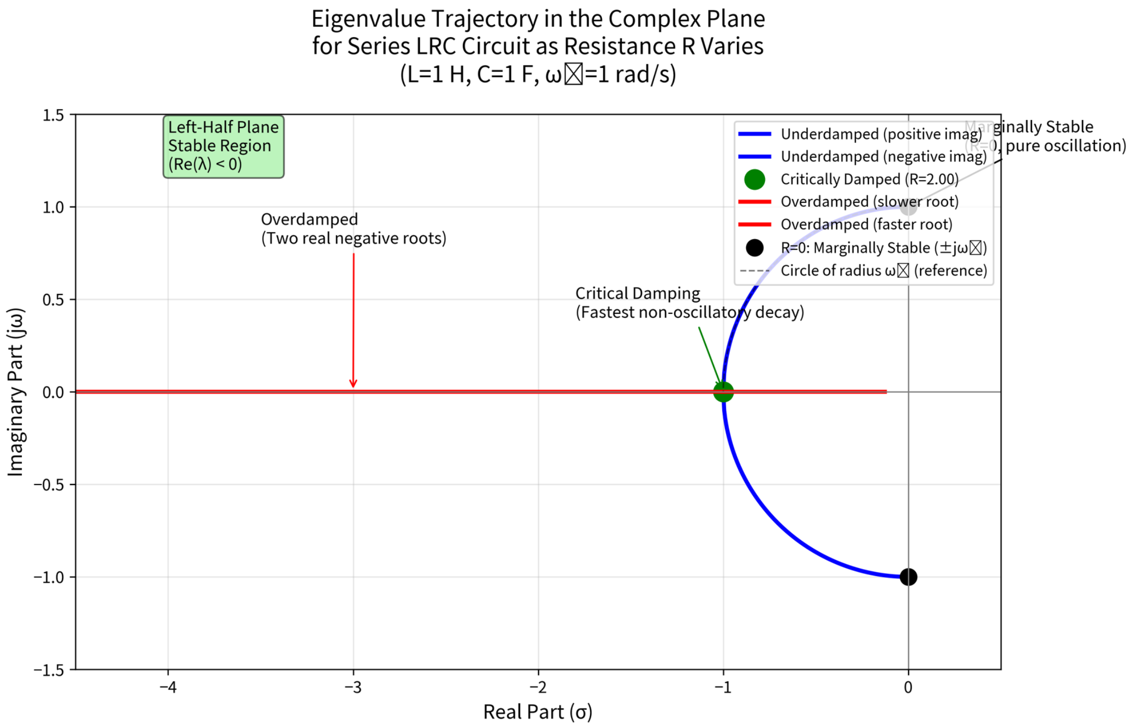

Figure 1.1: Eigenvalue trajectories in the complex plane for the series LRC circuit as resistance R varies from zero (marginally stable) through critical damping and into the overdamped regime. The circular arc for the underdamped case and the subsequent splitting along the negative real axis are invariants of the underlying families of calculation trajectories.

1.8 Conclusion and Transition to the Next Phase

By treating the explicit sequence of arithmetic operations as the primary finite trajectory, we obtain a consistent, non-analogical understanding of linear systems. Matrices remain useful shorthand, but they are no longer the final layer of description. Eigenvalues and eigenvectors regain concrete meaning as descriptors of how that sequential process unfolds and simplifies.

This perspective aligns directly with the core commitments of Geofinitism: every object of analysis remains finite, sequential, and, where possible, directly mappable into observable geometric structure via phase-space embedding.

The framework developed in this chapter—the primacy of the calculation time series, the descriptor status of eigenvalues/vectors, and the possibility of embedding computational trajectories—provides the necessary foundation for the next phase of work.

Next chapter preview: Application of the unfolded calculation-trajectory methodology to concrete Geofinitism case studies, including real measured time-series data from physical LRC circuits, optimisation of embedding parameters directly on computational streams, and the construction of fully finite, measurement-based stability certificates that require no intermediate matrix abstraction.

Chapter 2

Matrices as Sequential Calculation Trajectories: A First-Principles Geometric View

2.1 Introduction: Recovering the Concrete Process

In the previous chapter we established that eigenvalues and eigenvectors function as descriptors of finite sequences of matrix transformations. That perspective already moved us beyond treating matrices as opaque objects. However, even the sequence of whole matrices $A_0 \to A_1 \to A_2 \to \dots$ remains one level of abstraction above the actual work being performed.

The goal of this chapter is to go one layer deeper—to the **granular sequence of elemental calculations** themselves. We will treat every matrix operation as what it truly is: a long but strictly finite chain of individual multiplications and additions performed between specific numbers at specific moments in time. This is the level at which geometric relationships become directly visible and at which the compressed language of matrices can be unpacked without loss of meaning.

This approach aligns with the core commitment of Geofinitism: every process we analyse must remain finite, sequential, traceable, and open to geometric interpretation. When we stay at this level, we no longer need to "accept that it works." We can see, step by step, what is actually occurring.

2.2 Matrices as Temporary Containers for Sequential Operations

A matrix is a convenient organisational device. It arranges numbers into rows and columns so that rules for multiplication, addition, and decomposition can be stated compactly. Yet the moment any actual computation begins, the matrix notation is immediately expanded into ordered arithmetic.

Consider the simple act of matrix-vector multiplication $y = Ax$. The matrix A tells us which numbers must be combined with which, but the calculation itself consists of a definite sequence of scalar multiplications followed by additions. The same is true for every more complex operation—QR decomposition, eigenvalue iteration, or polynomial root-finding. The matrix provides the blueprint; the sequence of elemental calculations is the actual construction process.

When we work only with the matrix symbols, the geometric content of each individual scaling and addition becomes hidden. When we unfold the operations, that geometry reappears.

2.3 The Granular Time Series: Element-to-Element Calculations

Every matrix multiplication, every rotation or reflection inside a decomposition, and every step of an iterative algorithm ultimately reduces to operations of the form

$$\text{result} = a \times b + c \times d + \dots$$

where $a, b, c, d, \dots$ are specific numerical values taken from the current state of the calculation at that instant.

These operations occur in a strict temporal order. The output of one multiplication or addition becomes an input to later steps. The entire process is therefore a **time series of scalar calculations**. Each term in this series has a precise geometric interpretation at the moment it is performed:

- Multiplication scales one quantity by another.
- Addition combines two directed quantities (lengths or components).
- The running totals trace paths in a high-dimensional space whose coordinates are the intermediate numerical values.

Because the sequence is finite and completely determined by the original matrix entries and the chosen algorithm, it constitutes a well-defined trajectory. This trajectory can, in principle, be recorded, visualised, or further analysed using the same phase-space tools we apply to physical systems.

2.4 Geometric Content at the Most Elementary Level

Even the simplest operation carries geometric meaning. When we compute a single product $a_{ij} \times b_{kl}$, we are performing a scaling of one measured quantity by another. When we add two such products, we are combining two scaled contributions along a common direction (the output element).

In the context of linear transformations, repeated sequences of such operations collectively produce:

- Rotation (change of direction without change of length),
- Scaling (uniform or non-uniform stretching),
- Shear or reflection.

These geometric effects do not appear magically from the matrix symbol. They emerge from the ordered accumulation of many individual scalings and additions. Keeping the calculation sequence explicit makes this emergence visible rather than assumed.

2.5 Concrete Unfolding: A 2×2 Matrix Multiplication

To make the granular perspective tangible, consider the LRC state matrix in the over-damped regime ($L = 1$, $C = 1$, $R = 3$):

$$A = \begin{bmatrix} -3 & -1 \\ 1 & 0 \end{bmatrix}.$$

Suppose we need to multiply this matrix by another 2×2 matrix B whose entries are known at a particular stage of a calculation (this occurs repeatedly inside QR iterations). The product $C = AB$ is defined by four output elements, but each element expands into an explicit sequence of scalar operations.

Expanding the first output element c_{11} alone already produces the following ordered time series of primitive calculations:

1. Multiply: $-3 \times b_{11}$
2. Multiply: $-1 \times b_{21}$
3. Add the two results to obtain a partial sum for c_{11}

Repeating the same pattern for c_{12}, c_{21}, and c_{22} yields a longer but still completely finite sequence of twelve multiplications and four additions. Each multiplication scales a specific pair of numbers that exist at that moment; each addition combines two scaled contributions.

If we continue this expansion through an entire QR iteration, we obtain hundreds of such scalar steps for a modest 2×2 matrix and many thousands for larger systems. The resulting long time series is the true object of the calculation. The matrix A and the intermediate matrices serve only as temporary storage for the numbers being processed.

2.5.1 Pseudocode for Matrix Multiplication Unfolding

2.6 The Calculation Trajectory and Its Phase Space

Once we accept that every matrix algorithm generates a long time series of elemental arithmetic operations, a natural next step appears. We can treat the stream of intermediate results (partial products, running sums, or any chosen derived quantity) as a dynamical signal. Applying delay embedding to this signal reconstructs a phase portrait of the calculation process itself.

In this phase space of the calculations we can observe:

- How the sequence of numbers evolves,
- Whether the trajectory converges smoothly or exhibits transient complexity,
- Geometric features that correspond to the underlying linear transformation being computed.

This is the same technique we use to analyse measured data from physical LRC circuits, now turned inward onto the computational process. The geometry revealed is no longer only the geometry of the original physical system; it is also the geometry of the sequence of operations used to study that system.

Algorithm 2 Unfolded Matrix Multiplication $C = A \times B$

```
Input: Matrices A ∈ ℝ^{m×n}, B ∈ ℝ^{n×p}
Output: Matrix C ∈ ℝ^{m×p} and its calculation trajectory
Initialize trajectory Γ = []
for i = 1 to m do
    for j = 1 to p do
        c_ij ← 0
        for k = 1 to n do
            prod ← A_ik × B_kj
            Append (i, j, k, MULT, A_ik, B_kj, prod) to Γ
            c_ij ← c_ij + prod
            Append (i, j, k, ADD, c_ij, prod, c_ij) to Γ
        end for
        C_ij ← c_ij
    end for
end for
Return: C, Γ
```

2.7 Eigenvalues and Eigenvectors as Late-Emerging Descriptors

Within this granular view, eigenvalues and eigenvectors retain their importance but change their status. They do not exist as ready-made objects at the beginning of the calculation. They become visible only after a sufficient number of elemental operations have been executed and the running totals have aligned in particular ways.

An eigenvalue is an invariant that remains constant across the entire sequence of intermediate results. An eigenvector describes a direction in which the accumulation of scalings and additions produces a particularly simple outcome. Both are useful summaries that appear toward the end of the calculation trajectory rather than being its starting point.

This ordering—detailed sequential operations first, compact descriptors later—reverses the usual pedagogical presentation and restores a more natural order of understanding.

2.8 Implications for Finite Analysis

When every step remains explicit and finite, several practical consequences follow:

- Numerical stability can be examined directly by watching how errors propagate through the actual sequence of multiplications and additions.
- Time-varying systems ($A(t)$) generate new finite calculation trajectories at each instant rather than requiring a single fixed matrix to be "solved."
- Verification becomes possible at the level of individual operations rather than relying solely on final results.
- The same phase-space embedding tools used for physical measurements can be applied to computational streams, creating a unified geometric language.

This granular perspective therefore strengthens rather than replaces the finite-stability framework developed for the LRC circuit and similar systems.

2.9 Bridge to Algebraic Insights

The perspective developed in this chapter—matrices as organisers of explicit sequential calculation trajectories, with geometry visible at the element-to-element level—provides a foundation for examining purely algebraic questions. In particular, it offers a way to understand why certain algebraic tasks, such as finding roots of polynomials of degree five and higher, cannot be accomplished by any finite sequence of additions, multiplications, divisions, and root extractions.

These deeper algebraic implications, including the connection between calculation trajectories and nonlinear dynamics in phase space, are best treated as a distinct development. They will be explored in the following chapter so that the present chapter can remain focused on establishing the first-principles geometric view of sequential matrix calculations.

2.10 Conclusion

By unfolding matrix operations all the way to the ordered sequence of individual multiplications and additions, we recover a concrete picture in which every step carries geometric meaning. Matrices remain useful as compact notation, but they are no longer required to function as black boxes. The calculation process itself becomes the primary object—a finite trajectory whose geometry can be examined directly.

This view supports the deepest level of understanding while remaining fully compatible with practical computation. It also prepares the ground for examining more abstract algebraic questions without losing contact with the sequential, geometric reality that underlies them.

Chapter 3

Algebraic Operations as Calculation Trajectories: The Limits of Finite Radical Solutions

3.1 Introduction: From Sequential Matrix Calculations to Algebraic Unsolvability

The previous chapter established that every matrix operation unfolds into a long but finite sequence of elemental arithmetic steps, each carrying geometric meaning. Eigenvalues and eigenvectors appear as descriptors only after sufficient steps have been executed. This granular perspective restores clarity to linear algebra by treating the calculation process itself as the primary object.

The same perspective can now be extended to purely algebraic tasks—in particular, the problem of solving polynomial equations. When we attempt to find the roots of a polynomial using radicals (additions, subtractions, multiplications, divisions, and extraction of roots), we are again constructing a finite sequence of specific calculations. The question of solvability then becomes: can the required sequence of operations be kept within a form that remains expressible by a finite combination of radicals?

For polynomials of degree 2, 3, and 4, such sequences exist (though they grow rapidly in complexity). For degree 5 and higher, no general finite radical sequence exists. The reason, viewed through the lens of calculation trajectories, is that the necessary sequence of operations develops irreducible nonlinear dynamics that cannot be captured by the limited geometric operations permitted by radicals.

This chapter develops that insight without relying on the compressed language of field extensions or Galois theory. Instead, it stays with the explicit, sequential, geometric character of the calculations themselves.

3.2 Radical Operations as Explicit Calculation Sequences

A radical expression, such as $\sqrt[n]{x}$, is not an instantaneous operation. It represents a specific, finite (though possibly very long) sequence of arithmetic steps that, when executed, produce a number whose n-th power equals the input.

When we combine several such operations—for example, in the quadratic formula

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

—we are chaining together sequences of multiplications, additions, a square root, and a final division. The entire expression corresponds to one long but finite calculation trajectory whose intermediate results remain manageable because each new operation can be expressed using the same limited set of geometric primitives (scaling, addition, and one square-root extraction).

The same principle applies, with increasing complexity, to the cubic and quartic formulas. In each case, a finite sequence of radical operations can be arranged so that the overall calculation trajectory eventually yields the roots.

3.2.1 Radical Expression as a Trajectory Class

We define the class of radical-expressible calculation trajectories:

Definition 3.1 (Radical Trajectory)**.** *A calculation trajectory is a radical trajectory if every operation in the sequence is drawn from the set*

$$\mathcal{R} = \{+, -, \times, \div, \sqrt[n]{\cdot}\}$$

and the sequence length is finite.

The question of solvability by radicals is then: for a given polynomial $p(x)$, does there exist a radical trajectory that produces its roots as output?

3.3 Low-Degree Polynomials: Calculation Trajectories That Remain Controllable

For quadratic, cubic, and quartic equations, it is possible to construct a calculation sequence whose geometric structure stays within the operations permitted by radicals. The trajectory of intermediate values does not develop entanglements that cannot be undone by further scalings, additions, or root extractions.

In geometric terms, the path traced by the successive partial results can be "unwound" using the allowed operations. Each new radical introduces a controlled branching or scaling that remains reversible within the same formal language. The calculation process therefore closes after a finite number of steps, producing explicit expressions for the roots.

This controllability is not accidental. It reflects the fact that the algebraic relations imposed by polynomials of degree four or lower can be resolved by transformations whose sequential unfolding does not generate irreducible complexity in the space of intermediate calculations.

3.4 Higher-Degree Polynomials: The Emergence of Irreducible Nonlinear Dynamics

When we attempt the same approach for a general quintic (degree-5) polynomial, no such finite radical sequence exists. From the perspective of calculation trajectories, the reason becomes visible:

Any attempt to solve the equation by radicals requires constructing a sequence of multiplications, additions, and root extractions whose final output satisfies the original polynomial. However, when the operations are unfolded into their elementary steps and the trajectory of intermediate values is examined, that trajectory develops folding, stretching, and entanglement characteristic of nonlinear dynamical systems.

3.4.1 The Galois Connection

This geometric obstruction corresponds precisely to the algebraic obstruction identified by Galois theory. The Galois group of the general quintic is the symmetric group S_5, which is not solvable. A solvable group is one that can be decomposed into a tower of abelian extensions, corresponding to the nesting of radical operations.

In the calculation-trajectory view, the non-solvability of S_5 manifests as the impossibility of constructing a radical trajectory that can navigate the required geometric relations. The calculation trajectory would need to pass through regions whose geometry lies outside the transformations generatable by radicals. This is why the process cannot be completed in closed form using only those operations.

3.4.2 The Geometric Signature of Unsolvability

In other words, the calculation process itself becomes nonlinear in a way that cannot be straightened or resolved by any further finite combination of the allowed radical operations. New intermediate results appear whose relationships to previous values cannot be expressed using only scalings, additions, and root extractions without introducing new entanglements that themselves require further radicals—leading to an infinite regress.

The trajectory in the space of partial calculation results no longer converges to the roots within the permitted geometric operations. Instead, it exhibits the kind of irreducible complexity that appears when a dynamical system is allowed to evolve under its own nonlinear rules without external simplification.

This is the concrete meaning, at the level of sequential calculations, of Abel's impossibility theorem: there is no finite sequence of the allowed operations that can serve as a general solution method for degree 5 and higher.

3.5 Phase-Space View of the Calculation Process

If we record the stream of intermediate numerical values generated while attempting a radical solution, we obtain a time series. Applying delay embedding to this series reconstructs a phase portrait of the attempted solution process.

For solvable low-degree cases, the reconstructed attractor in this calculation phase space remains relatively simple—typically reducible to paths that can be traversed using the geometric operations corresponding to radicals. The trajectory eventually reaches the roots after a finite number of steps.

For unsolvable higher-degree cases, the phase portrait of the calculation trajectory develops complex folding and mixing. The path required to reach the roots cannot be expressed as a finite sequence of the simple geometric transformations permitted by radicals. The dynamics of the calculation process itself become too entangled.

This phase-space perspective makes the distinction tangible: solvability by radicals corresponds to the existence of a calculation trajectory whose geometry stays within a restricted class of operations. When that geometry escapes the permitted class, no finite radical expression can describe the solution.

3.6 Geometric Interpretation of the Barrier

At the most elementary level, each multiplication scales one quantity by another and each addition combines directed segments. A root extraction corresponds to finding a length that, when scaled by itself a certain number of times, returns the input. These are all local geometric operations.

When these operations are chained to solve a polynomial equation, they must collectively undo the original polynomial relation. For degrees up to four, such a chain can be arranged. For degree five and higher, the relations imposed by the polynomial require a global geometric rearrangement that cannot be achieved by any finite local sequence of scalings, additions, and root extractions.

The calculation trajectory would need to pass through regions whose geometry lies outside the transformations generatable by radicals. This is why the process cannot be completed in closed form using only those operations.

3.7 Implications for First-Principles Understanding

Viewing algebraic solvability through the lens of explicit calculation trajectories has several clarifying consequences:

- It removes the need to accept unsolvability as a mysterious theorem. The obstruction appears directly as the inability of a finite sequence of permitted operations to resolve the required geometric relationships.

- It unifies the treatment of linear algebra and algebra: both are understood as the construction and analysis of finite sequences of elemental calculations.

- It suggests that numerical or iterative methods succeed where radical solutions fail precisely because they allow more general sequences of operations (not restricted to radicals) that can navigate the more complex calculation trajectories.

- It reinforces the value of keeping calculations explicit rather than compressed: the deeper geometric reasons for both success and failure become visible only when the sequential process is unfolded.

This perspective does not replace classical results but supplies an intuitive, revisitable foundation for why those results hold.

3.8 Conclusion

By treating algebraic operations—including the search for polynomial roots—as explicit sequences of calculations, we see that the distinction between solvable and unsolvable cases is fundamentally geometric and dynamical. Low-degree polynomials permit calculation trajectories that remain controllable within the operations allowed by radicals.

Higher-degree polynomials generate trajectories whose nonlinear character cannot be resolved by any finite continuation of those same operations.

This insight completes the movement begun in the preceding chapters: from compressed matrix and algebraic formalisms toward a consistent view in which every process is understood as a finite, sequential, geometrically meaningful trajectory of calculations. The same principles that clarify the behaviour of LRC circuits and eigenvalue computations also illuminate the limits of radical solvability in algebra.

The framework developed across these chapters therefore provides a coherent foundation for further exploration in Geofinitism—one in which understanding arises from following the actual sequence of operations rather than accepting compressed results.

Chapter 4

Phase Space Visualization of Calculation Trajectories

4.1 Introduction

The preceding chapters developed the view that both linear algebraic operations and the search for polynomial roots can be understood as explicit sequences of elemental calculations. These sequences form trajectories whose geometry becomes visible when examined directly.

This short chapter presents a visual exploration of that idea. By recording the ordered sequence of intermediate values that arise when applying solution formulas (or numerical methods) and then applying delay embedding, we can reconstruct phase portraits of the calculation process itself. These portraits make the increasing complexity with polynomial degree directly observable.

4.2 Method: Delay Embedding of Calculation Sequences

For each solution process we extract a time series consisting of the major intermediate numerical results. We then apply a simple delay embedding (Takens' method with embedding dimension 2 and delay $\tau = 1$) to reconstruct a two-dimensional phase portrait. In this portrait:

- Each point represents the state of the calculation at a particular step.
- The path traced by the points reveals the geometric structure of the sequence of operations.

This approach allows us to compare the character of calculation trajectories across different degrees without relying on abstract algebraic language.

4.2.1 Detailed Embedding Procedure

For a calculation sequence producing intermediate values $v_1, v_2, \ldots, v_T$, the 2D delay embedding with $\tau = 1$ is:

$$\Phi_2(v_t) = (v_t,\ v_{t-1})$$

for $t = 2, \ldots, T$. This produces a trajectory in the plane whose points are pairs of successive intermediate values.

The embedding dimension d can be chosen using False Nearest Neighbours analysis, and the optimal delay τ using Average Mutual Information. For the purposes of visualization, we use $d = 2$, $\tau = 1$ to produce simple, interpretable portraits.

4.3 Quadratic Formula Trajectory

Consider the quadratic equation $x^2 - 5x + 6 = 0$, with roots $x = 2$ and $x = 3$. Applying the quadratic formula:

$$x = \frac{5 \pm \sqrt{25 - 24}}{2} = \frac{5 \pm 1}{2}.$$

The intermediate calculation sequence is:

1. Compute discriminant: $25 - 24 = 1$
2. Compute square root: $\sqrt{1} = 1$
3. Compute numerator: $5 + 1 = 6$
4. Compute denominator: $2 \times 1 = 2$
5. Compute root: $6/2 = 3$
6. Compute numerator: $5 - 1 = 4$
7. Compute root: $4/2 = 2$

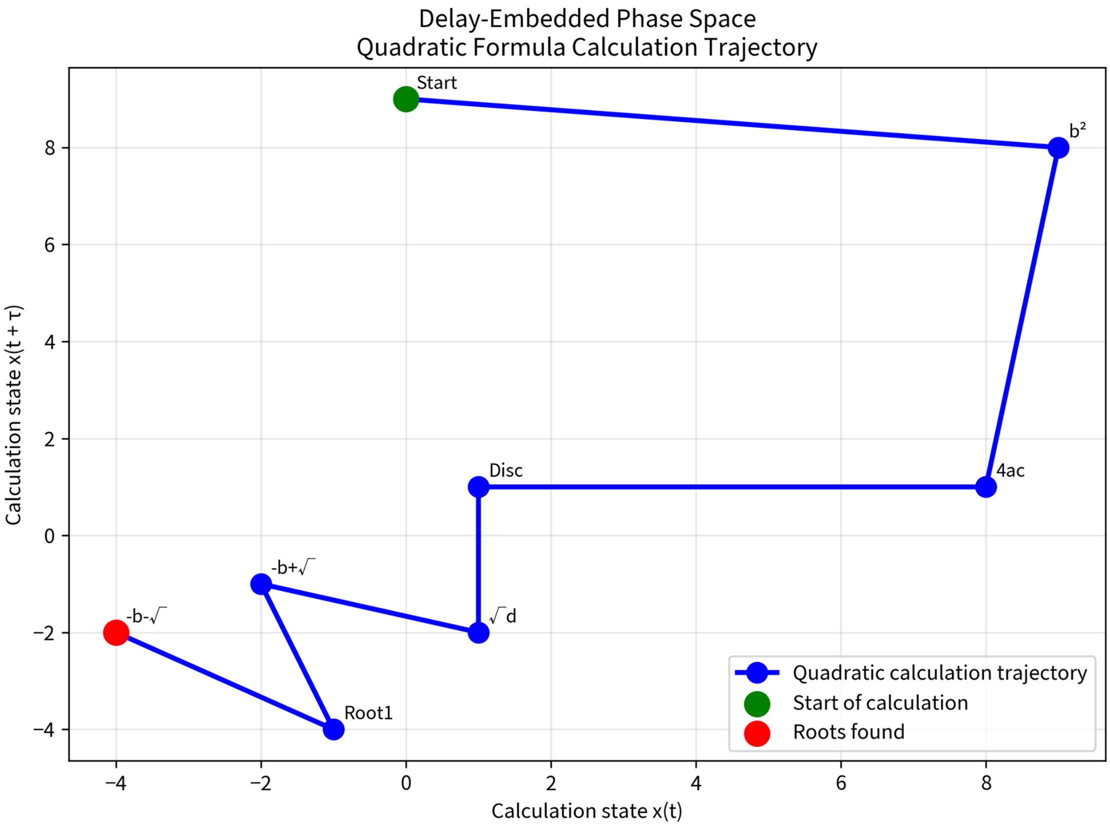

Figure 4.1: Delay-embedded phase space of the intermediate calculation steps when applying the quadratic formula. The trajectory is short, direct, and terminates cleanly at the roots.

The quadratic case produces a compact trajectory with relatively few turns. The calculation sequence remains simple and concludes after a small number of explicit steps.

4.4 Cubic Formula Trajectory

For the depressed cubic $x^3 + px + q = 0$, Cardano's formula gives:

$$x = \sqrt[3]{-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}.$$

Consider $x^3 - 6x - 6 = 0$ (so $p = -6$, $q = -6$):

$$x = \sqrt[3]{3 + \sqrt{1}} + \sqrt[3]{3 - \sqrt{1}} = \sqrt[3]{4} + \sqrt[3]{2}.$$

The intermediate calculation sequence is considerably longer, involving:

1. Compute $p/3 = -2$
2. Compute $(p/3)^3 = -8$
3. Compute $q/2 = -3$

4. Compute $(q/2)^2 = 9$
5. Compute discriminant $9 - 8 = 1$
6. Compute square root: $\sqrt{1} = 1$
7. Compute cube root: $\sqrt[3]{4} \approx 1.5874$
8. Compute cube root: $\sqrt[3]{2} \approx 1.2599$
9. Sum roots: $1.5874 + 1.2599 \approx 2.8473$
10. Compute remaining roots via quadratic factor

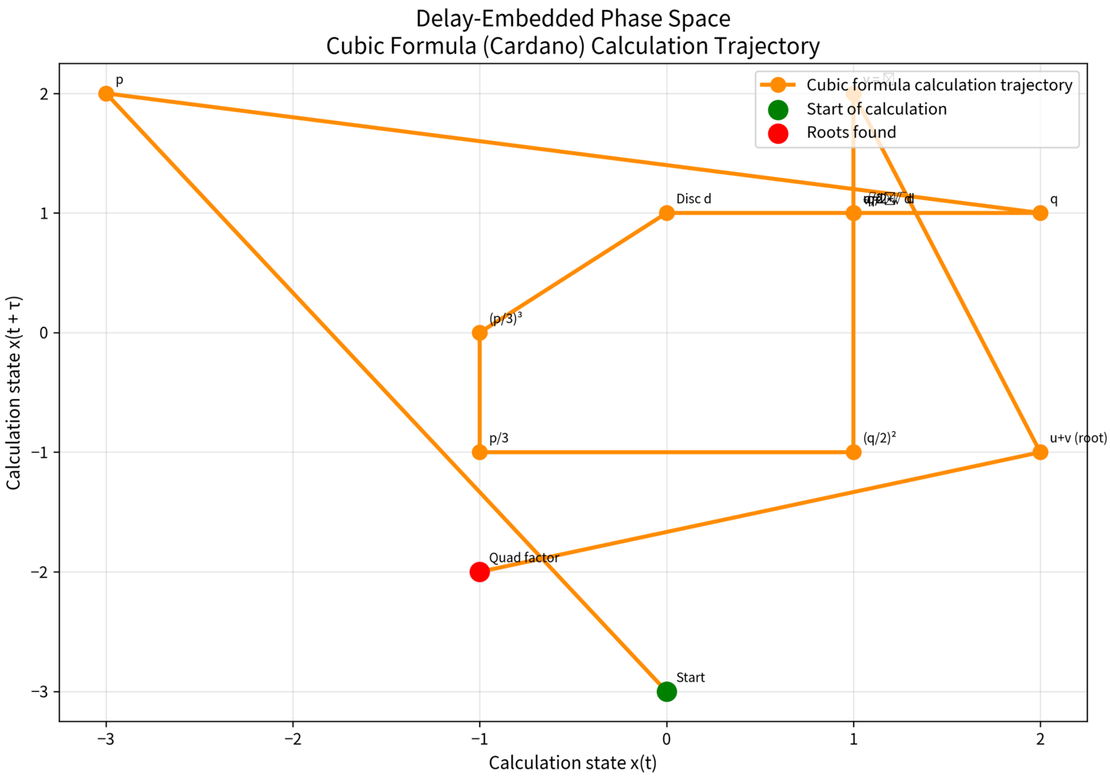

Figure 4.2: Delay-embedded phase space of the major intermediate steps in Cardano's formula for a depressed cubic. The trajectory is visibly longer and contains more branches and intermediate stages (discriminant, cube roots u and v, and the quadratic factor for remaining roots).

Already at degree 3 the embedded trajectory shows noticeably greater structure and length. Multiple distinct stages appear, corresponding to the more elaborate sequence of operations required by the cubic formula.

4.5 Contrast: Numerical Iteration on a Quintic

For polynomials of degree 5 and higher, no general finite radical formula exists. As a contrast, we examine the calculation trajectory generated by Newton's method applied to a quintic equation.

Newton's method for finding roots of $f(x) = 0$ iterates:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

For the quintic $x^5 - x - 1 = 0$, starting from $x_0 = 1$:

1. $x_0 = 1.0$, $f = -1.0$, $f' = 4.0$, $x_1 = 1.25$
2. $x_1 = 1.25$, $f \approx 1.3027$, $f' \approx 8.7695$, $x_2 \approx 1.1014$
3. $x_2 \approx 1.1014$, $f \approx 0.4269$, $f' \approx 6.6819$, $x_3 \approx 1.0375$
4. $\ldots$ continues converging to $x \approx 1.1673$

The trajectory length is determined by the convergence criterion, typically tens to hundreds of iterations.

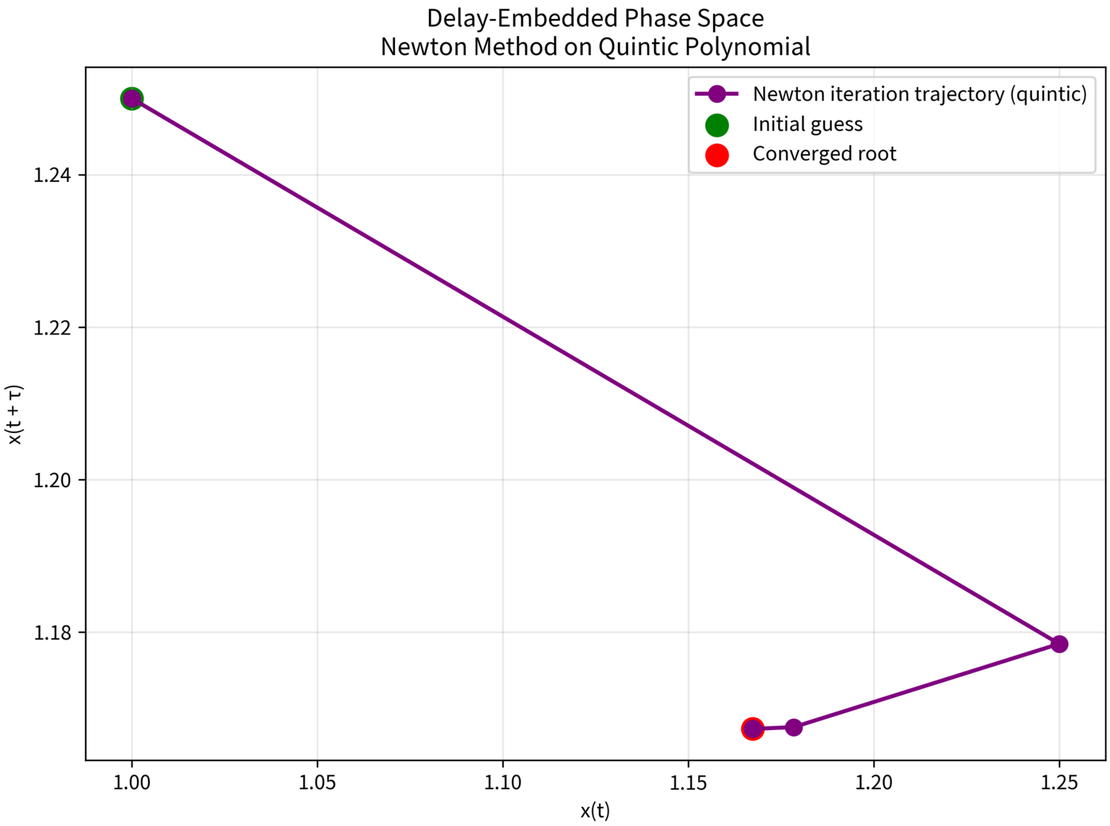

Figure 4.3: Delay-embedded phase space of Newton iterations on the quintic equation $x^5 - x - 1 = 0$. The trajectory is extended and exhibits spiraling behavior before convergence—characteristic of a longer, more complex iterative calculation process.

The Newton trajectory on the quintic is markedly more extended and dynamically richer than the radical-based trajectories of the quadratic and cubic cases. This illustrates the kind of calculation complexity that appears when one moves beyond what can be resolved by a finite sequence of radical operations.

4.6 Interpretation

These visualizations reveal a clear progression:

- Low-degree cases (quadratic) produce short, simple trajectories that terminate quickly.
- Cubic cases already display increased length and branching.
- Higher-degree problems, when approached numerically, generate extended trajectories with more intricate geometry.

This pattern supports the central insight of the preceding chapters: the obstruction to solving higher-degree polynomials by radicals is visible at the level of the calculation process itself. The geometric structure of the required sequence of operations grows beyond what can be expressed using only a finite combination of additions, multiplications, and root extractions.

The phase portraits make tangible why radical methods "run out of expressive power" while iterative numerical methods can continue navigating more complex calculation landscapes.

4.7 Conclusion

By embedding the sequences of intermediate calculation values into phase space, we obtain a geometric representation of algebraic processes. These portraits show, in a direct and visual way, how the character of the calculation trajectory changes with polynomial degree.

This approach reinforces the first-principles perspective developed throughout: understanding arises most clearly when we follow the actual ordered sequence of operations rather than remaining within compressed formalisms. The visualizations presented here provide a concrete reference point for revisiting and deepening that understanding.

Chapter 5

Conclusion: Geofinitism and the Reconstruction of Mathematics

5.1 Synthesis of the Argument

The preceding chapters have developed a unified perspective on linear algebra, algebraic solvability, and computation through the lens of Geofinitism. The core argument can be summarised in three propositions:

1. **Process Primacy**: Mathematical objects are not static entities but compressed records of finite processes. Matrices compress instruction sequences; eigenvalues compress stabilization phenomena; algebraic formulas compress calculation trajectories.

2. **Trajectory Geometry**: The unfolding of these compressed processes reveals genuine geometric structure—calculation trajectories that can be examined using the same phase-space techniques developed for physical systems.

3. **Descriptor Status**: Classical mathematical objects function as descriptors of these trajectories. They are useful fictions that summarise invariant or emergent features of the underlying sequential processes.

5.2 Implications for Mathematics and Computation

The Geofinitist perspective has several implications for how we understand and practice mathematics.

5.2.1 Finite Process Unfolding as a Methodological Principle

Finite Process Unfolding provides a systematic method for recovering the sequential structure compressed within static symbolic forms. This method can be applied to any mathematical formalism, revealing the underlying calculation trajectories and making visible the assumptions and limitations of the compressed representation.

5.2.2 The Role of Idealisation

Classical mathematics relies heavily on idealisations: infinite processes, continuous quantities, perfect precision. Geofinitism does not reject these idealisations but reframes them as useful fictions. The question is not whether they are "true" but whether they are productive and whether their limitations are understood.

The Finite-Symbol Embedding Theorem formalises this relationship: classical Takens embedding is the limit of finite-symbol embedding as resolution goes to zero. This establishes a precise relationship between ideal and finite representations.

5.2.3 Measurement as Foundation

The Geofinitism programme begins from measurement rather than from axioms. Every mathematical symbol carries measurement provenance, resolution bounds, and irreducible uncertainty. This measurement-first approach provides a foundation for mathematics that is grounded in finite experience rather than in platonic abstractions.

5.3 Connections to the Broader Geofinitism Programme

This work connects to several ongoing developments in the Geofinitism programme:

- **Finite Symbolic Mechanics (FSM)**: The framework of trajectories and descriptors developed here is a specific instance of the broader FSM programme, which reconstructs all of mathematics from finite measurement principles.
- **Language Dynamics**: The application of phase-space embedding to computational sequences parallels the treatment of language as a nonlinear dynamical system. Both involve the reconstruction of attractors from finite symbolic sequences.
- **The Takens-Haylett Theorem**: The Finite-Symbol Embedding Theorem provides the formal license for applying Takens reconstruction to discrete symbolic systems, including both language sequences and computational trajectories.
- **The Generon Framework**: Numbers and mathematical objects are reframed as generons—processes that, when executed, produce measured results. The calculation trajectories analysed in this work are specific instances of generonic processes.

5.4 Open Questions and Future Work

Several directions for further development remain.

5.4.1 Formalizing Calculation Trajectory Invariants

The manuscript has identified eigenvalues as invariants of matrix calculation trajectories. What other invariants exist? For example:

- Trace and determinant as trajectory invariants,
- Condition number as a measure of trajectory stability,

- Convergence rates as dynamical properties of the trajectory.

A systematic classification of calculation trajectory invariants would provide a richer understanding of the relationship between processes and their descriptors.

5.4.2 Phase-Space Embedding of Different Algorithms

Different algorithms for the same problem generate different calculation trajectories. Comparing the phase portraits of QR iteration, Jacobi method, and power iteration for the same matrix would reveal the geometric character of each algorithm and potentially guide algorithm selection.

5.4.3 Finite Stability Certificates

The concept of finite stability introduced in Chapter 1 could be developed into practical certificates: procedures that verify both physical stability and computational convergence within finite bounds. This would provide a rigorous foundation for numerical computation.

5.4.4 Connections to Algebraic Complexity Theory

The calculation-trajectory view connects naturally to algebraic complexity theory, which studies the minimum number of operations required to compute mathematical functions. The descriptor perspective suggests new complexity measures based on trajectory geometry rather than operation count alone.

5.4.5 AI and Language Models

The MARINA architecture demonstrates that explicit Takens embedding can replace attention in language models. The calculation-trajectory framework developed here suggests that language generation and matrix computation are both instances of the same underlying process: the generation of finite symbolic trajectories.

5.5 Final Reflection

The Geofinitist perspective offers a way of doing mathematics that remains grounded in finite, sequential, measurable processes while preserving the power and elegance of classical formalisms. The calculation trajectories analysed in this work are not merely philosophical abstractions but concrete, traceable sequences of operations that can be examined, visualised, and understood.

By treating matrices as compressed instruction sets, eigenvalues as stabilised descriptors, and algebraic operations as finite trajectories, we recover a geometry of computation that has been obscured by the very success of mathematical compression. This geometry is not opposed to classical mathematics but complementary to it—revealing the process behind the symbol, the trajectory behind the object, and the finite reality behind the idealisation.

The work presented here is one step in a larger programme: the reconstruction of mathematics from first principles grounded in finite measurement and sequential process.

If successful, this programme will not replace existing mathematics but will provide a deeper understanding of its foundations and a clearer view of its limits.

Bibliography

[1] Haylett, K. R. (2025). Geofinitism: A Measurement-First Philosophy of Language and Mathematics. *Attralucian Essays*, ATT_08.

[2] Haylett, K. R. (2025). The Five Pillars of Geofinitism. *Attralucian Essays*, ATT_49.

[3] Haylett, K. R. (2025). Finite Process Unfolding: A Method for Recovering Temporal Structure from Static Symbolic Forms. *Attralucian Essays*, ATT_52.

[4] Haylett, K. R. (2025). Mathematics as Lenses: Geofinitism and the Reconstruction of Discrete Dynamical Structure. *Foundational Monographs*, M01.

[5] Haylett, K. R. (2025). From Generon to Meaning: Compression, Boundary, and the Dynamics of Finite Symbols. *Attralucian Essays*, ATT_77.

[6] Haylett, K. R. (2025). Introducing the Functional Symbolic Trajectory: Words, Compression, and the Flow of Meaning in Finite Language. *Attralucian Essays*, ATT_81.

[7] Haylett, K. R. (2025). The Measured World: Where Compression Replaces Correspondence. *Selected Communications*, P06.

[8] Trefethen, L. N., & Bau, D. (1997). *Numerical Linear Algebra*. SIAM.

[9] Haylett, K. R. (2025). The Finite-Symbol Embedding Theorem: Phase Space Reconstruction for Finite Symbolic Dynamical Systems. *Foundational Monographs*, M02.

[10] Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, Lecture Notes in Mathematics, Vol. 898, pp. 366–381. Springer.

[11] Galois, É. (1832). Mémoire sur les conditions de résolubilité des équations par radicaux. *Journal de Mathématiques Pures et Appliquées*.

[12] Stewart, I. (2015). *Galois Theory* (4th ed.). CRC Press.

[13] Abel, N. H. (1826). Mémoire sur les équations algébriques, où l'on démontre l'impossibilité de la résolution de l'équation générale du cinquième degré. Cristiania.

[14] Kennel, M. B., Brown, R., & Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, 45(6), 3403–3411.

[15] Fraser, A. M., & Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2), 1134–1140.

[16] Haylett, K. R. (2025). Admissibility, Finite Symbols, and the Limits of Measurement. *Selected Communications*, P14.

[17] Haylett, K. R. (2025). The Principia Geometrica: Finite Symbolic Mechanics. *Foundational Monographs*, M07.

[18] Haylett, K. R. (2025). Language as a Nonlinear Dynamical System. *Selected Communications*, P05.

[19] Haylett, K. R. (2025). The Finite-Symbol Embedding Theorem (Takens-Haylett Theorem). *Selected Communications*, P04.

[20] Haylett, K. R. (2025). The Generon: Process, Measurement, and the Completion of the Geofinite Ontology. *Attralucian Essays*, ATT_23.

[21] Bürgisser, P., Clausen, M., & Shokrollahi, M. A. (1997). *Algebraic Complexity Theory*. Springer.

[22] Haylett, K. R. (2025). Introducing the Takens-Based Transformer (MARINA). *Selected Communications*, P01.

[23] Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Johns Hopkins University Press.

[24] Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms* (2nd ed.). SIAM.

[25] Stewart, G. W. (1998). *Matrix Algorithms: Volume I: Basic Decompositions*. SIAM.