

# Resources

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

August 15, 2025

## Contents

<b>1</b>	<b>Resources</b>	<b>1</b>
1.1	Core Concepts and Background	1
1.1.1	Information Geometry and Optimization	1
1.1.2	Active Inference and Free Energy	2
1.2	Quadrays and Synergetics (Core Starting Points)	2
1.2.1	Introductory Materials	2
1.2.2	Historical and Background Materials	2
1.3	4dsolutions Ecosystem: Comprehensive Computational Framework	2
1.3.1	Core Computational Modules	2
1.3.2	Primary Hub: School_of_Tomorrow (Python + Notebooks)	2
1.3.3	Additional Repositories	3
1.3.4	Educational Framework and Curricula	3
1.3.5	Media and Publications	3
1.4	Community Discussions and Collaborative Platforms	3
1.4.1	Active Platforms	3
1.4.2	Historical Archives	3
1.5	Related Projects and Applications	3
1.5.1	Tetrahedral Voxel Engines	3
1.5.2	Academic Publications	4
1.6	Tooling and Technical Resources	4
1.6.1	High-Precision Arithmetic	4
1.7	Cross-Language and Cross-Platform Validation	4
1.7.1	Implementation Consistency	4
1.7.2	Context and Integration	4

## 1 Resources

This section provides comprehensive resources for learning about and working with Quadrays, synergetics, and the computational methods discussed in this manuscript.

### 1.1 Core Concepts and Background

#### 1.1.1 Information Geometry and Optimization

- **Fisher information:** [Fisher information \(reference\)](#) — see also Eq. (??) in the equations appendix
- **Natural gradient:** [Natural gradient \(reference\)](#) — see Eq. (??) in the equations appendix

### 1.1.2 Active Inference and Free Energy

- **Active Inference Institute:** [Welcome to Active Inference Institute](#)
- **Comprehensive review:** [Active Inference — recent review \(UCL Discovery, 2023\)](#)

## 1.2 Quadrays and Synergetics (Core Starting Points)

### 1.2.1 Introductory Materials

- **Quadray coordinates (intro and conversions):** [Urner - Quadray intro](#), [Urner - Quadrays and XYZ](#)
- **Synergetics background and IVM:** [Synergetics \(Fuller, overview\)](#)
- **Quadray coordinates overview:** [Quadray coordinates \(reference\)](#)

### 1.2.2 Historical and Background Materials

- **RW Gray projects — Synergetics text:** [rwgrayprojects.com \(synergetics\)](#)
- **Fuller FAQ:** [C. J. Fearnley's Fuller FAQ](#)
- **Synergetics resource list:** [C. J. Fearnley's resource page](#)
- **Wikieducator:** [Synergetics hub](#)
- **Quadray animation:** [Quadray.gif \(Wikimedia Commons\)](#)
- **Fuller Institute:** [BFI — Big Ideas: Synergetics](#)

## 1.3 4dsolutions Ecosystem: Comprehensive Computational Framework

The [4dsolutions organization](#) provides the most extensive computational framework for Quadrays and synergetic geometry, spanning 29+ repositories with implementations across multiple programming languages.

### 1.3.1 Core Computational Modules

#### Primary Python Libraries

- **Math for Wisdom (m4w):** [m4w \(repo\)](#)
  - **Quadray vectors and conversions:** [qrays.py \(Qvector, SymPy-aware\)](#)
  - **Synergetic tetravolumes and modules:** [tetravolume.py](#) [with PdF-CM vs native IVM and BEAST algorithms](#)

#### Cross-Language Validation

- **Rust implementation:** [rusty\\_rays](#) (performance-oriented)
  - Sources: [Rust library implementation](#), [Rust command-line interface](#)
- **Clojure implementation:** [synmods](#) (functional paradigm)
  - Sources: [qrays.clj](#), [ramping\\_up.clj](#)

### 1.3.2 Primary Hub: School\_of\_Tomorrow (Python + Notebooks)

**Repository:** [School\\_of\\_Tomorrow](#)

#### Core Modules

- [qrays.py](#): Quadray implementation with normalization, conversions, and vector ops ([source](#))
- [quadcraft.py](#): POV-Ray scenes for CCP/IVM arrangements, animations, and tutorials ([source](#))
- [flectegrity.py](#): Polyhedron framework, concentric hierarchy, POV-Ray export ([source](#))
- **Additional modules:** [polyhedra.py](#), [identities.py](#), [smod\\_play.py](#) (synergetic modules)

## Key Notebooks

- `Qvolume.ipynb`: Tom Ace 5×5 determinant with random-walk demonstrations ([source](#))
- `VolumeTalk.ipynb`: Comparative analysis of bridging vs native tetravolume formulations ([source](#))
- `QuadCraft_Project.ipynb`: 1,255 lines of interactive CCP navigation and visualization tutorials ([source](#))
- **Additional notebooks**: `TetraBook.ipynb`, `CascadianSynergetics.ipynb`, `Rendering_IVM.ipynb`, `SphereVolumes.ipynb` (visual and curricular materials)

### 1.3.3 Additional Repositories

#### Tetravolumes (Algorithms and Pedagogy)

- **Repository**: [tetravolumes](#)
- **Code**: `tetravolume.py`
- **Notebooks**: [Atoms R Us.ipynb](#), [Computing Volumes.ipynb](#)

#### Visualization and Rendering

- **BookCovers**: VPython for interactive educational animations ([repo](#))
  - Examples: `bookdemo.py`, `stickworks.py`, `tetravolumes.py`

### 1.3.4 Educational Framework and Curricula

#### Oregon Curriculum Network (OCN)

- **OCN portal**: [OCN portal](#)
- **Python for Everyone**: [pymath page](#)

#### Historical Documentation

- **Python5 notebooks**: [Polyhedrons 101.ipynb](#)
- **Historical variants**: `grays.py` also appears in [Python5 \(archive\)](#)
- **Python edu-sig archives**: [Python edu-sig archives](#) tracing 25+ years of development

### 1.3.5 Media and Publications

- **YouTube demonstrations**: [Synergetics talk 1](#), [Synergetics talk 2](#), [Additional](#)
- **Academia profile**: [Kirby Urner at Academia.edu](#)

## 1.4 Community Discussions and Collaborative Platforms

### 1.4.1 Active Platforms

- **Math4Wisdom**: [Collaborative platform](#) with curated IVM↔XYZ conversion resources and cross-reference materials
- **synergeo discussion archive**: [Groups.io platform](#) with ongoing community discussions and technical exchanges

### 1.4.2 Historical Archives

- **GeodesicHelp threads**: [GeodesicHelp computations archive \(Google Groups\)](#) documenting computational approaches and problem-solving techniques

## 1.5 Related Projects and Applications

### 1.5.1 Tetrahedral Voxel Engines

- **QuadCraft**: [Tetrahedral voxel engine using Quadrays](#)

### 1.5.2 Academic Publications

- **Flextegrity**: [Generating the Flextegrity Lattice \(academia.edu\)](#)

## 1.6 Tooling and Technical Resources

### 1.6.1 High-Precision Arithmetic

- **GCC libquadmath (binary128)**: [Official GCC libquadmath documentation](#)

## 1.7 Cross-Language and Cross-Platform Validation

### 1.7.1 Implementation Consistency

- **Rust (rusty\_rays)** and **Clojure (synmods)** mirror the Python algorithms for vector ops and tetravolumes, serving as independent checks on correctness and performance comparisons.
- **POV-Ray (quadcraft.py)** and **VPython (BookCovers)** demonstrate rendering pipelines for CCP/IVM scenes and educational animations.

### 1.7.2 Context and Integration

These materials popularize the IVM/CCP/FCC framing of space, integer tetravolumes, and projective Quadray normalization. They inform the methods in this paper and complement the `src/` implementations (see `quadray.py`, `cayley_menger.py`, `linalg_utils.py`).

The ecosystem provides extensive validation, pedagogical context, and practical implementations that complement and extend the methods developed in this manuscript. Cross-language implementations serve as independent verification of algorithmic correctness while educational materials demonstrate practical applications across diverse computational environments.