

QuadMath: Front Matter and Abstract

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

August 15, 2025

Contents

1 QuadMath: An Analytical Review of 4D and Quadray Coordinates	1
1.1 Abstract	1
1.2 Manuscript structure	1
1.3 Reproducibility and data availability	2

1 QuadMath: An Analytical Review of 4D and Quadray Coordinates

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**. The future is open source and 4D!

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

1.2 Manuscript structure

- Introduction: motivates Quadrays, clarifies 4D namespaces (Coxeter.4D, Einstein.4D, Fuller.4D), 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.

1.3 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](#).
- 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/`; external validation includes comparisons with `qrays.py` and `tetravolume.py` algorithms.
- 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 (e.g., Figure 8) are used consistently.