Equations and Math Supplement

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

August 15, 2025

Contents

1 Equations and Math Supplement (Appendix)

1.1 Volume of a Tetrahedron (Lattice)

$$V = \tfrac{1}{6}\,\left|\det\left[\,P_1 - P_0,\ P_2 - P_0,\ P_3 - P_0\,\right]\right| \tag{1}$$

Notes.

- $P_0, \ldots, P_3$ are vertex coordinates; the determinant computes the volume of the parallelepiped spanned by edge vectors, with the $1/6$ factor converting to tetra volume.

Tom Ace 5×5 tetravolume (IVM units):

$$V_{ivm} = \tfrac{1}{4}\left|\det\begin{pmatrix} a_0 & a_1 & a_2 & a_3 & 1 \\ b_0 & b_1 & b_2 & b_3 & 1 \\ c_0 & c_1 & c_2 & c_3 & 1 \\ d_0 & d_1 & d_2 & d_3 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}\right| \tag{2}$$

Notes.

- Rows correspond to Quadray 4-tuples of the vertices; the last row encodes the affine constraint. Division by 4 returns IVM tetravolume.

XYZ determinant volume and S3 conversion:

$$V_{xyz} = \frac{1}{6}\left|\det\begin{pmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{pmatrix}\right|, \qquad V_{ivm} = S3\, V_{xyz}, \quad S3 = \sqrt{\frac{9}{8}} \tag{3}$$

Notes.

- Homogeneous determinant in Cartesian coordinates for tetra volume; conversion to IVM units uses $S3 = \sqrt{9/8}$ as used throughout.

See code: `tetra_volume_cayley_menger`. For tetrahedron volume background, see Tetrahedron – volume. Exact integer determinants in code use the Bareiss algorithm. External validation: these formulas align with implementations in the 4dsolutions ecosystem. See the Resources section for comprehensive details.

1.2 Fisher Information Matrix (FIM)

Background: Fisher information.

$$F_{i,j} = \mathbb{E}\left[\frac{\partial \log p(x;\theta)}{\partial \theta_i}\,\frac{\partial \log p(x;\theta)}{\partial \theta_j}\right] \tag{4}$$

Notes.

- Defines the Fisher information matrix as the expected outer product of score functions; see Fisher information.

Figure: empirical estimate shown in the FIM heatmap figure. See code: `fisher_information_matrix`.

See `src/information.py` — empirical outer-product estimator (`fisher_information_matrix`).

1.3 Natural Gradient

Background: Natural gradient (Amari).

$$\theta \leftarrow \theta - \eta\, F(\theta)^{-1}\, \nabla_\theta L(\theta) \tag{5}$$

Explanation.

- Natural gradient update: right-precondition the gradient by the inverse of the Fisher metric (Amari); see Natural gradient.

See code: `natural_gradient_step`.

See `src/information.py` — damped inverse-Fisher step (`natural_gradient_step`).

1.4 Free Energy (Active Inference)

$$\mathcal{F} = -\log P(o \mid s) + \mathrm{KL}\big[Q(s) \parallel P(s)\big] \tag{6}$$

Explanation.

- **Partition**: variational free energy decomposes into expected negative log-likelihood and KL between approximate posterior and prior; see Free energy principle.

See code: `free_energy`.

See `src/information.py` — discrete-state variational free energy (`free_energy`).

Note: The main figures demonstrating natural gradient trajectories and free energy landscapes are shown in Section 4: Optimization in 4D. The appendix focuses on unique figures specific to mathematical formulations and validation.

1.4.1 Figures

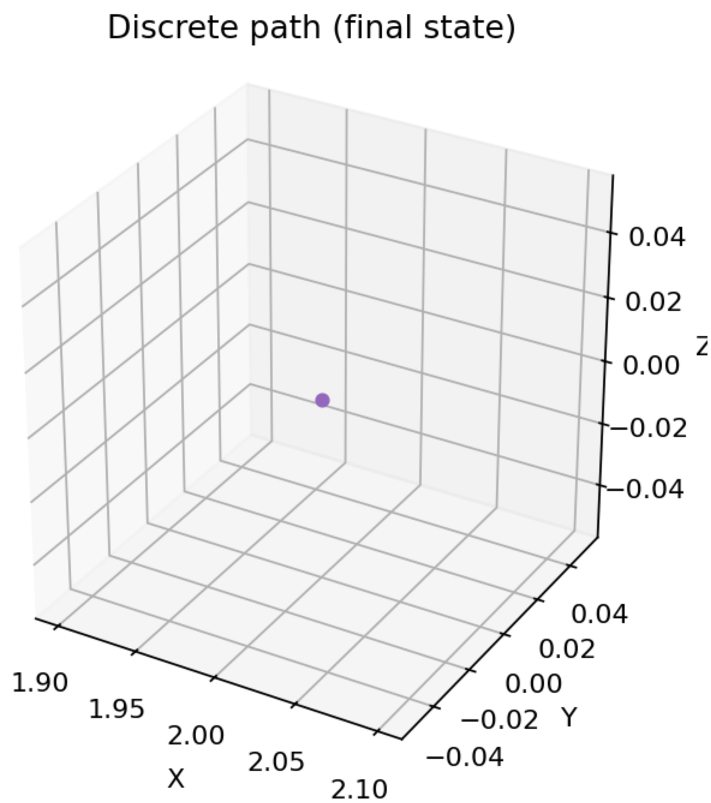

Figure 1: **Discrete IVM descent optimization path (final converged state)**. This static frame shows the final position of a discrete variational descent algorithm operating on the integer Quadray lattice. **Points**: Colored spheres representing the final optimization state, each positioned at integer Quadray coordinates projected to 3D space via the default embedding matrix. **Colors**: Each point has a distinct color for easy identification of different optimization components. **Optimization context**: These points represent the final state of the discrete IVM descent algorithm after converging to a local optimum on the integer lattice. The tight clustering of points indicates successful convergence, with the algorithm having found a stable configuration. **Lattice constraints**: All point positions correspond to integer Quadray coordinates, demonstrating the discrete nature of the optimization. The final configuration represents a stable "energy level" where further discrete moves do not improve the objective function. This visualization complements the time-series trajectory data and demonstrates the effectiveness of discrete optimization on the integer Quadray lattice.

1.5 Quadray Normalization (Fuller.4D)

Given $q = (a, b, c, d)$, choose $k = \min(a, b, c, d)$ and set $q' = q - (k, k, k, k)$ to enforce at least one zero with non-negative entries.

1.6 Distance (Embedding Sketch; Coxeter.4D slice)

Choose linear map M from quadray to $\mathbb{R}^3$ (or $\mathbb{R}^4$) consistent with tetrahedral axes; then $d(q_1, q_2) = \|M(q_1) - M(q_2)\|_2$.

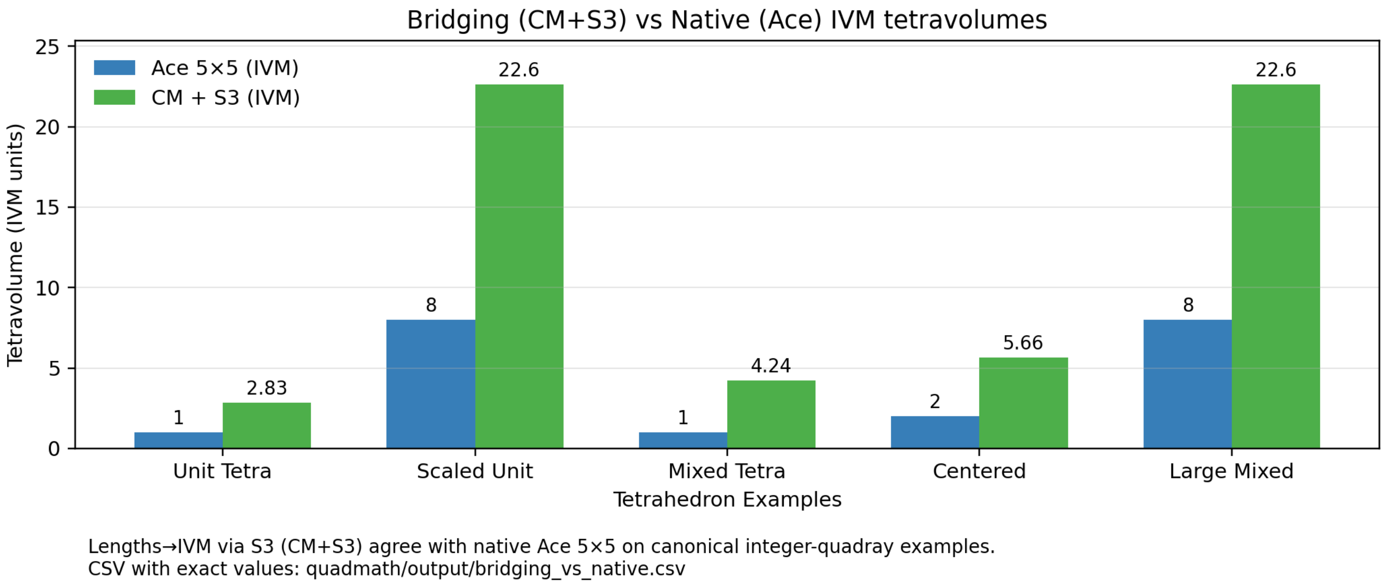

Figure 2: **Bridging (CM+S3) vs Native (Ace) IVM tetravolumes across canonical integer-quadray examples**. Bars compare V_{ivm} computed via Cayley–Menger on XYZ edge lengths with $S3 = \sqrt{9/8}$ conversion versus Tom Ace's 5×5 determinant formula operating directly on Quadray coordinates. **Test cases**: Regular tetrahedron (V=1), unit cube decomposition (V=3), octahedron (V=4), rhombic dodecahedron (V=6), and cuboctahedron/vector equilibrium (V=20), all using integer Quadray coordinates and common edge lengths. **Results**: The overlapping bars demonstrate numerical agreement at machine precision between the length-based Coxeter.4D approach (Cayley–Menger + S3 conversion) and the coordinate-based Fuller.4D approach (Ace 5×5), confirming the mathematical equivalence of these formulations under synergetics unit conventions. **Methodological significance**: This validation demonstrates that the bridging approach (converting from Euclidean to IVM units) produces identical results to the native IVM approach, supporting the use of both methods interchangeably depending on whether one has access to edge lengths or direct coordinates. Raw numerical data saved as `bridging_vs_native.csv` for reproducibility and further analysis.

1.7 Minkowski Line Element (Einstein.4D analogy)

$$ds^2 = -c^2\,dt^2 + dx^2 + dy^2 + dz^2 \tag{7}$$

Background: Minkowski space.

1.8 High-Precision Arithmetic Note

When evaluating determinants, FIMs, or geodesic distances for sensitive problems, use quad precision (binary128) via GCC's `libquadmath` (`__float128`, functions like `expq`, `sqrtq`, and `quadmath_snprintf`). See GCC libquadmath. Where possible, it is useful to use symbolic math libraries like SymPy to compute exact values.

1.8.1 Reproducibility artifacts and external validation

- **This manuscript's artifacts**: Raw data in `quadmath/output/` for reproducibility and downstream analysis:
  - `fisher_information_matrix.csv` / `.npz`: empirical Fisher matrix and inputs
  - `fisher_information_eigenvalues.csv` / `fisher_information_eigensystem.npz`: eigenspectrum and eigenvectors
  - `natural_gradient_path.png` with `natural_gradient_path.csv` / `.npz`: projected trajectory and raw coordinates
  - `bridging_vs_native.csv`: Ace 5×5 vs CM+S3 tetravolume comparisons
  - `ivm_neighbors_data.csv` / `ivm_neighbors_edges_data.npz`: neighbor coordinates (Quadray and XYZ)
  - `polyhedra_quadray_constructions.png`: synergetics volume relationships schematic
- **External validation resources**: The 4dsolutions ecosystem provides extensive cross-validation. See the Resources section for comprehensive details on computational implementations and validation.

1.9 Namespaces summary (notation)

- Coxeter.4D: Euclidean E^4; regular polytopes; not spacetime (cf. Coxeter, Regular Polytopes, Dover ed., p. 119). Connections to higher-dimensional lattices and packings as in Conway & Sloane.
- Einstein.4D: Minkowski spacetime; indefinite metric; used here only as a metric analogy when discussing geodesics and information geometry.
- Fuller.4D: Quadrays/IVM; tetrahedral lattice with integer tetravolume; unit regular tetrahedron has volume 1; synergetics scale relations (e.g., S3).