Getting Started with HyperSync Core

This guide will help you get started with HyperSync Core’s 43 geometric operations.

Installation

```
pip install hypersync-core
```

Or from source:

```
git clone https://github.com/emstar-en/HyperSync.git
cd HyperSync
pip install -e .
```

Basic Usage

1. Hyperbolic Geometry

```python
import numpy as np
from hypersync_core.geometry import hyperbolic_distance, hyperbolic_exp_map

# Create points in Poincaré ball
x = np.array([0.1, 0.2, 0.3])
y = np.array([0.4, 0.1, 0.2])

# Compute distance
dist = hyperbolic_distance(x, y)
print(f"Distance: {dist:.4f}")
```

2. Spherical Geometry

```python
from hypersync_core.geometry import spherical_distance, spherical_interpolation

# Points on sphere
x = np.array([1.0, 0.0, 0.0])
y = np.array([0.0, 1.0, 0.0])

# Great circle distance
dist = spherical_distance(x, y)  # π/2

# Interpolate
mid = spherical_interpolation(x, y, t=0.5)
```

3. Byzantine Consensus

```
from hypersync_core.consensus import spherical_bft_consensus

# Node proposals
proposals = [np.random.randn(3) for _ in range(10)]

# Run consensus (O(n), 1000x faster than PBFT)
consensus, byzantine = spherical_bft_consensus(proposals)
```

4. Security

```
from hypersync_core.security import hyperbolic_encrypt, hyperbolic_decrypt

# Encrypt with geometric scrambling
encrypted, nonce = hyperbolic_encrypt(data, key)
decrypted = hyperbolic_decrypt(encrypted, nonce, key)
```

5. Fast Approximations

```
from hypersync_core.heuristics import ricci_flow_ultra_fast

# Ultra-fast Ricci flow (10,000x speedup)
points = [np.random.randn(3) for _ in range(100)]
flowed = ricci_flow_ultra_fast(points)
```

Next Steps

- Read the API Reference (API.md)
- Explore examples/ (../examples/)
- See CORE_TIER_OPERATIONS.md (CORE_TIER_OPERATIONS.md) for all 43 operations