2.29 Numerical Fluid Mechanics
Fall 2011 – Lecture 23

REVIEW Lectures 22:

- Complex Geometries
- Grid Generation
  - Basic concepts and structured grids
    - Stretched grids
    - Algebraic methods
    - General coordinate transformation
    - Differential equation methods
    - Conformal mapping methods
  - Unstructured grid generation
    - Delaunay Triangulation
    - Advancing Front method
- Finite Volume on Complex geometries
  - Computation of convective fluxes
  - Computation of diffusive fluxes
  - Comments on 3D

TODAY (Lecture 23): Grid Generation (end), FV on Complex Geometries and Solution to the Navier-Stokes Equations

- Grid Generation
  - Basic concepts and structured grids
    - Algebraic methods (stretched grids), General coordinate transformation, Differential equation methods, Conformal mapping methods
  - Unstructured grid generation
    - Delaunay Triangulation
    - Advancing Front method
- Finite Volume on Complex geometries
  - Computation of convective fluxes
  - Computation of diffusive fluxes
  - Comments on 3D
- Solution of the Navier-Stokes Equations
  - Discretization of the convective and viscous terms
  - Discretization of the pressure term
  - Conservation principles

References and Reading Assignments

- Chapter 8 on “Complex Geometries” and Chapter 7 on “Incompressible Navier-Stokes equations” of “J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. Springer, NY, 3rd edition, 2002”
- Chapter 9 on “Grid Generation” and Chapter 11 on “Incompressible Navier-Stokes Equations” of T. Cebeci, J. P. Shao, F. Kafyeke and E. Laurendeau, *Computational Fluid Dynamics for Engineers*. Springer, 2005.
- Chapter 13 on “Grid Generation” and Chapter 17 on “Incompressible Viscous Flows” of Fletcher, *Computational Techniques for Fluid Dynamics*. Springer, 2003.
- Ref on Grid Generation only:
  - Thompson, J.F., Warsi Z.U.A. and C.W. Mastin, “Numerical Grid Generation, Foundations and Applications”, North Holland, 1985

Grid Generation: Unstructured Grids

- Generating unstructured grid is complicated but now relatively automated in “classic” cases
- Involves succession of smoothing techniques that attempt to align elements with boundaries of physical domain
- Decompose domain into blocks to de-couple the problems
- Need to define point positions and connections
- Most popular algorithms:
  - Delaunay Triangulation Method
  - Advancing Front Method
- Two schools of thought: structured vs. unstructured, what is best for CFD?

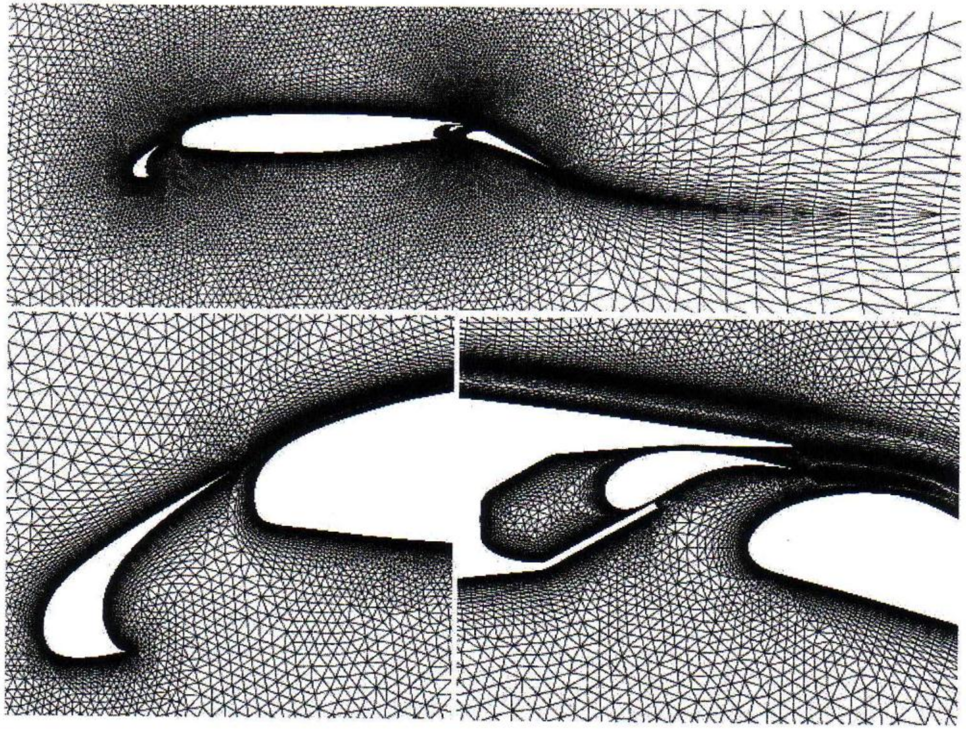


Fig. 9.16. 2D Unstructured grid for Navier–Stokes computations of a multi-element airfoil generated with the hybrid advancing front Delaunay method of Mavriplis [6].

- Structured grids: simpler grid and straightforward treatment of algebraic system, but mesh generation constraints on complex geometries
- Unstructured grids: generated faster on complex domains, easier mesh refinements, but data storage and solution of algebraic system more complex

Grid Generation: Unstructured Grids

- **Delaunay Triangulation (DT)**
  - Use a simple criterion to connect points to form conforming, non-intersecting elements
  - Maximizes minimum angle in each triangle
  - Not unique
  - Task of point generation is done independently of connection generation
- Based on Dirichlet's domain decomposition into a set of packed convex regions:
  - For a given set of points P, the space is subdivided into regions in such a way that each region is the space closer to P than to any other point = Dirichlet tessellation

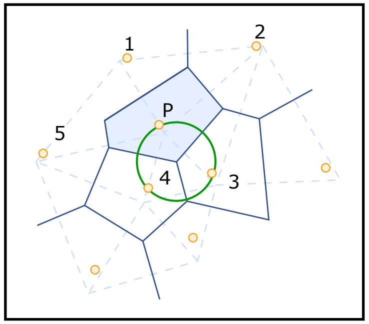


Image by MIT OpenCourseWare.

Note: at the end, points P are at summits of triangles

- This geometrical construction is known as the **Dirichlet tessellation**
- The tessellation of a closed domain results in a set of non-overlapping convex regions called **Voronoi regions/polygons**
- The sides of the polygon around P is made of segments bisectors of lines joining P to its neighbors: if all pair of points with a common segment are joined by straight lines, the result is a Delaunay Triangulation
- Each vortex of a Voronoi diagram is the circumcenter of the triangle formed by the three points of a Delaunay triangle
- Criterion: the circumcircle can not contain any other point than these three points

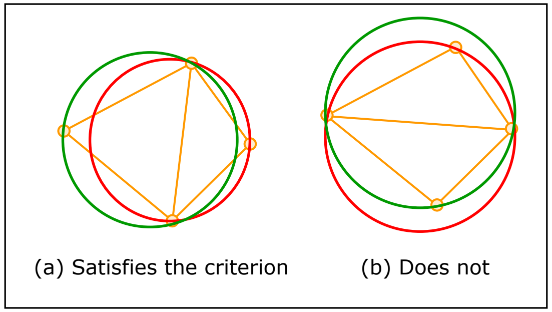


Image by MIT OpenCourseWare.

Grid Generation: Unstructured Grids

- **Advancing Front Method**
  - In this method, the tetrahedras are built progressively, inward from the boundary
  - An active front is maintained where new tetrahedra are formed
  - For each triangle on the edge of the front, an ideal location for a new third node is computed
  - Requires intersection checks to ensure triangles don't overlap

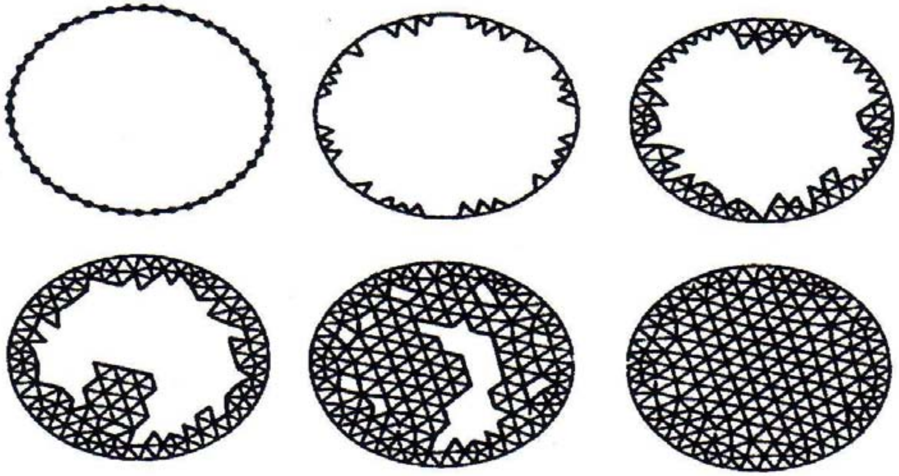


Fig. 9.20. Advancing Front technique for unstructured grid generation.

- In 3D, the Delaunay Triangulation is preferred (faster)

Finite Volumes on Complex geometries

- FD method (classic):
  - – Use structured-grid transformation (either algebraic-transfinite, general, differential or conformal mapping)
  - – Solve transformed equations on simple orthogonal computational domain
- FV method:
  - – Starts from conservation eqns. in integral form on CV

$$\frac{d}{dt}\int_{CV}\rho\phi dV+\underbrace{\int_{CS}\rho\phi\,(\vec{v}.\vec{n})dA}_{\text{Advective (convective) fluxes}}=\underbrace{-\int_{CS}\vec{q}_{\phi}.\vec{n}\,dA}_{\text{Other transports (diffusion, etc)}}+\underbrace{\sum\int_{CV}s_{\phi}\,dV}_{\substack{\text{Sum of sources and}\\ \text{sinks terms (reactions, etc)}}}$$

  - – We have seen principles of FV discretization
    - Convective/diffusive fluxes, from 1st - 2nd order to higher order discretizations
    - These principles are independent of grid specifics, but,
    - Several new features arise with non-orthogonal or arbitrary unstructured grids

Expressing fluxes at the surface based on cell-averaged (nodal) values: Summary of Two Approaches and Boundary Conditions

- Set-up of surface/volume integrals: 2 approaches (do things in opposite order)

1. (i) Evaluate integrals using classic rules (symbolic evaluation); (ii) Then, to obtain the unknown symbolic values, interpolate based on cell-averaged (nodal) values

$$\left.\begin{aligned}&(i)\ F_e=\int_{S_e} f_\phi\, dA \quad \Rightarrow \quad F_e=G(\phi_e)\\ &(ii)\ \phi_e=H\ (\bar{\phi}_P\,'s)\equiv H\ (\phi_P\,'s)\end{aligned}\right\}\Rightarrow F_e=F\ (\bar{\phi}_P\,'s)$$

Similar for other integrals:

$$(S_\phi=\int_V s_\phi\, dV\ ,\ \bar{\Phi}=\frac{1}{V}\int_V \rho\phi dV\, , etc)$$

2. (i) Select shape of solution within CV (piecewise approximation); (ii) impose volume constraints to express coefficients in terms of nodal values; and (iii) then integrate. (this approach was used in the examples).

$$\left.\begin{aligned}&\left.\begin{aligned}&(i)\ \phi_{a_i}(x)\equiv J_{\ a_i}(x)\\ &(ii)\ \int_{V_P}\phi_{a_i}(x)\equiv\bar{\phi}_P\end{aligned}\right\}\Rightarrow \phi_{a_i}(x)\equiv\phi_{\bar{\phi}_P}(x)\\ &(iii)\ F_e=\int_{S_e} f_{\phi_{\bar{\phi}_P}}\ dA\end{aligned}\right\}\Rightarrow F_e=F\ (\bar{\phi}_P\,'s)$$

Similar for higher dimensions:

$$\phi(x,y)\equiv J_{\ a_i}(x,y);\quad etc$$

$$\phi_{a_i}(x_P,y_P)\equiv\phi_P;\quad etc$$

- Boundary conditions:

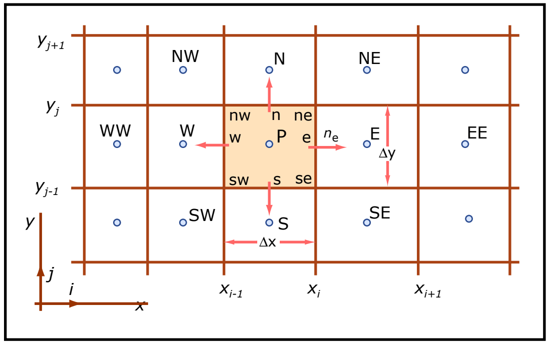
(From lecture 19)

  – Directly imposed for convective fluxes

  – One-side differences for diffusive fluxes

(summary from Lecture 18)

Approx. of Surface/Volume Integrals: Classic symbolic formulas



Notation used for a Cartesian 2D and 3D grid.
Image by MIT OpenCourseWare.

- Surface Integrals $F_e = \int_{S_e} f_\phi \, dA$
  - 2D problems (1D surface integrals)
    - Midpoint rule (2nd order): $F_e = \int_{S_e} f_\phi \, dA = \overline{f}_e S_e = f_e S_e + O(\Delta y^2) \approx f_e S_e$
    - Trapezoid rule (2nd order): $F_e = \int_{S_e} f_\phi \, dA \approx S_e \dfrac{(f_{ne} + f_{se})}{2} + O(\Delta y^2)$
    - Simpson's rule (4th order): $F_e = \int_{S_e} f_\phi \, dA \approx S_e \dfrac{(f_{ne} + 4 f_e + f_{se})}{6} + O(\Delta y^4)$
  - 3D problems (2D surface integrals)
    - Midpoint rule (2nd order): $F_e = \int_{S_e} f_\phi \, dA \approx S_e f_e \quad + O(\Delta y^2, \Delta z^2)$
    - Higher order more complicated to implement in 3D
- Volume Integrals: $S_\phi = \int_V s_\phi \, dV \,, \quad \overline{\Phi} = \dfrac{1}{V} \int_V \rho \phi dV$
  - 2D/3D problems, Midpoint rule (2nd order): $S_P = \int_V s_\phi \, dV = \overline{s}_P \, V \approx s_P \, V$
  - 2D, bi-quadratic (4th order, Cartesian): $S_P = \dfrac{\Delta x \, \Delta y}{36} \left[16 s_P + 4 s_s + 4 s_n + 4 s_w + 4 s_e + s_{se} + s_{sw} + s_{ne} + s_{nw} \right]$

FV: Approximation of **convective fluxes**

$$\underbrace{\int_{CS} \rho\phi\,(\vec{v}.\vec{n})dS}_{\text{Advective (convective) fluxes}}$$

- For complex geometries, one often uses the midpoint rule for the approximation of surface and volume integrals
- Consider first the mass flux: $\phi=1$: $f_{\phi=1} = \rho\,\vec{v}.\vec{n}$
  - Again, we consider one face only: east side of a 2D CV (same approach applies to other faces and to any CV shapes).
  - Mid-point rule for mass flux: $\underline{\dot{m}_e} = \int_{S_e} f_{\phi=1}\,dS = \bar{f}_e S_e = f_e S_e + O(\Delta^2) \approx \underline{(\rho\,\vec{v}.\vec{n})_e S_e}$

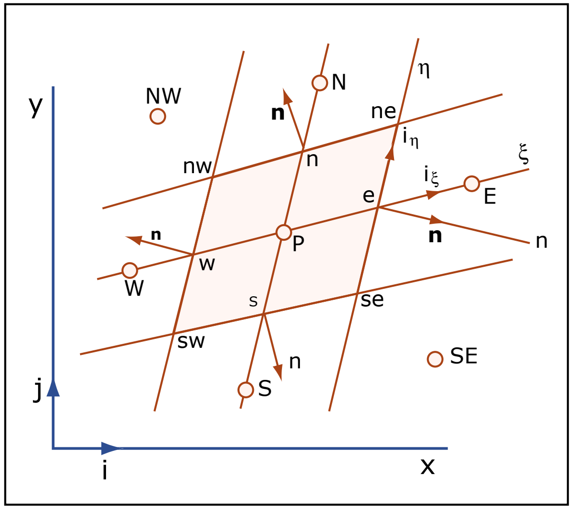


Image by MIT OpenCourseWare.

- The unit normal vector to face "*e*" and its surface S_e are defined as: $\mathbf{n}_e S_e = S_e^x\;\mathbf{i} + S_e^y\;\mathbf{j} = (y_{ne} - y_{se})\,\mathbf{i} - (x_{ne} - x_{se})\,\mathbf{j}$

  where $S_e = \sqrt{(S_e^x)^2 + (S_e^y)^2}$
- Hence, mass flux is:

$$\boxed{\dot{m}_e \approx (\rho\,\vec{v}.\vec{n})_e S_e = \rho_e\,\vec{v}_e.(S_e^x\;\mathbf{i} + S_e^y\;\mathbf{j}) = \rho_e\,(S_e^x\;u_e + S_e^y\;v_e)}$$

FV: Approximation of convective fluxes, Cont'd Mass Flux

- The mass flux for the mid-point rule: $\boxed{\dot{m}_e = \rho_e \,(S_e^x \, u_e + S_e^y \, v_e)}$

- What's the difference between the Cartesian and non-orthogonal grid cases?
  - In non-orthogonal case, normal to surface has components in all directions
  - All velocity components thus contribute to the flux (each component is multiplied by the projection of S_e onto the corresponding axis)

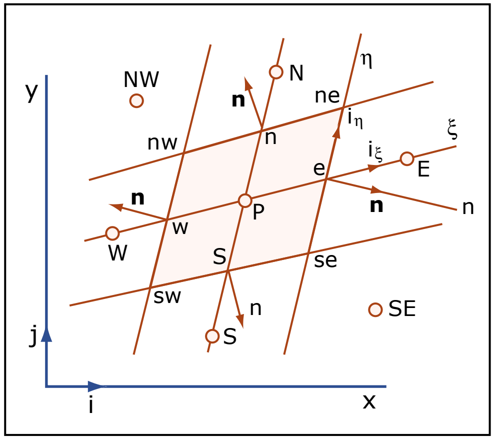


Image by MIT OpenCourseWare.

FV: Approximation of convective fluxes, Cont'd

- Mass flux for mid-point rule: $\boxed{\dot{m}_e = \rho_e \, (S_e^x \; u_e + S_e^y \; v_e)}$
- Convective flux for any transported ϕ
  - Is usually computed after the mass flux. Using the mid-point rule:

$$\boxed{F_e = \int_{S_e} \rho\phi \, (\vec{v}.\vec{n}) \, dS \approx f_e S_e = (\rho\phi\vec{v}.\vec{n})_e S_e = \phi_e \dot{m}_e}$$

where ϕ_e = value ϕ at center of cell face

  - How to obtain ϕ_e?, use either:
    - A linear interpolation between two nodes on either side of face (also 2nd order) ⇒ becomes trapezoidal rule
    - Fit ϕ to a polynomial in the vicinity of the face (piecewise shape function)
  - Considerations for unstructured grid:
    - Best compromise among accuracy, generality and simplicity is usually: Linear interpolation and mid-point rule
    - Indeed: facilitates use of local grid refinement, which can be used to achieve higher accuracy at lower cost than higher-order schemes. However, higher-order FE or compact FD are now being used/developed

FV: Approximation of **diffusive fluxes**

$$\underbrace{\int_{CS} k\nabla\phi.\vec{n}\, dS}_{\text{Diffusive Fluxes}}$$

- For complex geometries, we can still use the midpoint rule
- Mid-point rule gives: $F_e^d = \int_{S_e} k\nabla\phi.\vec{n}\, dS = \overline{f}_e S_e = f_e S_e + O(\Delta^2) \approx (k\nabla\phi.\vec{n})_e S_e$
- Here, gradient can be expressed in terms of global Cartesian coordinates (x, y) or local orthogonal coordinates (n, t)

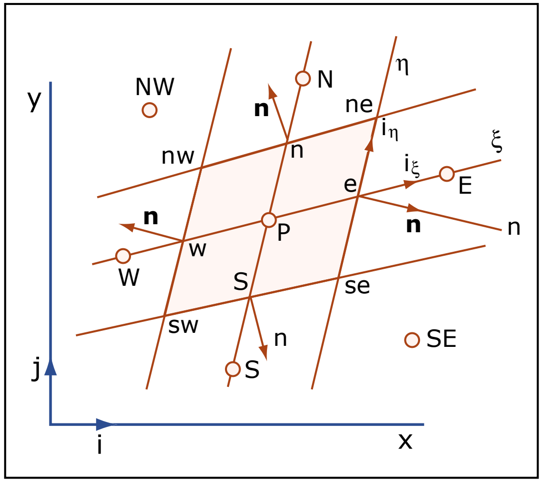


Image by MIT OpenCourseWare.

– In 2D: $\nabla\phi = \frac{\partial\phi}{\partial x}\mathbf{i} + \frac{\partial\phi}{\partial y}\mathbf{j} = \frac{\partial\phi}{\partial n}\mathbf{n} + \frac{\partial\phi}{\partial t}\mathbf{t}$

– There are many ways to approximate the derivative normal to the cell face or the gradient vector at the cell center

– As always, two main approaches:

- Approximate surface integral, then interpolate
- Specify shape function, constraints, then integrate

FV: Approximation of **diffusive fluxes, Cont'd**

1) If shape function $\phi(x, y)$ is used, with mid-point rule, this gives:

$$\boxed{F_e^d \approx (k\nabla\phi.\vec{n})_e S_e = k_e \left(S_e^x \left.\frac{\partial\phi}{\partial x}\right|_e + S_e^y \left.\frac{\partial\phi}{\partial y}\right|_e \right) = k_e \sum_i S_e^{x_i} \left.\frac{\partial\phi}{\partial x_i}\right|_e}$$

- Can be evaluated and relatively easy to implement explicitly
- Implicitly can be harder for high-order shape fct $\phi(x, y)$ (more cell involved)

2) Another way is to compute derivatives at CV centers first, then interpolate to cell faces (2 steps as for computing ϕ_e from ϕ_P)

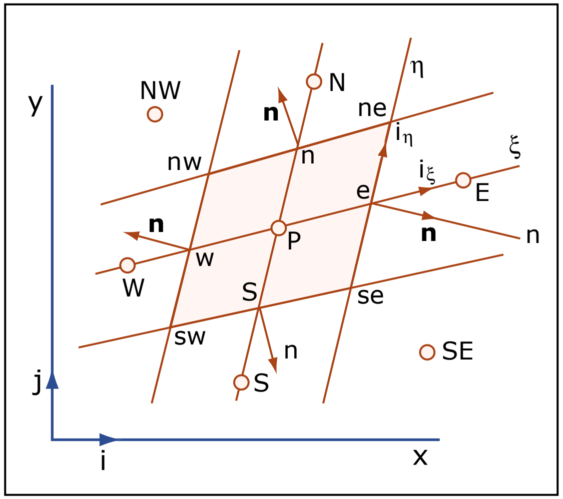


Image by MIT OpenCourseWare.

i) One can use averages + Gauss Theorem locally

- Derivative at center ≈ average derivative over cell

$$\left.\frac{\partial\phi}{\partial x}\right|_P \approx \int_{CV} \frac{\partial\phi}{\partial x} dV \Big/ dV = \left.\overline{\frac{\partial\phi}{\partial x}}\right|_P$$

- Gauss theorem for $\partial\phi/\partial x$ (similar for y derivative):

$$\int_{CV} \frac{\partial\phi}{\partial x} dV = \int_{CS} \phi\, \mathbf{i.n}\, dS \approx \sum_{4\,\text{faces}\,c} \phi_c S_c^x$$

FV: Approximation of **diffusive fluxes, Cont'd**

- Hence, the gradient at the CV center with respect to x_i is obtained by summing the products of each ϕ_c with the projection of its cell surface onto a plane normal to x_i, and then dividing the sum by the CV volume

$$\boxed{\left.\frac{\partial\phi}{\partial x_i}\right|_P \approx \left.\overline{\frac{\partial\phi}{\partial x_i}}\right|_P = \int_{CV}\frac{\partial\phi}{\partial x_i}\,dV \Big/ dV = \sum_{4\,c\text{ faces}} \phi_c S_c^{x_i} \Big/ dV}$$

- For ϕ_c we can use the approximation for the convective fluxes
- We can then interpolate to obtain the gradient at the centers of cell faces
- For Cartesian grids and linear interpolation, one retrieves centered FD

ii) Cell-center gradients can also be approximated to 2nd order assuming a linear variation of ϕ locally:

$$\phi_E - \phi_P \approx \nabla\phi\big|_P \, . \, (\mathbf{r}_E - \mathbf{r}_P)$$

- There are as many such equations as there are neighbors to the cell centered at P ⇒ need for least-square inversions (only n derivatives in nD)
- Issues with this approximation are oscillatory solutions and large computational stencils for implicit schemes ⇒ use deferred-correction approach

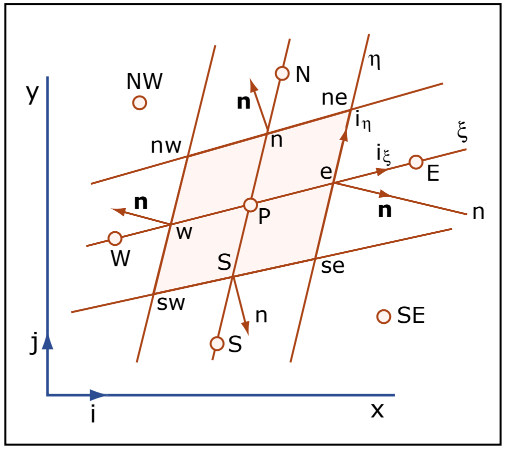


Image by MIT OpenCourseWare.

FV: Approximation of diffusive fluxes, Cont'd

iii) Deferred-Correction Approach:

- Idea behind deferred-correction is to identify possible options and combine them to reduce costs and eliminate un-desired behavior. Some options:
- If we work in local coordinates (n, t): $F_e^d \approx (k\nabla\phi.\vec{n})_e S_e = k_e S_e \left.\frac{\partial\phi}{\partial n}\right|_e$
- If grid was close to orthogonal Cartesian, using CDS: $\left.\frac{\partial\phi}{\partial n}\right|_e \approx \left.\frac{\partial\phi}{\partial\xi}\right|_e \approx \frac{\phi_E - \phi_P}{|\mathbf{r}_E - \mathbf{r}_P|}$ (1)
- If interpolate the gradient at the cell center: $\left.\frac{\partial\phi}{\partial n}\right|_e \approx \left.\frac{\partial\phi}{\partial\xi}\right|_e \approx \frac{1}{2}\frac{\phi_E - \phi_W}{|\mathbf{r}_E - \mathbf{r}_W|} + \frac{1}{2}\frac{\phi_{EE} - \phi_P}{|\mathbf{r}_{EE} - \mathbf{r}_P|}$ (2)

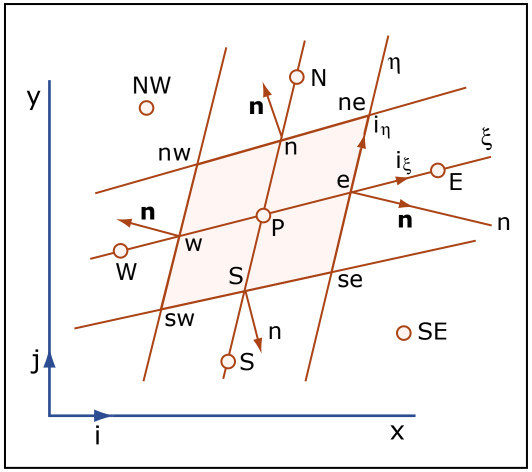


Image by MIT OpenCourseWare.

- Oscillatory solutions do not contribute to this third higher-order choice, but gradients at cell faces would still be large => oscillations do occur:

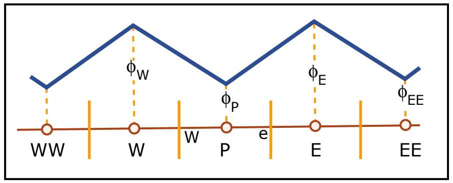


Image by MIT OpenCourseWare.

- The obvious solution: $\left.\frac{\partial\phi}{\partial n}\right|_e \approx \left[(1)^{\text{implicit}}\right]^{r+1} + \left[(2)^{\text{explicit}} - (1)^{\text{implicit}}\right]^{r}$

  will oscillate ⇒ Need to find other options/solution

FV: Approximation of **diffusive fluxes, Cont'd**

iii) Deferred-Correction Approach, Cont'd (Muzaferija, 1994):

- If line connecting nodes P and E is nearly orthogonal to cell face, derivative w.r.t n can be approximated with derivative w.r.t ξ (as before):
  - $\Rightarrow$ approximation close to 2nd order $F_e^d \approx k_e S_e \left.\frac{\partial \phi}{\partial n}\right|_e \approx k_e S_e \left.\frac{\partial \phi}{\partial \xi}\right|_e \approx k_e S_e \frac{\phi_E - \phi_P}{|\mathbf{r}_E - \mathbf{r}_P|}$
- If grid is not orthogonal, the deferred correction term should contain the difference between the gradients in the n and ξ directions $\Rightarrow$

$$F_e^d \approx k_e S_e \left.\frac{\partial \phi}{\partial \xi}\right|_e + k_e S_e \left[\left.\frac{\partial \phi}{\partial n}\right|_e - \left.\frac{\partial \phi}{\partial \xi}\right|_e \right]^{\text{old}}$$

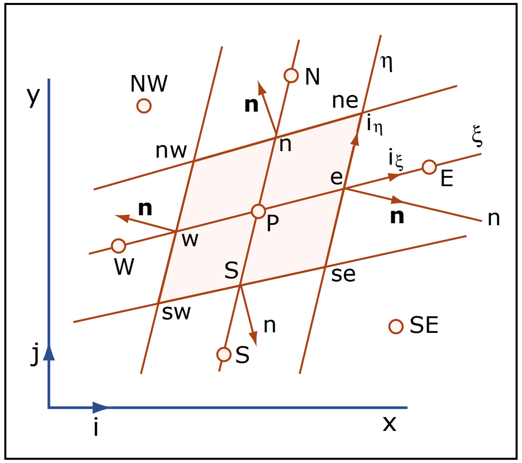


Image by MIT OpenCourseWare.

- where the first term is computed as: $k_e S_e \left.\frac{\partial \phi}{\partial \xi}\right|_e = k_e S_e \frac{\phi_E - \phi_P}{|\mathbf{r}_E - \mathbf{r}_P|}$
- Bracket term interpolated from cell center gradients (themselves obtained from Gauss theorem)

$$\left.\frac{\partial \phi}{\partial n}\right|_e \approx \left.\overline{\frac{\partial \phi}{\partial n}}\right|_e = \overline{\nabla \phi}\Big|_e . \mathbf{n} \quad \text{and} \quad \left.\frac{\partial \phi}{\partial \xi}\right|_e \approx \left.\overline{\frac{\partial \phi}{\partial \xi}}\right|_e = \overline{\nabla \phi}\Big|_e . \mathbf{i}_\xi$$

- Hence: $\boxed{F_e^d \approx k_e S_e \frac{\phi_E - \phi_P}{|\mathbf{r}_E - \mathbf{r}_P|} + k_e S_e \left[\overline{\nabla \phi}\Big|_e \right]^{\text{old}} \left(\mathbf{n} - \mathbf{i}_\xi \right)}$

FV: Approximation of **diffusive fluxes, Cont'd**

iii) Deferred-Correction Approach, Cont'd (Muzaferija, 1994):

– In the formula:

$$\boxed{F_e^d \approx k_e S_e \frac{\phi_E - \phi_P}{|\mathbf{r}_E - \mathbf{r}_P|} + k_e S_e \left[\overline{\nabla\phi}\Big|_e\right]^{\text{old}} \left(\mathbf{n} - \mathbf{i}_\xi\right)}$$

- The deferred correction term is (close to) zero when grid (close to) orthogonal, i.e. n and ξ directions are the same (close to each other).
- It makes the computation of derivatives simple (amounts to sums of neighbor values), recall that:

$$\left[\overline{\nabla\phi}\Big|_{\underline{e}}\right]^{\text{old}} \text{ interpolated from} \left[\overline{\nabla\phi}\Big|_{\underline{P}}\right]^{\text{old}},$$

$$\text{the latter given by e.g. } \overline{\left.\frac{\partial\phi}{\partial x_i}\right|_P} = \sum_{4\,c\ \text{faces}} \phi_c S_c^{x_i} \Big/ dV$$

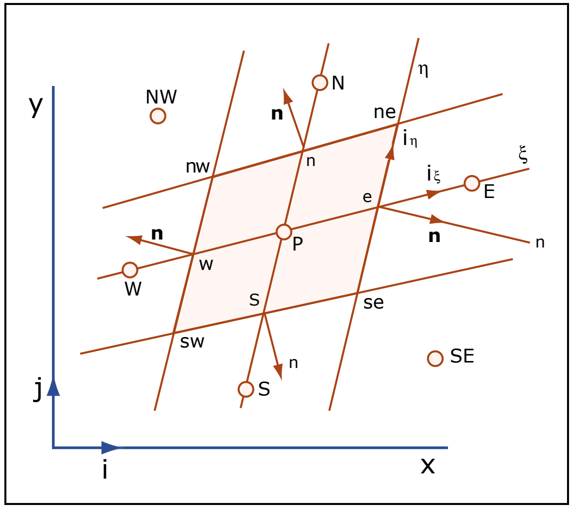
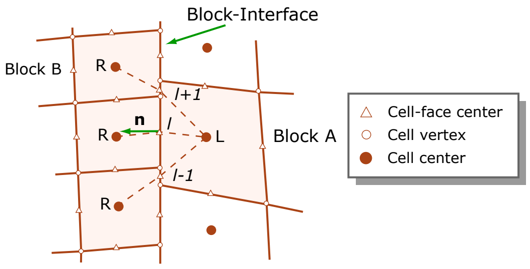


Image by MIT OpenCourseWare.

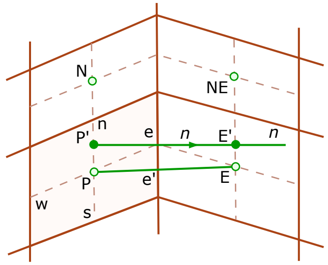
- Prevents oscillations since based on sums of ϕ_c, with positive coefficients
- We remained in Cartesian coordinates (no need to transform coordinates, we just need to know the normals & surfaces), which is handy for complex turbulent models

Some comments on FV on complex geometries

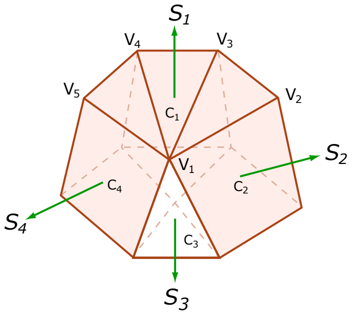
- Line P-E does not always pass through the cell center
  - ⇒ need some updates in that case
  - otherwise, scheme is of lower order (e.g. approximation is not second order anymore)
- Schemes can be extended to 3D grids but some updates can also be needed
  - For example, cell faces are not always planar in 3D
- Block-structured grids and nested grids also need special treatment
  - For example, matching at boundaries (usually interpolation and averaging)



The interface between two blocks with non-matching grids.



Collocated arrangement of velocity components and pressure on FD and FV grids.

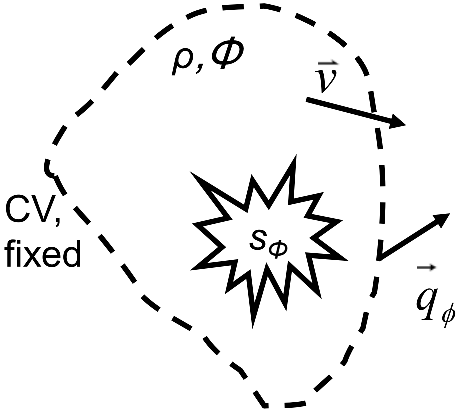


Cell volume and surface vectors for arbitrary control volumes.

Image by MIT OpenCourseWare.

Integral Conservation Law for a scalar ϕ

$$\left\{\frac{d}{dt}\int_{CM}\rho\phi dV=\right\}\boxed{\frac{d}{dt}\int_{CV_{\text{fixed}}}\rho\phi dV+\underbrace{\int_{CS}\rho\phi\,(\vec{v}.\vec{n})dA}_{\substack{\text{Advective fluxes}\\ \text{("convective" fluxes)}}}=\underbrace{-\int_{CS}\vec{q}_{\phi}.\vec{n}\,dA}_{\text{Other transports (diffusion, etc)}}+\underbrace{\sum\int_{CV_{\text{fixed}}}s_{\phi}\,dV}_{\substack{\text{Sum of sources and}\\ \text{sinks terms (reactions, etc)}}}}$$



Applying the Gauss Theorem, for any arbitrary CV gives:

$$\frac{\partial\rho\phi}{\partial t}+\nabla.(\rho\phi\vec{v})=-\nabla.\vec{q}_{\phi}+s_{\phi}$$

For a common diffusive flux model (Fick’s law, Fourier’s law):

$$\vec{q}_{\phi}=-k\nabla\phi$$

Conservative form of the PDE $\longrightarrow$

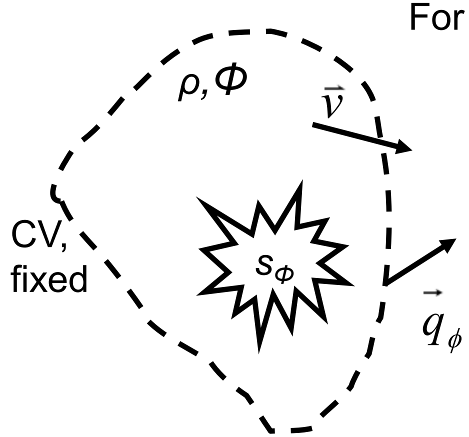
$$\frac{\partial\rho\phi}{\partial t}+\nabla.\,(\rho\phi\vec{v})=\nabla.\,(k\nabla\phi)+s_{\phi}$$

(from Lecture 6)

Strong-Conservative form of the Navier-Stokes Equations ($\phi \Rightarrow v$)

Cons. of Momentum: $$\frac{d}{dt}\int_{CV}\rho\vec{v}dV+\int_{CS}\rho\vec{v}(\vec{v}.\vec{n})dA=\underbrace{\int_{CS}-p\,\vec{n}dA+\int_{CS}\vec{\vec{\tau}}.\vec{n}dA+\int_{CV}\rho\vec{g}dV}_{=\sum\vec{F}}$$

Applying the Gauss Theorem gives: $$=\int_{CV}\left(-\nabla p+\nabla.\vec{\vec{\tau}}+\rho\vec{g}\right)dV$$



For any arbitrary CV gives: $$\frac{\partial\rho\vec{v}}{\partial t}+\nabla.(\rho\vec{v}\;\vec{v})=-\nabla p+\nabla.\vec{\vec{\tau}}+\rho\vec{g}$$

With Newtonian fluid + incompressible + constant μ:

Momentum: $$\frac{\partial\rho\vec{v}}{\partial t}+\nabla.(\rho\vec{v}\;\vec{v})=-\nabla p+\mu\nabla^2\vec{v}+\rho\vec{g}$$

Mass: $$\nabla.\vec{v}=0$$

Equations are said to be in "strong conservative form" if all terms have the form of the divergence of a vector or a tensor. For the *i* th Cartesian component, in the general Newtonian fluid case:

With Newtonian fluid only: $$\frac{\partial\rho v_i}{\partial t}+\nabla.(\rho v_i\;\vec{v})=\nabla.\left(-p\,\vec{e}_i+\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\vec{e}_j-\frac{2}{3}\mu\frac{\partial u_j}{\partial x_j}\vec{e}_i+\rho g_i x_i\vec{e}_i\right)$$

Solution of the Navier-Stokes Equations

- In the FD and FV schemes, we dealt with the discretization of the generic conservation equation

$$\frac{\partial \rho\phi}{\partial t} + \nabla.(\rho\phi\vec{v}) = -\nabla.\vec{q}_\phi + s_\phi$$

- These results apply to the momentum and continuity equations (the NS equations), e.g. for incompressible flows, constant viscosity

$$\frac{\partial \rho\vec{v}}{\partial t} + \nabla.(\rho\vec{v}\;\vec{v}) = -\nabla p + \mu\nabla^2\vec{v} + \rho\vec{g}$$

$$\nabla.\vec{v} = 0$$

- Terms that are discretized similarly
  - Unsteady and advection terms: they have the same form for ϕ scalar than for $\phi = \mathbf{v}$
- Terms that are discretized differently
  - Momentum (vector) diffusive fluxes need to be treated in a bit more details
  - Pressure term has no analog in the generic conservation equation => needs special attention. It can be regarded either as a
    - source term (treated non-conservatively as a body force), or as,
    - surface force (conservative treatment)
  - Finally, main variable $\mathbf{v}$ is a vector $\Rightarrow$ gives more freedom to the choice of grids

Discretization of the Convective and Viscous Terms

- Convective term: $\nabla.(\rho\vec{v}\,\vec{v})$ and $\int_S \rho\vec{v}(\vec{v}.\vec{n})dS$ $\left(\dfrac{\partial(\rho u_i u_j)}{\partial x_j}\text{ and }\int_S \rho u_i(\vec{v}.\vec{n})dS\right)$
  – Use any of the schemes (FD or FV) that we have seen (including complex geometries)

- Viscous term: $\nabla.\vec{\vec{\tau}}$ and $\int_{CS} \vec{\vec{\tau}}.\vec{n}dA$ $\left(\dfrac{\partial \tau_{ij}}{\partial x_j}\text{ and }\int_S \tau_{ij}\,\vec{e}_j\,.\vec{n}\,dS\right)$
  – For a Newtonian Fluid and incompressible flows: $\tau_{ij} = \mu\left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right)$
    - If μ is constant, the viscous term is as in the general conservation eqn. for ϕ
    - If μ varies, its derivative needs to be evaluated
  – For a Newtonian fluid and compressible flow:
    - Additional terms need to be treated, e.g. $\dfrac{2}{3}\mu\dfrac{\partial u_j}{\partial x_j}\vec{e}_i$
  – Note that in non-Cartesian coordinate systems, new terms also arise that behave as a "body force", and can thus be treated explicitly or implicitly
    - e.g $-2\mu\dfrac{u_r}{r^2}$

Discretization of the Pressure term

- For conservative NS schemes, gravity/body-force terms often included in the “pressure” term, giving: $\boxed{\tilde{p} = p - \rho\,\mathbf{g}.\mathbf{r} + \mu\frac{2}{3}\nabla.\mathbf{u}}$ $(p\,\vec{e}_i - \rho g_i x_i \vec{e}_i + \frac{2}{3}\mu\frac{\partial u_j}{\partial x_j}\vec{e}_i)$
  - “Pressure” then part of the stress tensor (shows up as divergence in NS eqns.)
  - Last term is null for incompressible flows
- In non-conservative NS forms, the pressure gradient is discretized

- FD schemes
  - FD schemes seen earlier are directly applicable, but pressure can be discretized on a different grid than the velocity grid (staggered grid)
- FV schemes
  - Pressure usually treated a surface force (conservative form):
    - For the u_i equation: $\int_S -p\,\vec{e}_i.\vec{n}dS$
    - Again, schemes seen in previous lectures are applicable, but pressure nodes can be on a different CV grid
  - Pressure can also be treated non-conservatively: $\int_V -\nabla p.\vec{e}_i dV$
    - Discretization then introduces a global non-conservative error

Conservation Principles for NS

- Momentum and Mass Conservation
  - Momentum is conserved in any control volume in the sense that “it can only change because of flow through the CV surfaces, forces acting on these surfaces or volumetric body forces”
  - This property is inherited in the CV formulation (if surface fluxes are identical on both sides)
  - Similar statements for Mass conservation
- Conservation of important secondary quantities, e.g. energy
  - More complex issues
  - In heat transfer, thermal energy equation can be solved after momentum equation has been solved if properties don’t vary much with temperature T ⇒T is then a passive scalar, with one way coupling
  - In incompressible, isothermal flows: kinetic energy is the significant energy
  - In compressible flows: energy includes compressible terms

    ⇒ two equations can be written, one for kinetic or internal energy and one for the total energy

Conservation Principles for NS: Cont'd
Kinetic Energy Conservation

- Derivation of Kinetic energy equation
  - Take dot product of momentum equation with velocity
  - Integrate over a control volume CV or full volume of domain of interest
  - This gives

$$\boxed{\frac{\partial}{\partial t}\int_{CV}\rho\frac{\|\vec{v}\|^2}{2}dV=-\int_{CS}\rho\frac{\|\vec{v}\|^2}{2}(\vec{v}.\vec{n})dA-\int_{CS}p\,\vec{v}.\vec{n}\,dA+\int_{CS}(\vec{\vec{\varepsilon}}.\vec{v}).\vec{n}\,dA+\int_{CV}\left(-\vec{\vec{\varepsilon}}:\nabla\vec{v}+p\,\nabla.\vec{v}+\rho\vec{g}.\vec{v}\right)dV}$$

  where $\varepsilon_{ij}=\tau_{ij}+p\delta_{ij}$ is the viscous component of the stress tensor

  - Here, the three RHS terms in the volume integral are zero if the flow is inviscid (term 1 = dissipation), incompressible (term 2) and there are no body forces (term 3)
  - Other terms are surface terms and kinetic energy is conserved in this sense: ⇒ discretization on CV should ideally lead to no contribution over the volume
- Some observations
  - Guaranteeing global conservation of the *discrete* kinetic energy is not automatic since the kinetic energy equation is a consequence of the momentum equation. Discrete momentum and kinetic energy conservations cannot be enforced separately.

2.29 Numerical Fluid Mechanics
Fall 2011