Introduction

Daniel Ari Friedman
ORCID: 0000-0001-6232-9096
Email: daniel@activeinference.institute

August 16, 2025

Contents

1 Introduction

1.1 Abstract

We review a unified analytical framework for four dimensional (4D) modeling and Quadray coordinates, synthesizing geometric foundations, optimization on tetrahedral lattices, and information geometry. Building on R. Buckminster Fuller's Synergetics and the Quadray coordinate system, with extensive reference to Kirby Urner's computational implementations across multiple programming languages (see the comprehensive 4dsolutions ecosystem including Python, Rust, Clojure, and POV-Ray implementations), we review how integer lattice constraints yield integer volume quantization of tetrahedral simplexes, creating discrete "energy levels" that regularize optimization and enable integer-based optimization. We adapt standard methods (e.g., Nelder–Mead method) to the quadray lattice, define Fisher information in Quadray parameter space, and analyze optimization as geodesic motion on an information manifold via the natural gradient. We review three distinct 4D namespaces — Coxeter.4D (Euclidean E^4), Einstein.4D (Minkowski spacetime), and Fuller.4D (synergetics/Quadrays) — develop analytical tools and equations, and survey extensions and applications across AI, active inference, cognitive security, and complex systems. The result is a cohesive, interpretable approach for robust, geometry-grounded computation in 4D. All source code for the manuscript is available at QuadMath.

Keywords: Quadray coordinates, 4D geometry, tetrahedral lattice, integer volume quantization, information geometry, optimization, synergetics, active inference.

1.2 Overview

Quadray coordinates provide a tetrahedral basis for modeling space and computation, standing in contrast to Cartesian cubic frameworks. Originating in Buckminster Fuller's Synergetics, quadray coordinates enable the replacement of right-angle orthonormal assumptions, with 60-degree coordination and a unit tetrahedron of volume 1. This reframing yields striking integer relationships among common polyhedra and provides a natural account of space via close-packed spheres and the isotropic vector matrix (IVM).

This paper unifies three threads:

- **Foundations**: Quadray coordinates and their relation to 4D modeling more generally, with explicit namespace usage (Coxeter.4D, Einstein.4D, Fuller.4D) to maintain clarity.
- **Optimization framework**: Leverages integer volume quantization on tetrahedral lattices to achieve robust, discrete convergence.
- **Information geometry**: Tools (e.g., Fisher Information, free-energy minimization) for interpreting optimization as geodesic motion on statistical manifolds.

1.3 4D Namespace Framework

In this synthetic review, we distinguish three internal meanings of "4D," following a dot-notation that avoids cross-domain confusion. For comprehensive details, see Section 2: 4D Namespaces.

- **Coxeter.4D** — four-dimensional Euclidean space (E^4), as in classical polytope theory. Coxeter emphasizes that Euclidean 4D is not spacetime; see the Dover edition of Regular Polytopes (p. 119) for a clear statement to this effect; background on lattice packings in four dimensions aligns with the treatment in Conway & Sloane's Sphere Packings, Lattices and Groups.
- **Einstein.4D** — Minkowski spacetime (3D + time) with an indefinite metric; appropriate for relativistic physics but distinct from Euclidean E^4.
- **Fuller.4D** — synergetics' tetrahedral accounting of space using Quadrays (four non-negative coordinates with at least one zero after normalization) and the Isotropic Vector Matrix (IVM) = Cubic Close Packing (CCP) = Face-Centered Cubic (FCC) correspondence. This treats the regular tetrahedron as a natural unit container and emphasizes angle/shape relations independent of time/energy.

1.4 Contributions

The paper makes the following key contributions:

- **Namespaces mapping**: Coxeter.4D (Euclidean E^4), Einstein.4D (Minkowski spacetime), and Fuller.4D (Quadrays/IVM) → analytical tools and examples.
- **Quadray-adapted Nelder–Mead**: Integer-lattice normalization and volume-level tracking.
- **Equations and methods**: Comprehensive supplement with guidance for high-precision computation using `libquadmath`.
- **Discrete optimizer**: Integer-valued variational descent over the IVM (`discrete_ivm_descent`) with animation tooling, connecting lattice geometry to information-theoretic objectives.

1.5 Manuscript Structure

- **Introduction**: motivates Quadrays, clarifies 4D namespaces, and summarizes contributions.
- **Methods**: details coordinate conventions, exact tetravolumes, conversions, and lattice-aware optimization methods (Nelder–Mead and discrete IVM descent).
- **Results**: empirical comparisons and demonstrations are shown inline and saved under `quadmath/output/` (PNG/CSV/NPZ/MP4) for reproducibility.
- **Discussion**: interprets results, limitations, and implications; outlines future work.
- **Appendices**: equations, free-energy background, and a consolidated symbols/glossary with an auto-generated API index.

1.6 Companion Code and Tests

The manuscript is accompanied by a fully-tested Python codebase under `src/` with unit tests under `tests/`. Key artifacts used throughout the paper:

- **Quadray APIs**: `src/quadray.py` (`Quadray`, `integer_tetra_volume`, `ace_tetravolume_5x5`).
- **Determinant utilities**: `src/linalg_utils.py` (`bareiss_determinant_int`).
- **Length-based volume**: `src/cayley_menger.py` (`tetra_volume_cayley_menger`, `ivm_tetra_volume_cayley_menger`).
- **XYZ conversion**: `src/conversions.py` (`urner_embedding`, `quadray_to_xyz`).

- **Examples**: `src/examples.py` (`example_ivm_neighbors`, `example_volume`, `example_optimize`).

For comprehensive background resources, computational implementations, and related work, see the Resources section.

1.7 Reproducibility and Data Availability

- The manuscript Markdown and code to generate the PDF are available on the project repository (`QuadMath` on GitHub, @docxology username). See the repository home page for source, figures, and scripts: QuadMath repository.
- The manuscript is licensed under the Apache License 2.0. See the LICENSE file for details.
- The manuscript is accompanied by a fully-tested Python codebase under `src/` with unit tests under `tests/`, complemented by extensive cross-validation against Kirby Urner's reference implementations in the 4dsolutions ecosystem. See the Resources section for comprehensive details on computational implementations and validation.
- All figures referenced in the manuscript are generated by scripts under `quadmath/scripts/` and saved to `quadmath/output/` with lightweight CSV/NPZ alongside images.
- Tests accompany all methods under `src/` and enforce 100% coverage for `src/`.
- Symbols and notation are standardized across sections; see Appendix: Symbols and Glossary for a consolidated table of variables and constants used throughout. Equation labels (e.g., Eq. (**??**) and Eq. (**??**)) and figure labels are automatically numbered by LaTeX for consistent cross-referencing.

1.8 Graphical Abstract

Panel A shows Quadray axes (A,B,C,D) under a symmetric embedding with wireframe context. **Panel B** shows close-packed spheres at the tetrahedron vertices (IVM/CCP/FCC, "twelve around one").
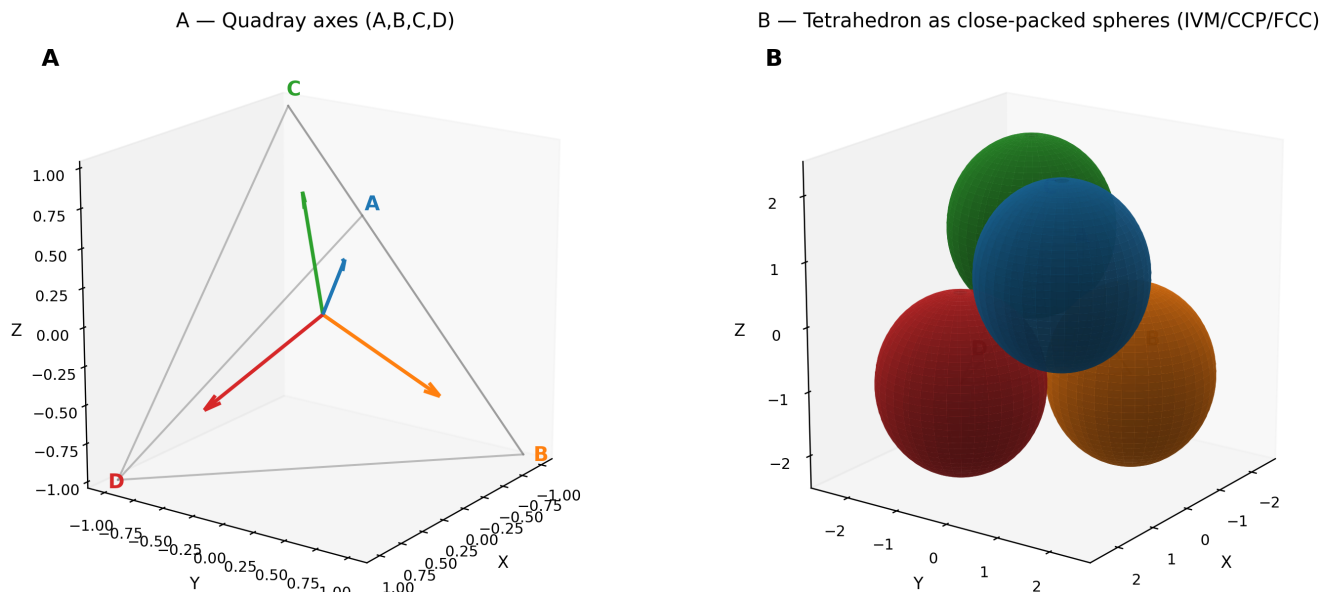
Figure 1: **Quadray coordinate system overview (graphical abstract)**. **Panel A**: Four Quadray axes (A,B,C,D) rendered as colored directional arrows from the origin to the vertices of a regular tetrahedron under the default symmetric embedding. Each axis is distinctly colored (A=blue, B=orange, C=green, D=red) with axis labels positioned at the vertex endpoints. A light gray wireframe connects the four vertices to emphasize the tetrahedral geometry underlying the coordinate system. This panel illustrates the fundamental Fuller.4D direction-based structure where Quadrays represent four canonical directions in tetrahedral space rather than orthogonal Cartesian dimensions. **Panel B**: The same tetrahedral vertices shown as close-packed spheres with radius chosen so neighboring spheres kiss along tetrahedron edges, emphasizing the connection to the Isotropic Vector Matrix (IVM), Cubic Close Packing (CCP), and Face-Centered Cubic (FCC) arrangements. Each sphere is colored to match its corresponding axis from Panel A, with light edge wireframes providing geometric context. This visualization demonstrates how Quadray coordinates naturally align with dense sphere packing and the "twelve around one" coordination motif central to synergetics and Fuller.4D modeling.