## Section 7. Numerical Computation Framework

Sections 5 and 6 specified the continuous PDE structure and the multi-scale, topological behavior of the entropic manifold $\mathcal{M}^{E}$. To make these structures operational, C-7 must implement a numerical computation framework capable of:

Representing $\mathcal{M}^{E}$ in a discrete form,

Integrating the collapse PDEs in entropic time $\tau$,

Detecting and handling stiffness, bifurcations, and topology changes (conifold pinches, wormholes),

Preserving, as far as possible, the geometric and topological information encoded in the continuous equations.

Section 7 outlines this framework at a conceptual level. The intention is not to fix a single numerical method, but to define the requirements and structure that any compatible solver must satisfy.

### 7.1 Discretization of $\mathcal{M}^{E}$ (Grid / FEM / Graph Manifold)

The numerical treatment begins by choosing a discrete representation of the entropic manifold $\mathcal{M}^{E}$. Three broad families of discretization are relevant:

Regular grid (lattice) representations,

Finite Element Method (FEM) meshes,

Graph-manifold representations.

Each has advantages and drawbacks. In CUWF, the geometry of $\mathcal{M}^{E}$ can be highly curved, multi-connected, and dynamic in topology; therefore, the framework must allow multiple discretization modes and, where necessary, hybridization between them.

### 7.1.1 Regular grid (lattice) discretization

In regimes where $\mathcal{M}^{E}$ is approximately flat or can be covered by a small number of coordinate patches with mild curvature, a regular grid (or mildly deformed structured grid) is efficient:

Choose a coordinate chart $\{x_i\}$ on a patch of $\mathcal{M}^{E}$,

Discretize each coordinate into $N_i$ points, forming an $N_1 \times \cdots \times N_d$ lattice,

Approximate derivatives in the PDE via finite differences (central, upwind, or higher order schemes).

Pros:

Simple implementation,

Efficient memory access and parallelization,

Straightforward coupling with standard PDE solvers.

Cons:

Poor adaptability to strong curvature gradients, conifold pinches, and topological defects,

Difficult to represent highly nontrivial connectivity or wormhole-like links.

Thus, regular grids are best used for testing, toy models, and locally Euclidean regimes of $\mathcal{M}^{E}$.

### 7.1.2 FEM (Finite Element) mesh discretization

For more general geometries, a finite element approach is more natural:

Represent $\mathcal{M}^{E}$ (or a large patch) as a simplicial or polyhedral mesh (triangles, tetrahedra, etc.),

Attach field values ($\Phi$, g, X, T, $\Xi$, …) to nodes or elements, Assemble stiffness and mass matrices to discretize differential operators ($\nabla$, $\nabla\cdot$, Laplacians, curvature operators) in weak form.

Pros:

Adapts to curvature and nontrivial shape,

Supports local refinement near regions of interest (e.g., near pinch points),

Well-developed mathematical theory for error estimates and stability.

Cons:

More complex data structures and assembly,

Mesh management becomes nontrivial under strong topology change (pinch, tear, wormhole).

FEM is the natural choice for global simulations where the geometry of $\mathcal{M}^{\mathrm{E}}$ is strongly non-Euclidean but still reasonably smooth and where topology changes are rare or controlled.

### 7.1.3 Graph manifold representation

For regimes dominated by basin structure, conifolds, and wormhole connections, it is often more efficient to represent $\mathcal{M}^{\mathrm{E}}$ as a graph manifold:

Nodes represent basins, sub-basins, or local patches of $\mathcal{M}^{\mathrm{E}}$,

Edges represent adjacency, transition channels, or wormhole links,

Edge weights encode effective entropic gradients, barrier heights, or entanglement strengths.

Differential operators on $\mathcal{M}^{\mathrm{E}}$ are approximated by graph Laplacians and related discrete operators. The PDE becomes a system of ODEs on node values plus flux terms on edges.

Pros:

Topology manipulation (adding/removing edges, splitting/merging nodes) is straightforward,

Well suited for modeling basin dynamics, phase transitions, and nonlocal wormhole coupling, Scales nicely when the manifold is “coarse” in terms of basins but “complex” in connectivity.

Cons:

Loses detailed geometric information inside nodes unless coupled with sub-solvers,

Requires careful calibration to match continuous PDE behavior.

In practice, the C-7 framework should support hybrid representations, e.g.:

FEM or grid for local patches,

Graph manifold for global basin connectivity,

Dynamic switching between representations as the simulation focus changes.

### 7.2 Temporal Integrators for Collapse PDE

The collapse dynamics in $\tau$ are governed by PDEs that can be nonlinear, stiff, and multi-scale. Choosing appropriate temporal integrators is crucial.

Let $U(\tau)$ denote a vector collecting all discretized fields ($\Phi$, X, g, T, $\Xi$, etc.). The PDE system can be schematically written as:

$dU / d\tau = F(U(\tau))$

where F encodes entropic drift, diffusion-like terms, curvature evolution, and entanglement coupling.

#### 7.2.1 Basic integrator families

Explicit integrators (e.g., forward Euler, Runge–Kutta):

Simple and easy to implement,

Applicable when the system is non-stiff or stiffness is mild,

Require small time steps $\Delta\tau$ controlled by the CFL-type stability condition.

Implicit integrators (e.g., backward Euler, implicit Runge–Kutta, BDF):

Stable under larger time steps for stiff components (diffusion-like terms, rapid collapse directions),

Require solving linear/nonlinear systems at each step,

More computationally expensive per step but allow fewer steps overall.

IMEX (Implicit–Explicit) schemes:

Split F into stiff and non-stiff parts: F = F_stiff + F_nonstiff,

Treat F_stiff implicitly and F_nonstiff explicitly,

Provide a compromise between stability and efficiency.

### 7.2.2 Requirements for CUWF collapse dynamics

The temporal integrators must:

Preserve monotonic decay of entropic functionals (or at least not artificially increase them),

Accurately capture critical events (bifurcations, pinches, wormhole threshold crossings),

Handle widely separated timescales: slow macro-evolution vs. fast micro-collapse.

Typical strategy:

Use adaptive time-stepping, where $\Delta\tau$ is adjusted based on local error estimates and stiffness indicators,

Employ IMEX or fully implicit methods in regions recognized as stiff (Section 7.3),

Use higher-order methods (e.g., RK4, higher-order BDF) where smooth evolution dominates and high accuracy is desired.

### 7.3 Detection of Stiff Regions & Solver Switching

The collapse PDE is not uniformly stiff across $\mathcal{M}^E$ or across $\tau$. Stiffness is typically localized around:

Strong entropic gradients (steep basins),

Regions close to topology change (pinch, rapid bifurcation),

Zones of intense entanglement-driven synchronization.

To integrate efficiently and robustly, the numerical framework must:

Detect stiff regions dynamically,

Switch to appropriate solvers (or solver modes) in those regions.

#### 7.3.1 Stiffness indicators

Several types of indicators can be used:

Spectral indicators:

Estimate the eigenvalues of the local Jacobian $J = \partial F/\partial U$. Large negative eigenvalues indicate fast-decaying modes (stiff directions).

Gradient-based indicators:

Monitor the magnitude of entropic gradients $\|\nabla\Phi\|$, curvature invariants (e.g., $|R|$, $|\det T|$), and derivatives $\partial\tau$ of these quantities. Rapid changes signal stiffness.

Time step rejection:

Frequent step rejections or very small $\Delta\tau$ enforced by stability constraints in a region indicate local stiffness.

### 7.3.2 Solver switching

Once stiff regions are identified, the framework can:

Locally change the integration scheme (e.g., switch from explicit to implicit/IMEX) for those degrees of freedom,

Subcycle micro-steps inside stiff regions while using larger steps elsewhere,

Reduce spatial resolution adaptively to smooth out stiff behavior in noncritical zones (with care not to miss small-scale events that matter).

The goal is to balance:

Accuracy of critical collapse dynamics,

Computational cost,

Numerical stability.

In CUWF, this is particularly important near:

Bifurcation points where $\lambda_{\min} \to 0$,

Conifold pinches where $\det T \to 0$,

Wormhole onset where $\Xi_{\text{eff}}$ crosses the critical threshold.

## 7.4 Bifurcation Tracking and Attractor Snapping

Section 5.2 described how soft-mode bifurcations generate new branches of attractors. Numerically, these events must be:

Detected,

Tracked, and

Represented as discrete attractor changes (“snap” of the system into a new basin or branch).

### 7.4.1 Bifurcation tracking

Tracking bifurcations involves:

Monitoring critical eigenvalues of local stability tensors (e.g., T, H).

When $\lambda_{min}$ approaches zero and changes sign, a bifurcation is imminent.

Following the associated eigenvector $v_{soft}$ in parameter space:

This direction defines the emerging branches.

Continuing solutions along new branches using continuation methods:

Predict–correct schemes (e.g., pseudo-arclength continuation) to follow solutions as $\tau$ or another control parameter changes.

In the CUWF context, the control parameter is often $\tau$ itself, but effective control parameters may also include macro-curvature $\bar{R}$, entanglement strengths, or other coarse quantities.

### 7.4.2 Attractor snapping

From the perspective of a collapse trajectory, a bifurcation appears as a sudden re-selection of attractor:

Before bifurcation: trajectory converges to a single basin center,

After bifurcation: there are multiple possible basin centers; the trajectory must “choose” one.

Numerically, this requires a snapping rule:

When the system enters a region where multiple stable attractors coexist and the dynamics is strongly influenced by microscopic fluctuations ($\eta$ terms), the solver should project the trajectory onto one of the attractors according to:

Local deterministic biases (e.g., slight asymmetries in $\Phi$ or T), Stochastic sampling (if a probabilistic layer is included),

Initial condition sensitivities encoded in the numerical state.

In a graph-manifold representation:

Attractor snapping corresponds to relabeling the node membership of the trajectory state: the system jumps from “branch-neutral” node to a specific branch node.

This snapping:

Must be consistent with conservation rules (e.g., probability mass conservation in ensembles),

Must be recorded in a history structure to analyze branching statistics and reconstruct collapse trees.

### 7.5 Conifold Detection & Wormhole Mapping

Finally, the solver must be able to handle topology change events: conifold pinches and wormhole formation as described in Section 5.3 and 5.4. Numerically, this involves:

Detecting when a conifold pinch is forming,

Deciding whether it evolves into a basin separation or a wormhole bridge,

Updating the discrete representation of $\mathcal{M}^E$ accordingly.

#### 7.5.1 Conifold detection

Conifold detection relies on monitoring invariants such as:

det T(x, $\tau$) and its derivative $\partial\tau$(det T),

Local geometric measures (e.g., minimal cross-sectional area A_min in the mesh or graph),

Curvature blow-up indicators (e.g., $|\mathcal{R}|$ exceeding a threshold).

A region is flagged as a candidate conifold if:

det T falls below a small tolerance ε_T (approaching zero),

The cross-sectional area or distance between two manifold regions shrinks below a geometric tolerance ε_geo,

The rate of change of these quantities indicates continued collapse (not just noise).

Once identified, the solver:

Locally refines the mesh or graph representation,

Focuses time stepping (possibly with smaller Δτ and implicit methods) to resolve the event.

### 7.5.2 Wormhole mapping

If, in addition to a conifold pinch, the entanglement strength between the two sides A and B of the neck satisfies:

Ξ_eff(A, B; τ) > Ξ_c

then the event is classified as a wormhole formation rather than a simple topological separation.

Numerically, this requires:

Evaluating Ξ_eff(A, B; τ) via integration or summation over nodes/regions in A and B,

Comparing it to the critical threshold Ξ_c,

If condition holds, introducing nonlocal connectivity in the discrete structure:

In an FEM or grid representation: add extra coupling terms in F(U) linking the degrees of freedom on both sides, with weights given by kernels like K Ξ in Section 5.4.

In a graph manifold: insert edges between nodes in A and B with weights determined by Ξ_eff and the local neck geometry.

This process is the wormhole mapping:

It maps the geometric conifold neck to explicit nonlocal links in the computational graph,

It ensures that subsequent collapse dynamics respect the wormhole transfer rules (synchronizing pull, mirrored collapse states).

### 7.5.3 Topology update and healing

After a conifold or wormhole event, the topology of $\mathcal{M}^E$ may:

Permanently change (e.g., separation into two components),

Temporarily deform and then “heal” (neck re-expands, wormhole collapses),

Settle into a new stable configuration with altered connectivity.

The solver must:

Update connectivity structures (mesh adjacency, graph edges),

Recompute coarse-grained fields (e.g., macro-curvature $\bar{R}$, basin registry, wormhole registry),

Adjust DOF counts and renormalization flows (Section 6.4) to reflect the new topology.

This maintains internal consistency between geometry, PDE dynamics, and topology in the numerical representation.

## 7.6 Recommended Kernel Families for $\Xi$_eff Nonlocal Coupling

To remove ambiguity for implementation in C-7, we specify canonical kernel options for computing $\Xi$_eff:

$$\Xi\_\mathrm{eff}{}^{I}{}_{j}(x,x';\tau) = \int K^{I}{}_{j}(|x-x'|;\ell(\tau))\,\psi(x',\tau)\,dx'$$

with selectable kernel families:

| Kernel Type | Formula | Behavior | When to Use |
|---|---|---|---|
| Gaussian | $K \propto \exp(-\lvert x-x'\rvert^2 / 2\ell^2)$ | Smooth, rapidly decaying | Quantum-like gentle spread |
| Lorentzian | $K \propto 1/(1 + \lvert x-x'\rvert^2 / \ell^2)$ | Longer tail | Macro-scale entanglement transport |
| 1/r $\alpha$-kernel | $K \propto 1/\lvert x-x'\rvert$^$\alpha$ | Strong long-range, fractal links | Wormhole regime / nonlocal bursts |

Default kernel for C-7 = Gaussian with adaptive $\ell(\tau)$

Optional switching rule: if wormhole ⟶ transition Gaussian ⟶ power-law kernel.

## Summary of Section 7

Section 7 specifies the numerical backbone needed to implement CUWF's PDE and topology dynamics:

7.1: Flexible discretization of $\mathcal{M}^E$ (grid, FEM, graph manifold, or hybrids),

7.2: Robust temporal integration of stiff, nonlinear collapse PDEs in $\tau$,

7.3: Dynamic detection of stiff regions and solver switching to maintain efficiency and stability,

7.4: Systematic tracking of bifurcations and attractor snapping to represent branching,

7.5: Detection of conifold pinches and explicit wormhole mapping in the discrete representation.

With Sections 5–7 combined, C-6 fully prepares the ground for C-7: CUWF Numerical Simulator, where these principles become concrete algorithms and implementations.