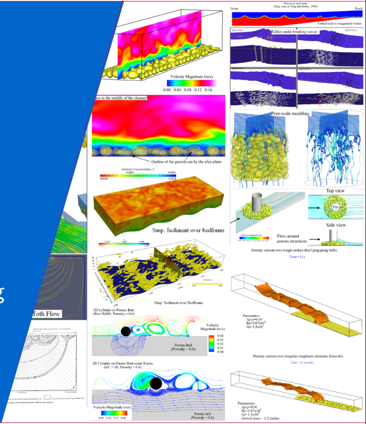
CE597D Special Topics, Fall, 2014

Chapter 4, Part 4: Unstructured Mesh

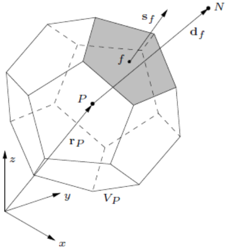
Xiaofeng Liu, Ph.D., P.E.
Assistant Professor
Department of Civil and Environmental Engineering
Pennsylvania State University
xliu@engr.psu.edu



Outline

Nomenclature

Computational Cell



- This is a convex polyhedral cell bounded by a set of convex polygons
- Point P is the computational point located at cell centroid $\vec{x}_P$. The definition of the centroid reads:

$$\int_{V_P} (\vec{x} - \vec{x}_P)\, dV = \vec{0} \tag{1}$$

Nomenclature

Computational Cell

- Cell volume is denoted by V_P
- For the cell, there is one neighbouring cell across each face. Neighbour cell and cell centre will be marked with N.
- The face centre f is defined in the equivalent manner, using the centroid rule:

$$\int_{S_f} (\vec{x} - \vec{x}_f)\, dS = \vec{0} \tag{2}$$

- Delta vector for the face f is defined as

$$\vec{d}_f = \overline{PN} \tag{3}$$

- Face area vector $\vec{s}_f$ is a surface normal vector whose magnitude is equal to the area of the face. The face is numerically never flat, so the face centroid and area are calculated from the integrals.

$$\vec{s}_f = \int_{S_f} \mathbf{n}\, dS \tag{4}$$

Nomenclature

Computational Cell

- The fact that the face centroid does not necessarily lay on the plane of the face is not worrying: we are dealing with surface-integrated quantities. However, we shall require the the cell centroid lays within the cell
- In practice, cell volume and face area calculated by decompositions into pyramids and triangles
- Types of faces in a mesh
  - **Internal face**, between two cells
  - **Boundary face**, adjacent to one cell only and pointing outwards of the computational domain
- When operating on a single cell, assume that all face area vectors $\vec{s}_f$ point outwards of cell P
- Discretisation is based on the integral form of the transport equation over each cell

$$\int_V \frac{\partial \phi}{\partial t}\, dV + \oint_S \phi\,(\mathbf{n}\cdot\mathbf{u})\, dS - \oint_S \gamma\,(\mathbf{n}\cdot\nabla\phi)\, dS = \int_V Q_v\, dV \qquad (5)$$

$$a_P\phi_P + \sum a_{nb}\phi_{nb} = S \qquad (6)$$

Spatial and Temporal Variation

Spatial Variation

- Postulating spatial variation of ϕ: second order discretisation in space

$$\phi(\vec{x}) = \phi_P + (\vec{x} - \vec{x}_P) \cdot (\nabla\phi)_P \tag{7}$$

This expression is given for each individual cell. Here, $\phi_P = \phi(\vec{x}_P)$.

Temporal Variation

- Postulating linear variation in time: second order in time

$$\phi(t + \Delta t) = \phi^t + \Delta t \left(\frac{\partial \phi}{\partial t}\right)^t \tag{8}$$

where $\phi^t = \phi(t)$

Polyhedral Mesh Support

- In FVM, we have specified the "shape function" without reference to the actual cell shape (tetrahedron, prism, brick, wedge). The variation is always linear. Doing polyhedral Finite Volume should be straightforward!

Volume and Surface Integrals

Evaluating Volume Integrals

$$\int_V \phi \, dV = \int_V \left[\phi_P + (\vec{x} - \vec{x}_P) \cdot (\nabla \phi)_P\right] \, dV \tag{9}$$

$$= \phi_P \int_V dV + (\nabla \phi)_P \cdot \int_V (\vec{x} - \vec{x}_P) dV \tag{10}$$

$$= \phi_P V_P \tag{11}$$

Evaluating Surface Integrals

- Surface integral splits into a sum over faces and evaluates in the same manner

$$\oint_S \mathbf{n} \, \phi \, dS = \sum_f \int_{S_f} \mathbf{n} \phi_f \, dS_f = \sum_f \int_{S_f} \mathbf{n}[\phi_f + (\vec{x} - \vec{x}_f) \cdot (\nabla \phi)_f] \, dS_f \tag{12}$$

$$= \sum_f \vec{s}_f \phi_f \tag{13}$$

- Assumption of linear variation of ϕ and selection of P and f in the centroid creates second-order discretisation

Gauss' Theorem

Gauss' Theorem in Finite Volume Discretisation

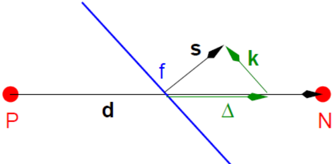
- Gauss' theorem is a tool we will use for handing the volume integrals of divergence and gradient operators
- Divergence form

$$\int_{V_P} \nabla \cdot \vec{a}\, dV = \oint_{\partial V_P} d\vec{s} \cdot \vec{a} \tag{14}$$

- Gradient form

$$\int_{V_P} \nabla \phi\, dV = \oint_{\partial V_P} d\vec{s}\, \phi \tag{15}$$

- Note how the face area vector operates from the same side as the gradient operator: fits with our definition of the gradient of a vector field
- In the rest of the analysis, we shall look at the problem face by face. A diagram of a face is given below for 2-D. Working with vectors will ensure no changes are required when we need to switch from 2-D to 3-D.



Operator Discretisation

Discretising Operators

- Typical numerical solution procedure
  - Split the space into cells and time into time steps
  - Assembled a discrete description of a continuous field variable
  - Postulated spatial and temporal variation of the solution for second-order discretisation
  - Generated expressions for evaluation of volume and surface integrals
- We shall now use this to assemble the discretisation of the differential operators
  1. Rate of change term
  2. Gradient operator
  3. Convection operator
  4. Diffusion operators
  5. Source and sink terms

Temporal Derivative

First Derivative in Time

- Time derivative captures the rate-of-change of ϕ. We only need to handle the volume integral.
- Defining time-step size Δt
- $t_{new} = t_{old} + \Delta t$, defining time levels ϕ^n and ϕ^o

$$\phi^o = \phi(t = t_{old}) \tag{16}$$

$$\phi^n = \phi(t = t_{new}) \tag{17}$$

- Temporal derivative, first and second order approximation

$$\frac{\partial \phi}{\partial t} = \frac{\phi^n - \phi^o}{\Delta t} \tag{18}$$

$$\frac{\partial \phi}{\partial t} = \frac{\frac{3}{2}\phi^n - 2\phi^o + \frac{1}{2}\phi^{oo}}{\Delta t} \tag{19}$$

Matrix Coefficients

First Derivative in Time

- Thus, with the volume integral:

$$\int_V \frac{\partial \phi}{\partial t}\, dV = \frac{\phi^n - \phi^o}{\Delta t} V_P \tag{20}$$

$$\int_V \frac{\partial \phi}{\partial t}\, dV = \frac{\frac{3}{2}\phi^n - 2\phi^o + \frac{1}{2}\phi^{oo}}{\Delta t} V_P \tag{21}$$

Temporal Derivative

- Calculus: given ϕ^n, ϕ^o and Δt create a field of the time derivative of ϕ
- Method: matrix representation. Since $\frac{\partial \phi}{\partial t}$ in cell P depends on ϕ_P only, the matrix will only have a diagonal contribution and a source. For the case of first-order Euler implicit scheme,
  - Diagonal coefficient: $a_P = \frac{V_P}{\Delta t}$
  - Source contribution: $r_P = \frac{V_P \phi^o}{\Delta t}$

Gradient Calculation

Evaluating the Gradient

- How to evaluate a gradient of a given field: Gauss Theorem

$$\int_{V_P} \nabla\phi \, dV = \oint_{\partial V_P} d\vec{s}\, \phi \tag{22}$$

- Discretised form splits into a sum of face integrals

$$\oint_S \mathbf{n}\phi \, dS = \sum_f \vec{s}_f \phi_f \tag{23}$$

- It still remains to evaluate the face value of ϕ. Consistently with second-order discretisation, we shall assume linear variation between P and N

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{24}$$

where $f_x = \overline{fN}/\overline{PN}$

Convection Operator

Convection Operator

- Convection term captures the transport by convective velocity
- Convection operator splits into a sum of face integrals (integral and differential form)

$$\int_V \nabla \cdot (\phi \mathbf{u})\, dV = \oint_S \phi(\mathbf{n} \cdot \mathbf{u}) dS \tag{25}$$

- Integration follows the same path as before

$$\oint_S \phi(\mathbf{n} \cdot \mathbf{u}) dS = \sum_f \phi_f(\vec{s}_f \cdot \mathbf{u}_f) = \sum_f \phi_f F \tag{26}$$

where ϕ_f is the face value of ϕ and

$$F = \vec{s}_f \cdot \mathbf{u}_f \tag{27}$$

is the **face flux**: measure of the flow through the face

- In order to close the system, we need a way of evaluating ϕ_f from the cell values ϕ_P and ϕ_N: **face interpolation**

Face Interpolation

Face Interpolation Scheme for Convection

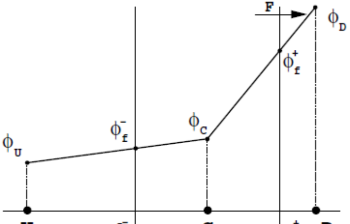
- Simplest face interpolation: **central differencing**. Second-order accurate, but causes oscillations

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{28}$$

- Upwind differencing: taking into account the **transportive property** of the term: information comes from upstream. No oscillations, but smears the solution

$$\phi_f = max(F, 0)\,\phi_P + min(F, 0)\,\phi_N \tag{29}$$

- There exists a large number of schemes, trying to achieve good accuracy without causing oscillations: *e.g.* TVD, and NVD families: $\phi_f = f(\phi_P, \phi_N, F, \ldots)$



PENNSTATE

Matrix Coefficients

Convection Discretisation

- In the convection term, ϕ_f depends on the values of ϕ in two computational points: P and N.
- Therefore, the solution in P will depend on the solution in N and vice versa, which means we've got an **off-diagonal coefficient** in the matrix. In the case of central differencing on a uniform mesh, a contribution for a face f is
  - Diagonal coefficient: $a_P = \frac{1}{2}F$
  - Off-diagonal coefficient: $a_N = \frac{1}{2}F$
  - Source contribution: in our case, nothing. However, some other schemes may have additional (gradient-based) correction terms
  - Note that, in general the P-to-N coefficient will be different from the N-to-P coefficient: the matrix is asymmetric
- Upwind differencing
  - Diagonal coefficient: $a_P = max(F, 0)$
  - Off-diagonal coefficient: $a_N = min(F, 0)$

Diffusion Operator

Diffusion Operator

- Diffusion term captures the gradient transport
- Integration same as before

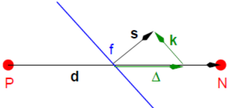
$$\oint_S \gamma(\mathbf{n} \cdot \nabla\phi) dS = \sum_f \int_{S_f} \gamma(\mathbf{n} \cdot \nabla\phi)\, dS \tag{30}$$

$$= \sum_f \gamma_f\, \vec{s}_f \cdot (\nabla\phi)_f \tag{31}$$

- γ_f evaluated from cell values using central differencing
- Evaluation of the face-normal gradient. If $\vec{s}$ and $\vec{d}_f = \overline{PN}$ are aligned, use difference across the face

$$\vec{s}_f \cdot (\nabla\phi)_f = |\vec{s}_f| \frac{\phi_N - \phi_P}{|\vec{d}_f|} \tag{32}$$

- This is the component of the gradient in the direction of the $\vec{d}_f$ vector
- For non-orthogonal meshes, a correction term may be necessary

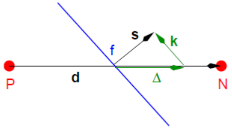


PENNSTATE

Matrix Coefficients

Diffusion Operator

- For an orthogonal mesh, a contribution for a face f is
  - Diagonal value: $a_P = -\gamma_f \frac{|\vec{s}_f|}{|\vec{d}_f|}$
  - Off-diagonal value: $a_N = \gamma_f \frac{|\vec{s}_f|}{|\vec{d}_f|}$
  - Source contribution: for orthogonal meshes, nothing. Non-orthogonal correction will produce a source
- The P-to-N and N-to-P coefficients are identical: **symmetric matrix**. This is an important characteristic of the diffusion operator
- For **non-orthogonal meshes**, a correction is added to compensate for the angle between the face area and $\overline{PN}$ vectors



Source and Sink Terms

Source and Sinks

- Source and sink terms are local in nature

$$\int_V q_v \, dV = q_v V_P \tag{33}$$

- In general, q_v may be a function of space and time, the solution itself, other variables and can be quite complex. In complex physics cases, the source term can carry the main interaction in the system. Example: complex chemistry mechanisms. We shall for the moment consider only a simple case.
- Typically, linearisation with respect to ϕ is performed to promote stability and boundedness

$$q_v(\phi) = q_u + q_d \phi \tag{34}$$

where $q_d = \frac{\partial \mathbf{q_v}(\phi)}{\partial \phi}$ and for cases where $q_d < 0$ (sink), treated separately

Matrix Coefficients

- Source and sink terms do not depend on the neighborhood
  - Diagonal value created for $q_d < 0$: "boosting diagonal dominance"
  - Explicit source contribution: q_u

PENNSTATE

Numerical Boundary Conditions

Implementation of Boundary Conditions

- Boundary conditions will contribute the the discretisation through the prescribed boundary behaviour
- Boundary condition is specified for the whole equation
- . . . but we will study them term by term to make the problem simpler

Dirichlet Condition

Dirichlet Condition: Fixed Boundary Value

- Boundary condition specifies $\phi_f = \phi_b$
- Convection term: fixed contribution $F\,\phi_b$. Source contribution only (only true in OpenFOAM since it enforces the B.C. strictly).
- Diffusion term: need to evaluate the near-boundary gradient

$$\mathbf{n} \cdot (\nabla\phi)_b = \frac{\phi_b - \phi_P}{|\vec{d}_b|} \tag{35}$$

  This produces a source and a diagonal contribution
- What about source, sink, rate of change? No effect from the B.C. since they only depend on the cell center value.

Neumann Condition

Neumann and Gradient Condition

- Boundary condition specifies the near-wall gradient $\mathbf{n} \cdot (\nabla\phi)_b = g_b$
- Convection term: evaluate the boundary value of ϕ from the internal value and the known gradient

$$\phi_b = \phi_P + \vec{d}_b \cdot (\nabla\phi)_b = \phi_P + |\vec{d}_b| g_b \tag{36}$$

  Use the evaluated boundary value as the face value. This creates a source and a diagonal contribution
- Diffusion term: boundary-normal g_b gradient can be used directly. Source contribution only

Mixed Condition

- Combination of the above
- Very easy: α times Dirichlet plus $(1 - \alpha)$ times Neumann

Geometric and Coupled Conditions

- Symmetry plane condition is enforced using the mirror-image of internal solution
- Cyclic and periodic boundary conditions couple near-boundary cells to cells on another boundary

Questions?