Continuum: Grid, Topology, and Geometry

1 Purpose and High-Level View

This document describes the core mathematical roles of three systems in the `continuum` project:

- **Grid**: represents the *computational domain* (logical index space) and stores discrete fields.
- **Topology**: represents *connectivity* between discrete entities (neighbors, boundaries, patch seams).
- **Geometry**: represents the *mapping* between computational and physical space and provides metric quantities (volumes, face area-vectors) required by the solver.

The central design goal is to keep the finite-volume solver largely independent of the physical coordinate system or embedding: if the solver can query *neighbor relations* (Topology) and *geometric measures* (Geometry), it can advance conservative PDEs on many kinds of domains (Cartesian, curvilinear, multi-block, cubed-sphere, etc.).

2 Computational vs Physical Domain

Let $\boldsymbol{\xi}$ denote *computational coordinates* (logical coordinates), e.g.

$$\boldsymbol{\xi} = (\xi, \eta, \zeta) \in \hat{\Omega},$$

where $\hat{\Omega}$ is typically a simple structured domain such as $[0, 1]^d$. The **Grid** discretises $\hat{\Omega}$ with a structured index space (i, j, k).

Let $\mathbf{x}$ denote *physical coordinates* in the real domain $\Omega \subset \mathbb{R}^d$:

$$\mathbf{x} = (x, y, z) \in \Omega.$$

The **Geometry** provides a map

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}),$$

so that each logical cell in $\hat{\Omega}$ corresponds to a (possibly curved or distorted) physical control volume in Ω.

3 Finite Volume Update: What the Solver Needs

Consider a conservative PDE in physical space,

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F}(U) = S(U, \mathbf{x}, t),$$

where U is a vector of conserved quantities, $\mathbf{F}$ is a flux, and S is a source term.

Finite volume methods evolve *cell averages* over physical control volumes V_i:

$$\bar{U}_i(t) = \frac{1}{|V_i|} \int_{V_i} U(\mathbf{x}, t)\, dV.$$

A standard semi-discrete FV form is:

$$\frac{d}{dt}\left(|V_i|\,\bar{U}_i\right) \;+\; \sum_{f\in\partial V_i} \int_{A_f} \mathbf{F}(U)\cdot\mathbf{n}\,dA \;=\; \int_{V_i} S\,dV.$$

In practice, the solver typically needs:

- **Cell volume** $|V_i|$
- **Face area-vector** $\mathbf{S}_{f,i} := \mathbf{n}_{f,i}\,|A_f|$ (physical normal times physical face area, oriented outward from cell i)
- **Neighbor info** for each face: which cell or boundary is adjacent to the face

The update step can be written schematically as

$$\bar{U}_i^{n+1} = \bar{U}_i^{n} - \frac{\Delta t}{|V_i|} \sum_{f\in\partial V_i} \widehat{\mathbf{F}}_f \cdot \mathbf{S}_{f,i} \;+\; \Delta t\,\bar{S}_i,$$

where $\widehat{\mathbf{F}}_f$ is a numerical flux (e.g. from a Riemann solver), computed using left/right states provided by Topology and reconstruction.

4 Grid: The Computational Domain

Role

The **Grid** is the discrete representation of the computational domain $\hat{\Omega}$:

- Defines the structured index space (2D or 3D), e.g. (i, j, k) with extents (N_x, N_y, N_z).
- Owns storage for discrete fields (cell-centered, face-centered, etc.).
- Provides iteration order and memory layout (AoS/SoA), ghost layers, and views/slices.

Key Idea

A cell index (i, j, k) is *not* automatically a physical cube in Ω. It is a *logical* cell in $\hat{\Omega}$; its physical shape and size are provided by Geometry.

5 Topology: Connectivity and Boundaries

Role

The **Topology** provides the relationships needed to traverse the grid as a control-volume complex:

- For each cell, enumerates its faces and adjacent entities.
- For each face, identifies the neighbor cell across the face, or a boundary condition object (wall, inflow, outflow, periodic, etc.).
- For multi-block grids (e.g. cubed-sphere), encodes cross-block connections and index mappings at seams.

Key Idea

Topology is *purely combinatorial*: it answers "who is adjacent to whom?" without needing to know physical distances or areas. This lets the solver walk the mesh consistently, including at boundaries and patch interfaces.

6 Geometry: Mapping and Metric Quantities

Role

The **Geometry** supplies physical meaning to logical grid entities by exposing metric quantities derived from the mapping $\mathbf{x}(\boldsymbol{\xi})$:

- Computes/caches physical **cell volumes** $|V_i|$.
- Computes/caches physical **face area-vectors** $\mathbf{S}_{f,i}$.
- Optionally provides physical cell/face centers $\mathbf{x}_i$, $\mathbf{x}_f$ for source terms and reconstruction.

Jacobian Viewpoint

Let J be the Jacobian of the mapping:

$$J = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}.$$

Then physical volume elements relate via

$$dV = |\det J| \; d\boldsymbol{\xi}.$$

In structured mapped grids, Geometry uses J (and related cofactor data) to build the FV measures $|V_i|$ and $\mathbf{S}_{f,i}$ that appear in the flux balance.

Key Idea

The solver should not need to know whether it is on Cartesian space, polar coordinates, or a cubed-sphere patch. If Geometry returns correct $(|V_i|, \mathbf{S}_{f,i})$, the same conservation update applies.

7 Putting It Together

At a high level, a solver timestep looks like:

1. **Grid**: access current fields $\bar{U}_i^n$.
2. **Topology**: for each cell i, iterate its faces and obtain neighbor/boundary information.
3. **Geometry**: for each face and cell, query $\mathbf{S}_{f,i}$ and $|V_i|$ (and optionally centers).
4. **Solver**: reconstruct left/right states, compute numerical flux $\widehat{\mathbf{F}}_f$, accumulate $\widehat{\mathbf{F}}_f \cdot \mathbf{S}_{f,i}$, divide by $|V_i|$, update $\bar{U}_i$.

This separation supports extending `continuum` from simple Cartesian CFD to mapped domains (curvilinear grids, multi-block cubed-sphere) and, with richer Geometry, to more general coordinate systems where metric factors are required.