Triangular Finite-Volume Grid: Construction, Tables, and Use for the Laplace Equation

1 Purpose

This document describes the general idea behind constructing and using a two-dimensional *triangular* grid for a *cell-centered finite volume (FV)* method. It explains:

- the minimal mesh representation,
- the geometric and connectivity quantities required for FV discretization,
- the meaning of every column in the two tables produced by the MATLAB script,and how these data are used to discretize the Laplace/diffusion operator on a generally *non-orthogonal* unstructured mesh.

In a cell-centered FV method:

- Each triangle is a *control volume* (cell).
- A scalar unknown (e.g. ϕ or temperature T) is stored *per triangle*, typically at the triangle centroid.
- The discrete operator is built from *fluxes across cell faces*. In 2D triangles, faces are edges.

2 Minimal mesh description

A triangular mesh in 2D is defined by the pair $(\mathbf{p}, \mathbf{t})$:

- **Node coordinates p**:

$$\mathbf{p}(k) = (x_k, y_k), \qquad k = 1, \ldots, N_{\text{nodes}}.$$

- **Triangle connectivity t**:

$$\mathbf{t}(i) = (v_1, v_2, v_3), \qquad i = 1, \ldots, N_{\text{tri}},$$

where each v_ℓ is an integer node index referencing $\mathbf{p}$.

From $(\mathbf{p}, \mathbf{t})$ one can compute centroids, edge lengths, normals, and neighbors.

3 Consistent triangle orientation (CCW)

Finite volume fluxes require a consistent definition of the *outward* normal direction. This is achieved by enforcing that each triangle (v_1, v_2, v_3) is ordered **counter-clockwise (CCW)**.

For triangle (v_1, v_2, v_3), define the signed double area:

$$2A = (x_{v_2} - x_{v_1})(y_{v_3} - y_{v_1}) - (y_{v_2} - y_{v_1})(x_{v_3} - x_{v_1}).$$

If $2A < 0$ the ordering is clockwise; swapping $v_2 \leftrightarrow v_3$ makes it CCW. After this step, "outward normals" can be computed consistently across the entire mesh.

4 Cell centers: triangle centroids

In a *cell-centered* FV method, each triangle i has a representative point (cell center), most commonly the centroid:

$$\mathbf{c}_i = (c_{x,i}, c_{y,i}) = \frac{1}{3}\Big(\mathbf{p}(v_1) + \mathbf{p}(v_2) + \mathbf{p}(v_3)\Big).$$

The discrete unknown is interpreted as

$$\phi_i \approx \phi(\mathbf{c}_i).$$

5 Local edges and outward unit normals

Each triangle has three *local edges*, named by vertex pairs:

$$12 : (v_1, v_2), \qquad 23 : (v_2, v_3), \qquad 31 : (v_3, v_1).$$

Edges are treated as *directed* segments consistent with CCW ordering: $v_1 \to v_2$, $v_2 \to v_3$, $v_3 \to v_1$.

For a directed edge vector $\mathbf{e} = (dx, dy)$ on a CCW triangle, the outward *unit* normal is

$$\mathbf{n}_{\text{out}} = \frac{1}{\|\mathbf{e}\|}(dy, -dx).$$

Thus each triangle stores three outward unit normals:

$$\mathbf{n}_{12,i}, \quad \mathbf{n}_{23,i}, \quad \mathbf{n}_{31,i}.$$

6 Immediate neighbors and boundary edges

FV diffusion/Laplacian discretization couples adjacent cells across shared faces (edges). Therefore, for each triangle and each local edge, we need the index of the triangle on the other side of that edge.

The neighbor construction uses edge matching:

- Each triangle contributes three edges.
- A geometric edge is represented by the sorted node pair $(\min(n_a, n_b), \max(n_a, n_b))$.
- If a sorted pair appears twice, it is an interior edge shared by two triangles.
- If it appears once, it is a boundary edge.

Boundary edges have no neighbor triangle and are treated using boundary conditions.

7 Unique edge (face) list

For many FV implementations it is convenient to store each geometric edge once in a global list. This avoids double counting interior faces and simplifies assembly. For each unique edge, the script stores:

- its two node indices (n_1, n_2),
- the adjacent triangles (`triL` and `triR`),
- the edge length L_e,
- the edge midpoint $\mathbf{m}_e$,
- the outward unit normal with respect to triangle `triL`.

If the edge lies on the boundary, `triR` is set to 0.

8 Meaning of all table columns produced by the script

The MATLAB script displays two tables (first 10 rows). This section defines every column name.

8.1 Triangle (cell) table: one row per triangle

- `triID`: triangle (cell) index.
- `cx`, `cy`: centroid coordinates (c_x, c_y) of this triangle.
- `v1`, `v2`, `v3`: the three vertex node indices (v_1, v_2, v_3) of this triangle.
- `nbr12`: neighbor triangle index across local edge (v_1, v_2); equals `0` if this edge is on the boundary.
- `nbr23`: neighbor across edge (v_2, v_3); equals `0` on the boundary.
- `nbr31`: neighbor across edge (v_3, v_1); equals `0` on the boundary.
- `n12x`, `n12y`: components of the outward unit normal $\mathbf{n}_{12,i}$ for edge 12 of this triangle.
- `n23x`, `n23y`: components of the outward unit normal $\mathbf{n}_{23,i}$ for edge 23.
- `n31x`, `n31y`: components of the outward unit normal $\mathbf{n}_{31,i}$ for edge 31.

8.2 Unique edge (face) table: one row per geometric edge

- `edgeID`: unique edge index.
- `n1`, `n2`: the two node indices defining the edge.
- `triL`, `triR`: the triangle indices adjacent to the edge. For boundary edges, `triR`=`0`.
- `len`: the edge length $L_e = \|\mathbf{p}(n_2) - \mathbf{p}(n_1)\|$.
- `mx`, `my`: midpoint coordinates $(m_x, m_y) = \frac{1}{2}(\mathbf{p}(n_1) + \mathbf{p}(n_2))$.
- `nLx`, `nLy`: components of the outward unit normal with respect to triangle `triL` for this edge.

9 Using the grid for the FV Laplace operator

9.1 Continuous problem

The Laplace equation (steady diffusion) is

$$\nabla^2 \phi = 0 \quad \text{in } \Omega,$$

with boundary conditions on $\partial\Omega$, such as Dirichlet $\phi = \phi_b$ and/or Neumann $\partial\phi/\partial n = g$.

9.2 Finite-volume integral form

Integrate over a control volume (triangle) Ω_i and apply the divergence theorem:

$$\int_{\Omega_i} \nabla^2 \phi \, dA = \int_{\partial\Omega_i} \nabla\phi \cdot \mathbf{n} \, ds.$$

Approximating the boundary integral by a sum over the three edges gives

$$\int_{\partial\Omega_i} \nabla\phi \cdot \mathbf{n} \, ds \approx \sum_{f \in \{12,23,31\}} (\nabla\phi \cdot \mathbf{n}_{i,f})_f \; L_{i,f},$$

where $L_{i,f}$ is the edge length and $\mathbf{n}_{i,f}$ is the outward unit normal for that edge.

9.3 Interior edge coupling (two neighboring triangles): baseline (two-point) flux

Consider an interior edge shared by triangles i and j. Let:

- $\mathbf{c}_i$ and $\mathbf{c}_j$ be the triangle centroids,
- $\mathbf{n}_{i,f}$ be the outward unit normal from cell i across the face f,
- L_f be the face (edge) length.

Define the centroid-to-centroid vector and its projection on the face normal:

$$\mathbf{d}_{ij} = \mathbf{c}_j - \mathbf{c}_i, \qquad d_n = \mathbf{d}_{ij} \cdot \mathbf{n}_{i,f}.$$

A basic two-point FV approximation of the face-normal derivative is

$$(\nabla\phi \cdot \mathbf{n}_{i,f})_f \approx \frac{\phi_j - \phi_i}{d_n}.$$

Then the flux contribution from this face to cell i is approximated by

$$\text{Flux}_{i,f} \approx L_f \, \frac{\phi_j - \phi_i}{d_n}.$$

9.4 Non-orthogonality and its treatment: Minimum Correction Method

On a structured orthogonal grid, the line connecting neighboring cell centers is aligned with the face normal. On a general triangular (unstructured) mesh this is **not guaranteed**:

$$\mathbf{d}_{ij} \not\parallel \mathbf{n}_{i,f}.$$

This is called **mesh non-orthogonality**. To treat non-orthogonality, the flux through each interior face must be computed using the **Minimum Correction Method** (as presented in the recitation of Lecture 2).

The minimum-correction approach is applied at the level of each face:

- decompose the face-normal flux into an *orthogonal (two-point)* contribution based on the normal projection d_n,
- and add the *minimum* additional correction required to account for the non-orthogonal component.

This method uses the same core grid data provided by the script (centroids, outward normals, face lengths, face midpoints, and neighbors) and augments the baseline two-point flux with a controlled non-orthogonal correction.

9.5 Boundary edges and boundary conditions (flux form)

Boundary edges are identified by:

`nbr12` $= 0$ (or nbr23/nbr31) in the triangle table, or `triR` $= 0$ in the unique edge table.

Neumann boundary condition

For a Neumann boundary condition

$$\frac{\partial\phi}{\partial n} = g \quad \text{on a boundary face } f,$$

the boundary flux is prescribed directly:

$$(\nabla\phi \cdot \mathbf{n})_f = g(\mathbf{m}_f), \qquad \Rightarrow \qquad \text{Flux}_{i,f} = g(\mathbf{m}_f)\, L_f,$$

where $\mathbf{m}_f = (m_x, m_y)$ is the face midpoint and L_f is the face length.

Dirichlet boundary condition (baseline face-normal flux)

For a Dirichlet boundary condition

$$\phi = \phi_b \quad \text{on a boundary face } f,$$

use the boundary value at the face midpoint $\phi_b(\mathbf{m}_f)$ and approximate the face-normal derivative using the distance from the cell centroid to the face along the outward normal:

$$d_{if} = (\mathbf{m}_f - \mathbf{c}_i) \cdot \mathbf{n}_{i,f}.$$

Then the boundary face-normal flux is approximated by

$$(\nabla\phi \cdot \mathbf{n})_f \approx \frac{\phi_b(\mathbf{m}_f) - \phi_i}{d_{if}}, \qquad \Rightarrow \qquad \text{Flux}_{i,f} \approx L_f \, \frac{\phi_b(\mathbf{m}_f) - \phi_i}{d_{if}}.$$

If a non-orthogonal correction is applied at boundaries, it should also follow the same **Minimum Correction Method** principle used for interior faces.

10 Two equivalent assembly strategies

The discrete Laplace operator can be assembled in two equivalent ways:

1. **Triangle-based loop:** for each triangle i, loop over its three local edges (12,23,31). Use `nbr12/nbr23/nbr31` to distinguish interior vs. boundary edges and use `n12/n23/n31` for flux direction.

2. **Edge-based loop:** loop over the unique edge list. For an interior edge with `triL`=i and `triR`=j, compute one face flux and add equal-and-opposite contributions to the two neighboring cells. For boundary edges (`triR`=0), apply the boundary condition.

Both methods rely on the same geometric building blocks: centroids, edge lengths, outward normals, edge midpoints, and neighbor connectivity.

11 Summary of quantities used in FV diffusion/Laplace discretization

From the mesh $(\mathbf{p}, \mathbf{t})$ and the derived tables, the following are used:

- Triangle centroids (`cx`, `cy`) for unknown placement.
- Face (edge) lengths (`len`).
- Outward unit normals: `n12x,n12y,n23x,n23y,n31x,n31y` (triangle table) and `nLx,nLy` (edge table).
- Neighbor connectivity: `nbr12,nbr23,nbr31` (triangle table) and `triL,triR` (edge table).
- Boundary detection: neighbor = 0 and `triR`=0.
- Edge midpoints (`mx`,`my`) for boundary data and face-based coefficients.
- Projected distances $d_n = (\mathbf{c}_j - \mathbf{c}_i) \cdot \mathbf{n}_{i,f}$ and $d_{if} = (\mathbf{m}_f - \mathbf{c}_i) \cdot \mathbf{n}_{i,f}$ for baseline face-normal flux approximations.
- Non-orthogonal flux treatment using the **Minimum Correction Method** (Lecture 2 recitation).