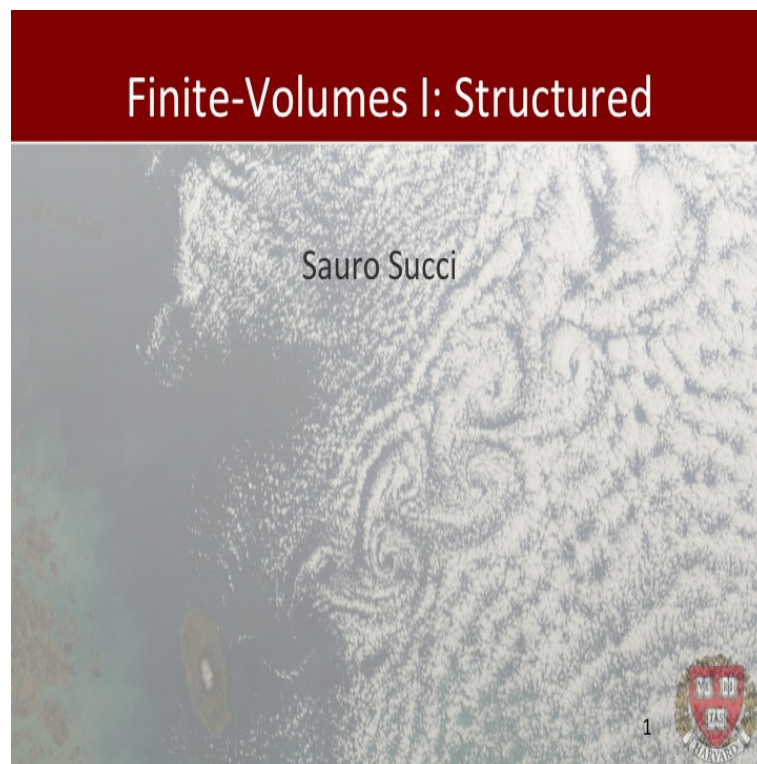


Lecture 10: Finite Volumes

October 20, 2015



1 Goal

As we have seen in the previous lectures, FD are very intuitive but lack flexibility and accuracy towards complex geometries. FEM solve the problem very effectively, but at the cost of a significant computational burden. As we shall see, the FVM offers a very effective intermediate strategy between the two. In this lecture we shall describe the main ideas behind the Finite Volume (FV) method, currently the most popular method for commercial CFD software.

2 Gauss theorems

The mathematical foundation of the FVM is the Gauss integral theorem

$$\oint_V (\nabla \cdot \vec{a}) dV = \oint_S \vec{a} \cdot \hat{n} dS$$

where \vec{a} is a generic vector. The above states that the integral of the divergence of a given vector over a closed volume is equal to the flux of this vector across a closed surface bounding the volume. In the rhs above \hat{n} denotes the outer normal to the surface.

The above theorem is particularly relevant to equations in conservation form. Consider for instance the continuity equation of fluid dynamics:

$$\partial_t \rho + \nabla \cdot \rho \vec{u} = 0$$

Gauss theorem gives

$$\frac{d}{dt} \oint_V \rho dV = - \oint_S \rho \vec{u} \cdot \hat{n} dS$$

In other words, the change in time of the total mass in the volume V is equal to the flux of the current across the surface bounding the volume. This permits to construct dynamic equations for volume-averaged quantities, such as the total mass in the volume, rather than pointlike ones, like in FD. This is interesting because i) volume averaged quantities make much more physical meaning than pointlike ones, ii) the FV formulation secures built-in conservation laws, which is an extremely useful property for numerical implementations.

3 FV formulation of Advection-Diffusion equation

Let us illustrate the practical set-up of a FV formulation for the case of a multidimensional advection-diffusion-equation:

$$\partial_t \phi = \nabla \cdot (-\vec{u} \phi + D \nabla \phi)$$

Note that the current \vec{J} contains an advective and convective components. The flow fields is assumed to be given, hence there is only one scalar unknown, the field ϕ . Although much simpler than fluid-dynamics, this case helps establishing a wide body of general notions of the FVM.

3.1 Geometry

First, let us define the geometry of the computational system. We consider a two-dimensional *structured* grid of finite volumes defined by the *nodes* P_i , $i = 1, NN$ and four connecting links to East, North, West and South neighbors. Each node is thus surrounded by four trapezoids, North-East $NE = (P, E, NE, N)$,

North-West $NW = (P, N, NW, W)$ and so on. To each node we associate a closed *control volume* (CV) obtained by joining link midpoints and cell centers, namely

$$CV(P) = \{e, ne, n, nw, w, sw, s, se, e\}$$

In the above e is the midpoint along link EP, ne is the barycenter of volume NE and so on (see Figure). A simpler choice is to define the CV only via the volume centers

$$CV(P) = \{se, ne, nw, sw, se\}$$

Obviously, the two come to the same for the case of cartesian volumes.

With the basic geometry in place, we next define the actual unknown as

$$\phi_P = \frac{1}{V_P} \oint_{V_P} \phi dV$$

where V_P is the CV centered around node P . For the case of two-dimensional volumes, Gauss theorem gives

$$V_P \frac{d\phi_P}{dt} = - \sum_{k=e,n,w,s} \rho_k u_k S_k$$

where the sum runs over the four (east,north,west, south) edges of the trapezoidal volume. In the above $u_k S_k \int_{S_k} \equiv \vec{u} \cdot \hat{n}_k dS_k$ is the surface integral of the projection of the velocity field on the k th surface along the normal to the surface, while S_k is the area (length in d=2) of the surface. Each volume V_P generates its own equation, so that by letting P run over the entire domain, a system of N equations is obtained, N being the number of volumes.

It should be noted, though, that the rhs involve the unknowns ϕ and \vec{u} on the surfaces S_k , while the actual unknowns are located at the cell (volume) centers P . Therefore, a procedure to transfer data from celle centers to cell surfaces is required. The procedure, basically an interpolation, is not unique, as it depends on the order of interpolation used. However, with reference with a structured grid in which the actual volume V_P is surrounded by eight neighbors (East,North,West,South, North-East,North-West,South-West,South-East), a simple interpolation would read as follows

$$\phi_e = a_{PE}\phi_P + (1 - a_{PE})\phi_E$$

and similar for the other four faces. Here lower capital indices denote surface values while upper capital indices denote volume centers (see Slides). The interpolation coefficients are given by

$$a_{PE} = |eP|/|EP|$$

where eP denotes the vector $e - P$ and $EP = E - P$.

By the time the interpolation is performed on all four cells, the surface flux picks up contribution from all four neighbors E,N,W,S, and possibly also the next-neighbors NE,NW,SW,SE.

Thus, the equation of motion for the unknowns take the form

$$\frac{d\phi_P}{dt} = - \sum_{Q=E,N,W,S,\dots} C_{PQ}\phi_Q \quad (1)$$

one for each volume. The connectivity coefficients C_{PQ} are linear combinations of the interpolation coefficients a_{PQ} , as well as of the geometric factors S_k/V_P associated with the FV geometry. This is a self contained system of ordinary differential equations, and can be discretized in time with one's favored time-marching scheme.

A few general considerations are in order. First, we note that the flux across a surface separating nodes P and Q is the same as the one crossing from Q to P, with just opposite sign, i.e.:

$$\Phi_{PQ} = -\Phi_{QP}$$

This identity guarantees that mass conservation is achieved to machine roundoff.

Generally, the "quality" of the matrix depends on the shape of the volumes. For instance, once time marching is taken, the connectivity coefficients pick up dimensionless factors

$$CFL_k \equiv S_k u_k \Delta t / V_P$$

which are clearly reminiscent of the CFL condition. This shows that very thin, elongated volumes, generating high values of S/V lead to very strong restrictions on the timestep (if an explicit method is used). Likewise, highly distorted volumes, far from square shape, lead to ill-conditioned matrix problems, and should be avoided as much as possible. That is, the grid should be such to keep local CFL close to a constant value everywhere, which means that the grid should adapt to the flow \vec{u} .

See Slides figs.

4 Diffusive fluxes

The machinery described above applies verbatim to the diffusive fluxes as well, with the important caveat that now the surface integrals involve gradients of the unknown, i.e.

$$\Phi_k^D = \int_{S_k} \nabla\phi \cdot \hat{n} dS$$

Following the procedure described for the advective fluxes, one can approximate the flux integral as

$$\Phi_k^D \sim S_k g_k$$

where $g_k \equiv \nabla\phi \cdot \hat{n}_k$ at position r_k on the surface S_k .

With reference to the east surface,

$$\partial_x\phi|_e = (\phi_E - \phi_P)/(x_E - x_P)$$

and similarly for the y component.

Caveat: this is very dangerous, since $y_E - y_P$ can be very small or even zero!
What to do? Take the data along the y direction.

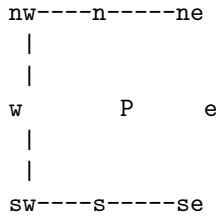
$$\partial_y \phi_e = (\phi_{ne} - \phi_{se}) / (y_{ne} - y_{se})$$

where ne is interpolated from P, E, NE, N and se from N, NW, SW, P .

5 Practical example

Let us derive the FV equations for a rectangular FV grid with mesh size a and b along x and y .

See Slides.



Basic geometrical quantities:

Volume centers: $P = (x, y)$, $E = (x + a, y)$, $W = (x - a, y)$, $N = (x, y + b)$,
 $S = (x, y - b)$.

Edges: $e = (x + a/2, y)$, $w = (x - a/2, y)$, $n = (x, y + b/2)$, $s = (x, y - b/2)$

Volume and edge length: $V = ab$, $S_e = S_w = b$, $S_n = S_s = a$.

Convective CFL's:

$S_e u_e h / V = u_x h / a$, $S_w u_w h / V = -u_x h / a$, $S_n u_n h / V = u_y h / b$, $S_s u_s h / V = -u_y h / b$.

Hence the sum of the fluxes is:

$$\frac{[u_x \phi]_e - [u_x \phi]_w}{a} + \frac{[u_y \phi]_n - [u_y \phi]_s}{b}$$

Next we interpolate the surface values as follows

$$\phi_e = (\phi_P + \phi_E) / 2, \phi_w = (\phi_P + \phi_W) / 2, \phi_n = (\phi_P + \phi_N) / 2, \phi_s = (\phi_P + \phi_S) / 2$$

so that the sum of the fluxes becomes (take $u_x = u_y = U = \text{constant}$):

$$(\phi_E - \phi_W)U / 2a + (\phi_N - \phi_S)U / 2b$$

which coincides with a centered finite difference scheme on a rectangular mesh.

Diffusive fluxes:

With $g_a \equiv \partial_a \phi$, $a = x, y$.

$$\begin{aligned}\text{East: } \Phi(e) &= Dg_x(e)b, \\ g_x(e) &= (\phi(E) - \phi(P))/a, \\ g_x(w) &= (\phi(P) - \phi(W))/a, \\ g_x(e) - g_x(w) &= [\phi(E) - 2\phi(P) + \phi(W)]/a, \\ g_y(n) - g_y(s) &= [\phi(N) - 2\phi(P) + \phi(S)]/b,\end{aligned}$$

6 Boundary conditions

Essentially, like for FD schemes, if the conditions are on the fluxes (von Neumann) even simpler.

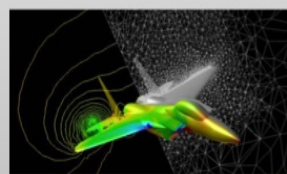
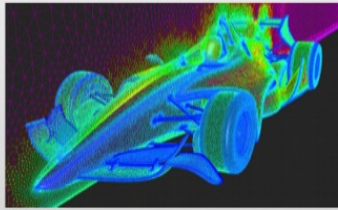
7 Exercises

Write a computer program to solve the diffusion equation with the FVM method, test the accuracy and stability upon changing the size and shape of the volumes.

Motivation

Real-life geometries: coordinate-free

Courtesy of Prof. M. Porfirí, NYU



2



Purpose

Most suited to CONSERVATIVE equations in the form
 $d/dt * \text{DENSITY} + \text{divergence FLUX} = \text{Source}$, where Density is a tensor of order
0,1,2 (scalar,vector,tensor) FLUX is a tensor of order 1,2,3.

$$\partial_t \varphi + \nabla \cdot \vec{\Phi} = S$$

For ADR eqs:

$$\vec{\Phi} = -\bar{u}\varphi + D\nabla\varphi$$

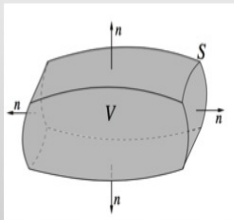
$$S = R(\varphi)\varphi$$

3



Gauss theorem

Given a Volume V ,
enclosed in a piecewise smooth boundary (surface) S ,
characterized by the normal \mathbf{n} in each point:



$$\oint (\nabla \cdot \vec{\Phi}) dV = \oint (\vec{\Phi} \cdot \hat{n}) dS$$

Nice, since it lowers the differential order by one.

**Price: densities live in the bulk, fluxes live on the surf;
Bulk2Surf and viceversa communication is required**

4

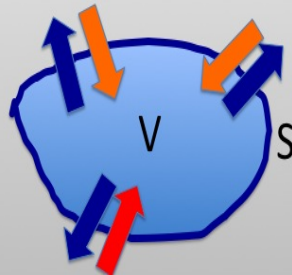


Gauss theorem: continuity equation

$$\partial_t \rho + \nabla \cdot (\rho \vec{u}) = 0$$

$$\frac{d}{dt} \oint \rho dV = - \oint (\rho \vec{u} \cdot \hat{n}) dS$$

Mass change rate = Flux_in - Flux_out



5



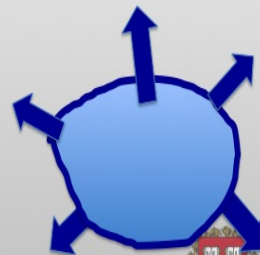
Gauss theorem: ADE

$$\partial_t \varphi = -\nabla \cdot (\varphi \vec{u} - D \nabla \varphi)$$

$$\frac{d}{dt} \oint \varphi dV = \oint (-\varphi \vec{u} + D \nabla \varphi) \cdot \hat{n} dS$$

$$\vec{u} \cdot \hat{n} = u_x n_x + u_y n_y + u_z n_z$$

$$\nabla \varphi \cdot \hat{n} = n_x \partial_x \varphi + n_y \partial_y \varphi + n_z \partial_z \varphi$$



6



Gauss theorem: NSE

$$\partial_t(\rho \vec{u}) + \nabla \cdot [\rho \vec{u} \vec{u} + p \vec{I} - \vec{\sigma}(\nabla u)] = \vec{f}$$

$$\frac{d}{dt} \oint (\rho \vec{u}) dV = \oint (-\rho \vec{u} \vec{u} - p \vec{I} + \vec{\sigma}) \cdot \hat{n} dS + \oint \vec{f} dV$$

Bivectors: [normal;vector]

$$\vec{u} \vec{u} \cdot \hat{n} = \begin{bmatrix} u_x(u_x n_x + u_y n_y + u_z n_z) \\ u_y(u_x n_x + u_y n_y + u_z n_z) \\ u_z(u_x n_x + u_y n_y + u_z n_z) \end{bmatrix}; \quad p \vec{I} \cdot \hat{n} = \begin{bmatrix} p_x n_x \\ p_y n_y \\ p_z n_z \end{bmatrix}$$



7

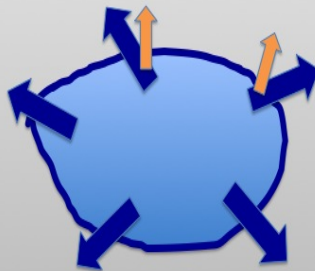


Gauss theorem: conservation laws

$$\vec{J} = \rho \vec{U}(\rho)$$

Volume and Surface average:

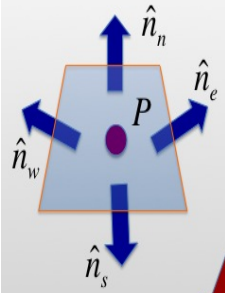
$$\dot{M}_P \equiv \dot{\rho}_P V_P = -S \bar{\rho} \bar{U}$$



8



Gauss theorem: Control Volume




$$\frac{d}{dt} \oint \varphi dV = \oint (-\varphi \vec{u} + D\nabla \varphi) \cdot d\vec{S}$$

Volume average: more physical!

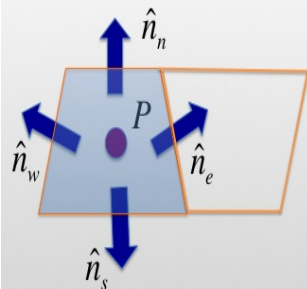
$$\varphi_P = \frac{1}{V_P} \oint \varphi dV$$

$A_k \equiv S_k$

$$V_P \frac{d\varphi_P}{dt} = \sum_{k=\{e,n,w,s\}} A_k [-\varphi \vec{u} + D\nabla \varphi]_k \cdot \hat{n}_k$$

9 

From cell centers to cell surface




Volume average: more physical!

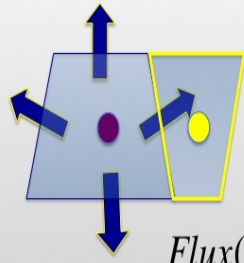
$$\varphi_P = \frac{1}{V_P} \oint \varphi dV$$

$$V_P \frac{d\varphi_P}{dt} = \sum_{k=\{e,n,w,s\}} A_k [-\varphi \vec{u} + D\nabla \varphi]_k \cdot \hat{n}_k$$

The values at volume edges must be computed via interpolation, i.e. Using information from neighboring volumes.
The interpolation is not unique.
Before going into this matter, some general considerations

10 

Conservativeness



$$\text{Flux}(P,Q) \equiv -\text{Flux}(Q,P)$$

The outgoing flux from one cell to the neighbor cell is the same as the incoming one, with opposite sign:

Conservativeness is preserved exactly

11



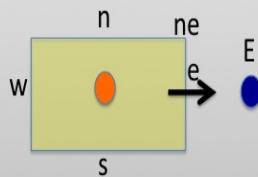
Centers to Edges: Interpolate

For a simple CARTESIAN grid:

$$\Phi_e = (1 - a_e)\Phi_P + a_e\Phi_E$$

Order 0: $a_e = 1$

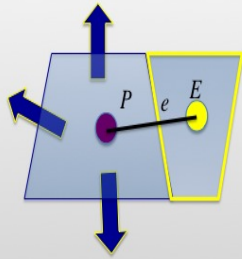
Order 1: $a_e = \frac{x_e - x_P}{x_E - x_P}$



12



Non cartesian, structured



$$a_e = 1 - \frac{l_{eP}}{l_{EP}}$$

$$l_{EP} = |\vec{r}_E - \vec{r}_P|$$

$$l_{eP} = |\vec{r}_e - \vec{r}_P|$$

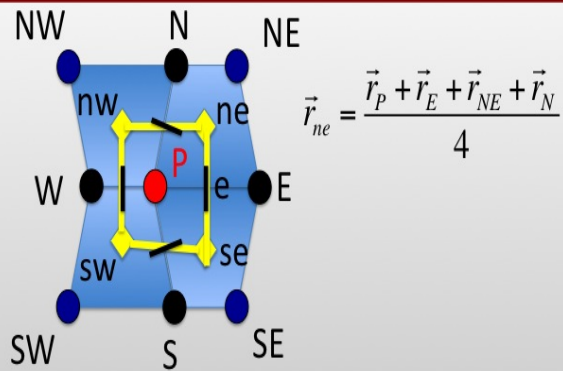
All geometrical info has to be stored:

Coordinates of volume centers; volumes, normals to the edges...

13



Non-cartesian, structured



Note that the control volume is dual to the mesh {P,E,NE,N,NW,W,SW,S,SE...}.

The segment PE is not necessarily aligned with the normal at the east edge. This affects the flux calculation

14

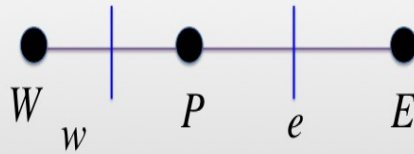


1d and 2d example: advection-diffusion

15



One Dimension



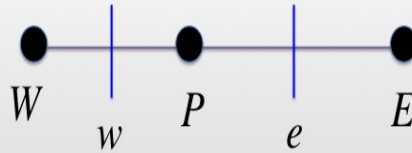
$$\varphi_P(t + \Delta t) = \varphi_P(t) + \frac{U\Delta t}{\Delta x} \sum_{Q \in \{W, P, E\}} L_{PQ} \varphi_Q(t)$$

$$Flux_{PQ} + Flux_{QP} = 0$$

16



Putting all together



$$\varphi_p(t + \Delta t) = \varphi_w(t) \left[\frac{U_e \Delta t}{\Delta x} a_w + \frac{1}{\lambda_w} \frac{D_w \Delta t}{\Delta x^2} \right] + \varphi_e(t) \left[-\frac{U_e \Delta t}{\Delta x} a_e + \frac{1}{\lambda_e} \frac{D_e \Delta t}{\Delta x^2} \right] + \varphi_p(t) \left[1 - \frac{\Delta t}{\Delta x} (U_e a_e - U_w a_w) - (D_w + D_e) \frac{\Delta t}{\Delta x^2} \right]$$

$$a_e = (x_p - x_e) / (x_p - x_e) \quad a_w = (x_p - x_w) / (x_p - x_w) \quad \text{Interpolation}$$

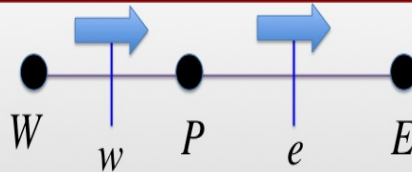
$$D_e = D \Delta x / (x_p - x_e) \quad D_w = D \Delta x / (x_p - x_w) \quad \text{Non-uniformity}$$

Still tridiagonal form (time-explicit), but coeffs are more involved, as they reflect the geometry

$$\varphi_p(t + \Delta t) = L_{PW} \varphi_w(t) + L_{PP} \varphi_p(t) + L_{PE} \varphi_e(t) \quad 20$$



Advective fluxes



$$[U\varphi]_e = a_e [U\varphi]_E + (1 - a_e) [U\varphi]_P$$

$$[U\varphi]_w = a_w [U\varphi]_W + (1 - a_w) [U\varphi]_P$$

$$\text{Linear Interp: } a_e = 1 - \frac{x_E - x_e}{x_E - x_P} \quad a_w = 1 - \frac{x_P - x_w}{x_P - x_W}$$

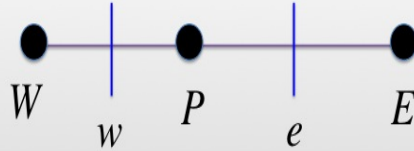
Upwind:

$$a_e = 0 \quad \text{if } U_e > 0 \quad a_w = 1 \quad \text{if } U_w > 0$$

$$a_e = 1 \quad \text{if } U_e < 0 \quad a_w = 0 \quad \text{if } U_w < 0$$



1d Diffusive fluxes



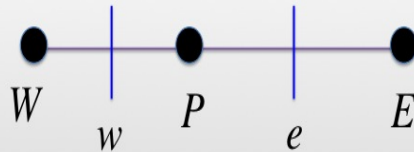
$$[D\partial_x\varphi]_e = D[\varphi_E - \varphi_P]/[x_E - x_P]$$

$$[D\partial_x\varphi]_w = D[\varphi_P - \varphi_W]/[x_P - x_W]$$

19



FV versus FD



$$\varphi_P(t + \Delta t) = L_{PW}\varphi_W(t) + L_{PP}\varphi_P(t) + L_{PE}\varphi_E(t)$$

- CFL's still recognizable, although they are "dressed" by geometrical factors
- Still tridiagonal because we have used Euler, with CN would be a matrix problem
- What do we gain wrt finite-differences?
- Non-uniformity does no harm, since we are dealing with VOLUME-AVERAGED quantities, not pointwise ones like in FD!

21

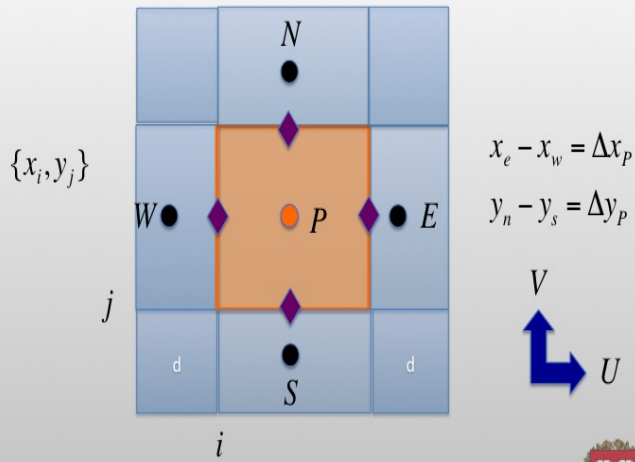


2d : advection-diffusion

22



Cartesian d=2



23



FV: AD $d=2$

$$\varphi_P(t + \Delta t) = \varphi_P(t) + (\Phi_e + \Phi_n + \Phi_w + \Phi_s)\Delta t$$

Discretized Fluxes

$$\Phi_k^A = - \int_{S_k} \varphi \vec{U} \cdot \hat{n}_k dS$$

$$\Phi_k^D = \int_{S_k} D \nabla \varphi \cdot \hat{n}_k dS$$

24



Structured cartesian: Advection fluxes

$$\Phi_e = U_e \varphi_e \Delta y_P \quad \Phi_w = U_w \varphi_w \Delta y_P$$

$$\Phi_n = V_n \varphi_n \Delta x_P \quad \Phi_s = V_s \varphi_s \Delta x_P$$

$$\Delta_E x = x_E - x_P = \frac{\Delta x_P + \Delta x_E}{2} \quad \Delta_W x = x_W - x_P$$

$$\Delta_N y = y_N - y_P \quad \Delta_S y = y_S - y_P$$

$$\varphi_e = (1 - a_E) \varphi_P + a_E \varphi_E \quad a_E = \frac{\Delta x_P}{\Delta x_P + \Delta x_E}$$

$$U_e = -\vec{u} \cdot \hat{n}_e = U$$

25



$$Adv_Flux = -\alpha_e a_E \varphi_E + -\alpha_w a_W \varphi_W - \alpha_n a_N \varphi_N + \alpha_s a_S \varphi_S +$$

$$[-\alpha_e(1-a_E) + \alpha_w(1-a_W) - \alpha_n(1-a_N) + \alpha_s(1-a_S)]\varphi_P$$

$$\alpha_e = U_e \Delta y_P \quad \alpha_n = V_n \Delta x_P$$

$$Adv_Flux = \sum_{Q=\{P,E,N,W,S\}} A_{PQ} \varphi_Q$$

$$V_P \frac{d\varphi_P}{dt} = \sum_{Q=\{P,E,N,W,S\}} A_{PQ} \varphi_Q$$

Structured matrix:
5 nonzero entries

26



27



Goal: same CFL everywhere!

$$CFL = UhS / V = const$$

$$U \gg V \quad \text{[blue bar]} \quad \partial_y U \gg \partial_x V$$

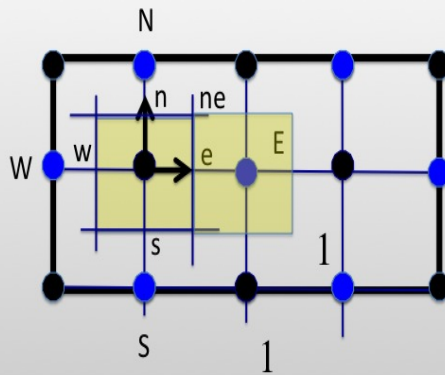
$$Uh / dx = const \Rightarrow dx \propto U$$

$$Vh / dy = const \Rightarrow dy \propto V$$

28



Cartesian colocated



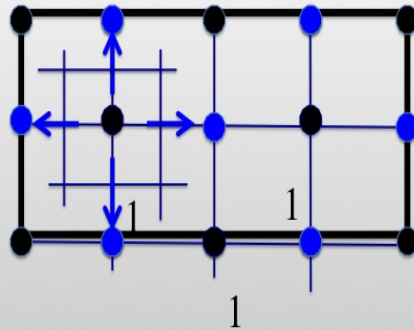
Simple, but not good for surfint > geos

VP uncoupled, hourglass

29



Cartesian staggered



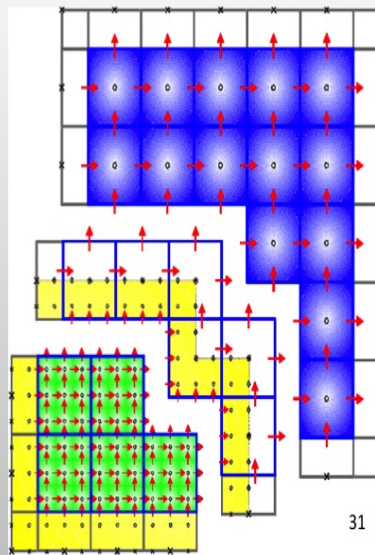
Laborious, no interpolation > simple geos

No hourglass, VP coupled

30



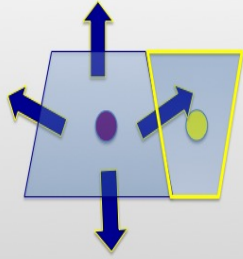
Two-dimensional staggered



31



Non cartesian



Cumbersome: The flux vector is generally not aligned with the normal, hence both all vector (tensor) components contribute $U_x n_x + U_y n_y$ and so on. It is basically a non-orthogonal grid.

32



Topological issues

$$C = hS/V \gg 1 \quad l_x \sim l_y \sim l_z \sim V^{1/3}$$

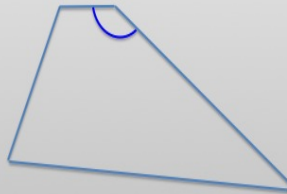
$S/V \gg 1$



$\vartheta \ll \pi/2$



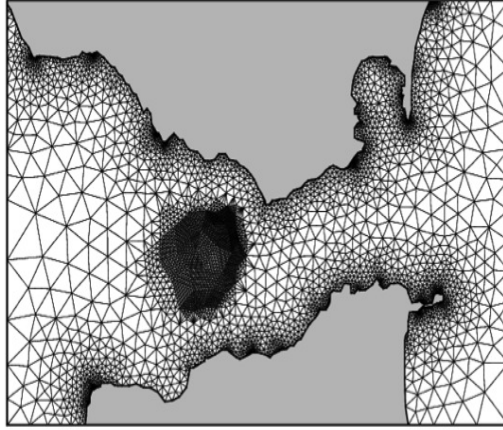
$\vartheta \gg \pi/2$



33



Unstructured



The volumes must be indexed sequentially, no ijk ordering.
Store the node locations and their connectivity, plus normals, volumes. Significant labor, but once and for all (for static geometries)



FV of increasing complexity

Cartesian uniform



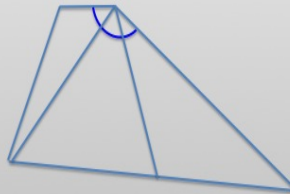
Cartesian non-uniform



Non-Cartesian, Structured



Unstructured



35



Going to 2 and 3D: same principles but more labour!

Structured Cartesian (Orthogonal), similar to FD

Structured Trapezoidal (non-orthogonal)

Unstructured (similar to FEM)

36



Assignments

1. Solve the 1d ADR equation using uniform rectangular Volumes and explicit Euler

2. Same with non-uniform volumes

3. Same with Crank-Nicolson

4. For the incredibly brave:
3d unstructured with Crank-Nicolson!

5. 4. is a joke! (-:

37



End Lecture

38

