

P R E P R I N T

Online Encyclopedia of Fractals (OEF) Standard

Vincenzo Manto, James C. McMahon

June 2, 2026

DOI: <https://doi.org/10.5281/zenodo.20260602><https://github.com/Datastripes/OEF>

Abstract

This document specifies the Data Schema, Meta-syntax, and Execution Core for the Online Encyclopedia of Fractals (OEF) alongside its accompanying Domain-Specific Language, Fracta. OEF provides a plaintext, peer-submittable registry for mathematical fractals, architectural structures, and chaotic systems. It addresses the structural equivalence challenge (where disparate algorithmic generators produce identical geometric attractors) by indexing topologies via an automated box-counting Hausdorff dimension algorithm combined with a discrete, coordinate-normalized Jaccard intersection engine. This enables algorithmic duplicate prevention, deterministic cataloging, and the automatic classification of mathematical anomalies.

Keywords: math, Fractals

Status of This Memo

This technical specification outlines the infrastructure and grammar requirements for the deployment of the Online Encyclopedia of Fractals (OEF). Implementations claiming conformance with this specification **MUST** satisfy all criteria outlined in the state-machine and algorithmic definitions herein. Technical feedback is invited.

Table of Contents

1. Introduction and Rationale
2. Core Architectural Principles
3. Protocol Syntax & Technical Specification
 - 3.1. OEF Standard Metadata Fields
 - 3.2. Fracta Grammar Architecture
1. Reference Syntax Examples
2. Algorithmic Matching & Metric Definitions
 - 5.1. Visual Fingerprinting via Jaccard IoU
 - 5.2. Empirical Hausdorff Dimension Estimation
 - 5.3. Conflict Resolution & Mathematical Mimicry
1. Technical Matrix & Comparative Analysis
2. Persistence & Serialization Requirements

1. Introduction and Rationale

Existing online registries for mathematical phenomena, such as the Online Encyclopedia of Integer Sequences (OEIS), utilize deterministic plaintext indices to catalog complex mathematical objects. Geometrical fractals, however, represent infinitely recursive, multidimensional point-sets that cannot

be natively captured by sequential integer arrays.

Historical approaches to cataloging fractals rely on lossy rasterized image databases (e.g., PNG/JPEG archives). This methodology introduces structural inefficiencies:

1. **Algorithmic Opacity:** Static images isolate the visual output from the generative formula or ruleset.
2. **The Equivalence Problem:** Disparate mathematical frameworks (e.g., a Lindenmayer System versus an Iterated Function System) can converge on isomorphic geometric attractors. Comparing code strings or raw hashes fails to identify these relational mappings.

The Online Encyclopedia of Fractals (OEF) standard addresses these limitations by introducing **Fracta**, a multi-engine domain-specific language, integrated into a unified database format that isolates, normalizes, and indexes the underlying fractal geometry through its fractional dimensionality and pixel-occupancy vectors.

2. Core Architectural Principles

Compliant implementations of the OEF/Fracta ecosystem **MUST** conform to the following architectural design constraints:

- **Poly-Algorithmic Execution:** The compiler engine must handle string-rewriting systems, complex escape-time math, and affine transformation matrices under a single syntactic interpreter.
 - **Isomorphic Invariance:** Duplicate prevention **MUST** operate on the final spatial point-cloud rather than the source text, ensuring that identical geometries written via different engines are correctly cross-referenced.
 - **Metadata Integrity:** Every structural entry must be richly documented with historical, academic, and systemic metadata fields to mirror the rigorous cross-referencing capabilities of the OEIS.
 - **Deterministic Compactness:** Visual footprints used for searching and cataloging must be normalized to fixed-size sparse bitmasks, eliminating scale and translation noise while keeping search operations highly performant.
-

3. Protocol Syntax & Technical Specification

3.1. OEF Standard Metadata Fields

An OEF entry is represented by a sequence of record tags, structured to allow direct human readability and line-oriented stream parsing. Compliant records **MUST** implement the following specific header prefixes:

- **`%N` (Name):** Explicit, unique taxonomic name of the fractal attractor.
- **`%D` (Dimension):** The calculated or theoretical fractional/Hausdorff dimension.
- **`%C` (Comments):** Multi-line prose detailing structural, geometric, or historical behaviors.
- **`%R` (References):** Formal academic citations (Books, Journals, DOIs).
- **`%H` (Links):** Verifiable External Hyperlinks to mathematical resources.
- **`%K` (Keywords):** Standardized lookup tags used for indexing (e.g., `ifs`, `l_system`, `organic`).

3.2. Fracta Grammar Architecture

The underlying execution code within an entry is encapsulated under the programmatic blocks of the Fracta language. Execution routes are governed by the `ENGINE` directive, which alters the syntax state machine.

```
ENGINE <L_SYSTEM | PIXEL | IFS>
```

Engine Execution Configurations:

- 1. **`L\_SYSTEM` Mode:** Evaluates string substitution arrays using an axiom-rule loop.
 - Required Directives: `AXIOM`, `RULE <char> -> <string>`, `ANGLE <float>`, `ITER <int>`.
- 2. **`PIXEL` Mode:** Evaluates algebraic coordinate formulas over a continuous complex plane array.
 - Required Directives: `FORMULA <string>`, `X_RANGE <min max>`, `Y_RANGE <min max>`, `ITER <int>`.
- 3. **`IFS` Mode:** Computes the Chaos Game via stochastic linear transformations.
 - Required Directives: `RULE <prob a b c d e f>`.

4. Reference Syntax Examples

The following valid OEF payloads demonstrate standard formatting, metadata populations, and multi-engine scripting.

Example 1: F000001 (Iterated Function System)

```
=====
Ø<ß  ONLINE ENCYCLOPEDIA OF FRACTALS (OEF) – ENTRY F000001
=====
%N Name:      Barnsley Fern
%D Hausdorff: D_H = 1.7214
%C Comments:
    - Simula con incredibile precisione la fisionomia della felce nera
    (Asplenium adiantum-nigrum).
    - Uno dei primi e più famosi esempi di frattale generato tramite
    sistemi di funzioni iterate (IFS).
%D Code (Fracta):
    ENGINE IFS
    ITER 50000
    RULE 0.01  0.0  0.0  0.0  0.16  0.0  0.0
    RULE 0.85  0.85  0.04 -0.04  0.85  0.0  1.6
    RULE 0.07  0.2  -0.26  0.23  0.22  0.0  1.6
    RULE 0.07  -0.15  0.28  0.26  0.24  0.0  0.44
%R References:
    - Barnsley, Michael F. (1988). Fractals Everywhere. Academic Press.
%H Links:
    - https://en.wikipedia.org/wiki/Barnsley_fern
%K Keywords:
    ifs, botanical, barnsley, chaos-game, nature
=====
```

Example 2: F000002 (Escape-Time Geometry)

```

=====
Ø<β  ONLINE ENCYCLOPEDIA OF FRACTALS (OEF) – ENTRY F000002
=====
%N Name:      Classic Mandelbrot Set Boundary
%D Hausdorff: D_H = 2.0000
%C Comments:
- Il locus dei punti nel piano complesso per cui l'orbita di 0 sotto la
mappa quadratica non diverge.
- Shishikura ha dimostrato che il boundary di questo insieme ha
dimensione di Hausdorff pari a 2.
%D Code (Fracta):
ENGINE PIXEL
FORMULA z**2 + c
X_RANGE -2.0 1.0
Y_RANGE -1.5 1.5
RES 500
ITER 100
%R References:
- Shishikura, M. (1998). "The Hausdorff dimension of the boundary of
the Mandelbrot set is 2."
%H Links:
- https://mathworld.wolfram.com/MandelbrotSet.html
%K Keywords:
pixel, escape-time, mandelbrot, polynomial, complex
=====

```

5. Algorithmic Matching & Metric Definitions

To ensure automatic data sanitization upon submission, the OEF compiler routes every incoming script through a decoupled background rendering matrix to extract physical geometric properties.

5.1. Visual Fingerprinting via Jaccard IoU

The generated point-set is bound-normalized and mapped to a discrete binary grid $M \in \{0, 1\}^{64 \times 64}$. Spatial occupancy similarity between a new submission A and an existing entry B is computed via the Intersection over Union (IoU) index:

$$\text{IoU}(A, B) = \left(\frac{|A \cap B|}{|A \cup B|} \right) \times 100$$

A score of $\text{IoU} \geq 75\%$ indicates a high degree of spatial correlation, triggering structural cross-checks.

5.2. Empirical Hausdorff Dimension Estimation

The database evaluates structural roughness by mapping spatial complexity trends across multiple resolutions. Compliant parsers compute the Minkowski-Bouligand (Box-Counting) dimension by partitioning the binary footprint matrix across decreasing box scale-factors ϵ :

$$D_H = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$

Where $N(\epsilon)$ represents the total number of grid sub-matrices of linear dimension ϵ containing at least one positive cell allocation.

5.3. Conflict Resolution & Mathematical Mimicry

When an incoming submission matches an existing entry with an $\text{IoU} \geq 75\%$, the validation engine compares the respective Hausdorff values:

```
Let Delta_D = |D_H(Submission) - D_H(Existing)|  
  
IF IoU >= 75% AND Delta_D <= 0.15 THEN:  
    REJECT (Submission is a structural duplicate or scale variant)  
  
IF IoU >= 75% AND Delta_D > 0.15 THEN:  
    ACCEPT AS ANOMALY (Phenomenon of Mathematical Mimicry detected)  
    FLAG RECORD: "Anomalia Investigativa"
```

***Note on Mathematical Mimicry:** This state indicates that two different mathematical formulas construct a visually similar spatial footprint, yet exhibit completely divergent scale behavior and fractional densities. This configuration is flag-tracked as highly significant for research purposes.*

6. Technical Matrix & Comparative Analysis

The following matrix contrasts operational features across disparate entry formats for cataloging fractal structures:

Capabilities	Raster Files (PNG/JPG)	Pure Source Code	OEF Standard v1.0
Plaintext Portability	No	Yes	Yes
Isomorphic Code Matching	No	No	Yes
Multi-Engine Encapsulation	No	No	Yes
Automated Geometry Indexing	No	No	Yes
Integrated Academic Metadata	No	No	Yes
Git Delta Tracking Optimization	No	Yes	Yes

7. Persistence & Serialization Requirements

Compliant OEF database instances **MUST** implement a deterministic byte-stream serialization format. Files written to disk **SHOULD** encapsulate the global state using a compressed tuple dictionary framework containing:

1. A unique hash index map linking alphanumeric IDs (FXXXXXX) to structural schema records.
2. An integer sequence pointer managing incremental ID allocation (`next_id`).
3. Binary-packed multi-dimensional NumPy arrays representing the spatial footprints to guarantee sub-millisecond logical bitwise evaluations during global database queries.