Lecture 21

Intro to CFD (Fluent)

1. Fluent instructions / tutorial

2. General idea of gaseous combustion model

3. Governing equations
   - overall continuity
   - species continuity
   - energy

4. Elliptic vs. Parabolic Eqns.

5. Pressure

General Idea of Comprehensive Model

1. Pick geometry

2. Pick variables
   - turbulent vs laminar
   - gaseous vs particle-laden
   - adiabatic vs heat losses
   - radiation?
   - reacting vs non-reacting
   - steady-state vs transient

3. Set up grid system to solve basic equations
   - cartesian (x-y-z) vs cylindrical axi-symmetric (r-z-θ)
   - staggered vs. non-staggered (collocated)
   - "stair-step" boundary vs. more complicated coordinate mapping (like body-fitted coordinates, unstructured grid etc.)
   - finite difference vs finite volume vs finite element

GOAL: Solve all conservation equations for continuum phase as painlessly as possible with one algorithm!

2. Governing Equations

A. Overall Continuity

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot \rho \vec{v} = S_{pm}$$

($\frac{\partial \rho}{\partial t} = 0$ @ steady-state)

S_{pm} = mass source term from condensed phase
= 0 for gaseous combustion only

B. Species Continuity

$$\frac{\partial \rho_i}{\partial t} + \vec{\nabla} \cdot \vec{n}_i = r_i \qquad \text{BSL 18.1-6}$$

$\rho_i = \rho \omega_i$ $\qquad \omega_i$ = mass fraction

r_i = net production rate of species i, including source (if any) from condensed phase

$$\vec{n}_i = -\rho \mathcal{D}_{ij} \vec{\nabla} \omega_i + \omega_i \vec{n}_T$$

$$\vec{n}_T = \rho \vec{v} \quad \Rightarrow \quad \omega_i \vec{n}_T = \rho_i \vec{v}$$

$$\boxed{\frac{\partial \rho_i}{\partial t} + \vec{\nabla} \cdot \rho_i \vec{v} = \vec{\nabla} \cdot \left(\rho \mathcal{D}_{ij} \vec{\nabla} \omega_i \right) + r_i} \qquad \text{(BSL 18.1-14)}$$

transient | convection | diffusion | production from reaction

Momentum

$$\frac{\partial \rho \vec{v}}{\partial t} = -\left[\vec{\nabla} \cdot \rho \vec{v}\vec{v}\right] - \vec{\nabla} p - \left[\vec{\nabla} \cdot \bar{\tau}\right] + \rho \vec{g} + F_p$$

transient term — momentum exchange — pressure — viscous forces — gravity — interaction with solid phase

BS; L 3.2-10

(Table 10.4-1)

$$\tau_{ij} = \left[-\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \delta_{ij}\frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}$$

laminar

BS; L 3.2-11 -16

kno 3-57

Energy

pick h or T (usually h in large systems)

$$\rho\frac{D\hat{H}}{Dt} = -\nabla \cdot q + \frac{Dp}{Dt} - \tau : \nabla v + \cancel{\varepsilon_{ji} g_i}^{\to 0}$$

BS; L 18.3-1E

$$\rho\left[\frac{\partial \hat{H}}{\partial t} + \vec{v} \cdot \nabla \hat{H}\right] = -\nabla \cdot q + \left[\frac{\partial p}{\partial t} + \vec{v} \cdot \nabla p\right] - \tau : \nabla v$$

multiply continuity eqn. by $\hat{H}$

$$\hat{H}\frac{\partial \rho}{\partial t} + \hat{H}\vec{\nabla}\rho\vec{v} = \hat{H} S_{pm}$$

Now add to energy eqn.

$$\underbrace{\hat{H}\frac{\partial \rho}{\partial t} + \rho\frac{\partial \hat{H}}{\partial t}}_{\frac{\partial(\rho\hat{H})}{\partial t}} + \underbrace{\hat{H}\nabla \cdot \rho\vec{v} + \rho\vec{v} \cdot \nabla\hat{H}}_{\vec{\nabla}\cdot(\rho\hat{H}\vec{v})} = -\nabla \cdot q + \frac{\partial p}{\partial t} + \vec{v} \cdot \nabla p - \vec{\tau} : \nabla\vec{v} + \hat{H} S_{pm}$$

Also, remember that $q = q^{(c)} + q^{(d)}$ BS; L 18.4-2

$$= -k\nabla T + \sum \hat{H}_{ji}\vec{j}$$

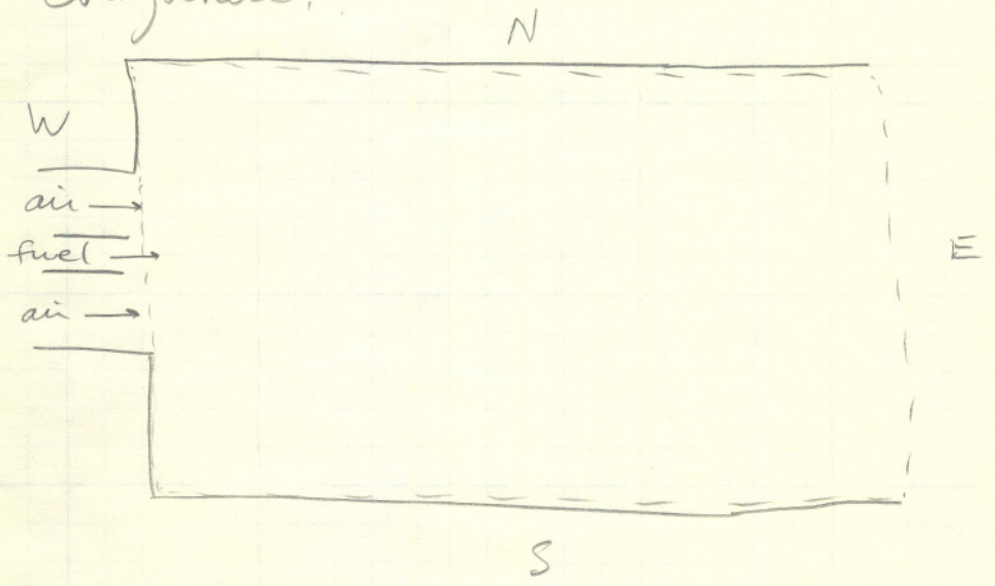
$$A\frac{\partial^2 u}{\partial \xi^2} + B\frac{\partial^2 u}{\partial \xi \partial \eta} + C\frac{\partial^2 u}{\partial \eta^2} + \ldots = 0$$

$B\frac{\partial^2 u}{\partial \xi \partial \eta}$: only in viscous terms $\Rightarrow B \to 0$

$$B^2 - 4AC \doteq -4AC < 0$$

elliptic eqns!

In other words, we need boundary conditions everywhere!!



Boundary Conditions needed

- inlets
- walls
- outlets
- symmetry axis (if any)

flow (flux)
- set boundary
- extrapolated boundary

Pressure (Makes problem difficult)

→ What equation do we solve for pressure?

- we have to monkey around, using the Continuity and Momentum equations!

– Is it important?

How much velocity can a $\Delta P = 1$ Pascal drive?

$$-\Delta P = \frac{\rho \Delta(v^2)}{2} \quad \text{(Bernoulli's eqn.)}$$

$$1 \text{ Pascal} = 0.987 \times 10^{-5} \text{ atm} = 1 \frac{kg}{m^2 s}$$

$\rho =$ @ 1500 k, 1 atm $= .2322 \frac{kg}{m^3}$

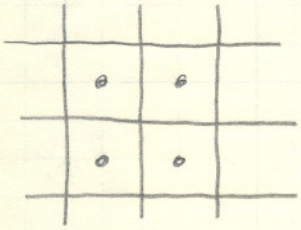
$$\frac{1 \frac{kg}{m^2 s}}{.2322 \frac{kg}{m^3}} = 4.3 \frac{m}{s} = \Delta(v^2) = v_2^2 - v_1^2$$

$v_1 = 0$	$v_1 = 5$ m/s	$v_1 = 20$ m/s
$v_2 = 2.07$ m/s	$v_2 = 5.4$ m/s	$v_2 = 20.1$ m/s
$\Delta v = 207$ cm/s	$\Delta v = 40$ cm/s	$\Delta v = .1$ cm/s

message: small Δp can cause large Δv, especially when v is small to begin with!

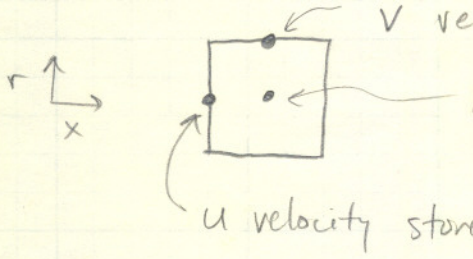
GRIDDING CONCEPTS

A. Collocated grid



- All variables are stored at grid centers (nodes)
- Interpolation necessary when properties at cell interfaces are needed.

B. Staggered grid (used in PCGC-3, Fluent)



Reason: when you integrate the PDE's, velocities are needed at the faces of P-cells, so why not store them there instead of interpolating all the time

Advantages: - Code, numerical stability (better than standard collocated grid)

Disadvantages: - complexity, especially w/ non-uniformly spaced grids
- hard to implement for BFC, advanced gridding concepts

U, V, W cells now constitute whole new grid cell systems

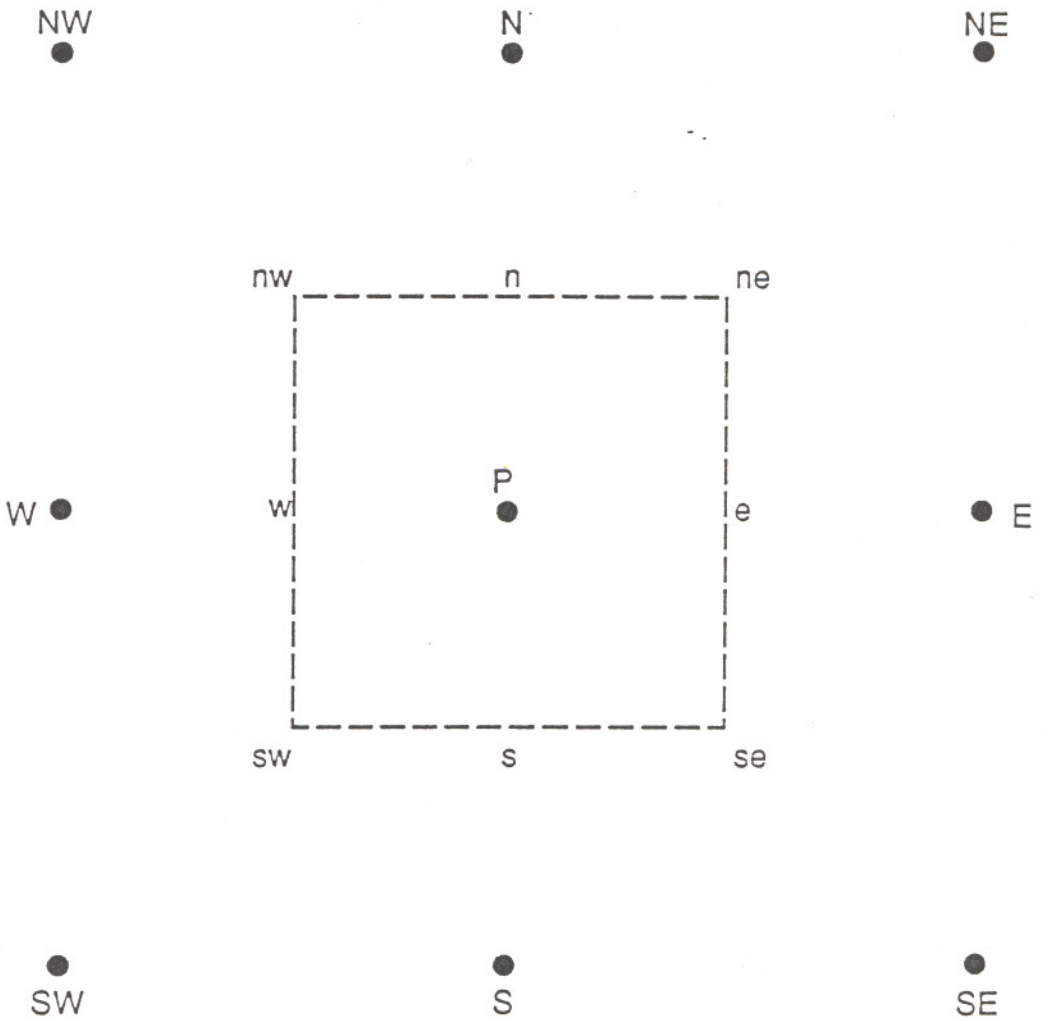


Figure 10. Illustration of the grid symbols for a computational cell.

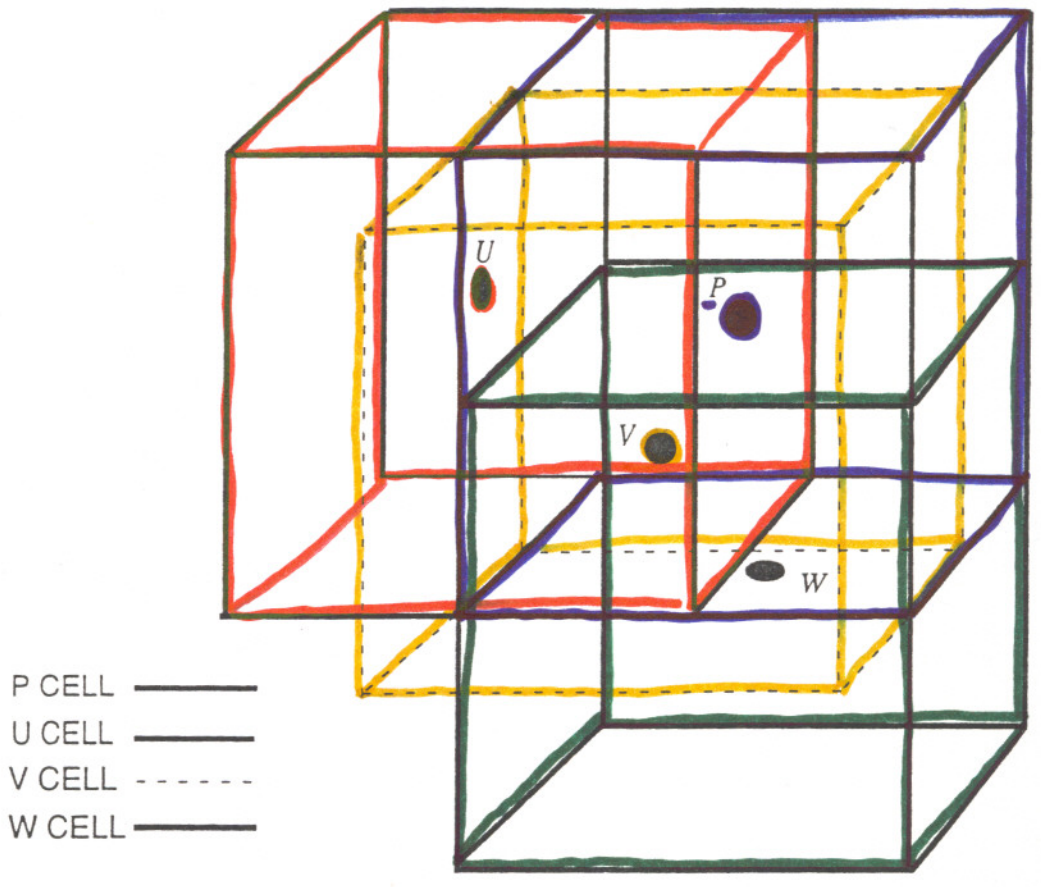


Figure 3.12: Side View of the Computational Cell Types Used in PCGC-3

Basic TEACH algorithm

(method for getting algebraic equations from PDE's based on finite-volume integration of PDE terms)

A. Put all equations in a standard form (2-d r-z coordinates shown here)

$$0 = \frac{\partial}{\partial x}(\rho u r \phi) + \frac{\partial}{\partial r}(\rho v r \phi) - \frac{\partial}{\partial x}\left(r\Gamma \frac{\partial \phi}{\partial x}\right) - \frac{\partial}{\partial r}\left(r\Gamma \frac{\partial \phi}{\partial x}\right) - r S^{\phi}$$

B. Integrate these eqns. over the P cell, one term at a time.

$$\int_{x_w}^{x_e}\int_{r_s}^{r_n}\int_{0}^{2\pi} [\ \text{equ.}\ ]\, dx\, dr\, d\theta = 0 = \int_{x_w}^{x_e}\int_{r_s}^{r_n} (\ )\, dx\, dr$$

due to symmetry, multiplying by 2π.

1st convection term

$$= \int_{x_w}^{x_e}\int_{r_s}^{r_n} \frac{\partial}{\partial x}(\rho u r \phi)\, dx\, dr = \int_{r_s}^{r_n} \{\rho u r \phi\}_w^e\, dr = \int_{r_s}^{r_n} \{\rho u \phi\}_w^e\, r\, dr$$

(integrate a derivative)

using mean value,

$$= \left(\left[\overline{\rho u \phi}\right]_e - \left[\overline{\rho u \phi}\right]_w\right)\left(\frac{r_n^2}{2} - \frac{r_s^2}{2}\right)$$

Because of staggered grid, $u_e = u_E$ ← node of u-cell on East side, using capital "E"

$u_w = u_W$

$$= \left[\rho_e u_E \phi_e - \rho_w u_W \phi_w\right] A_{ew}$$

↑ interpolate to get these

Define:

$C_E = \rho_e u_E A_{ew}$ (convection coefficients)

$C_W = \rho_w u_W A_{ew}$

Similar stuff to other terms (radial convection, also diffusion terms)

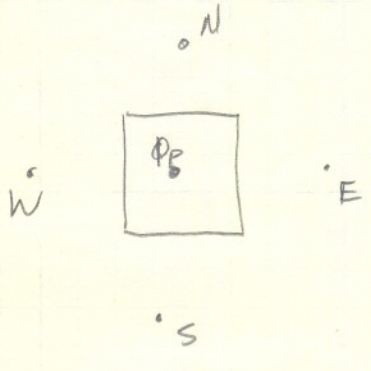
Define: $A_E = D_E - \frac{1}{2} C_E$

↑ diffusion coefficient ↑ convection coefficient

Final form

$$\left[A_E + A_W + A_N + A_S\right] \phi_P = A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + \underbrace{S_u + S_\phi \phi_P}_{\text{leftover terms}}$$

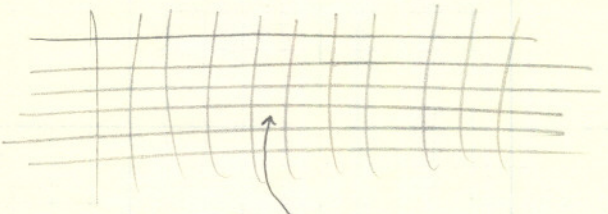
(some tricks then played for convergence)



* This says that ϕ_P depends on contributions from surrounding neighbors.

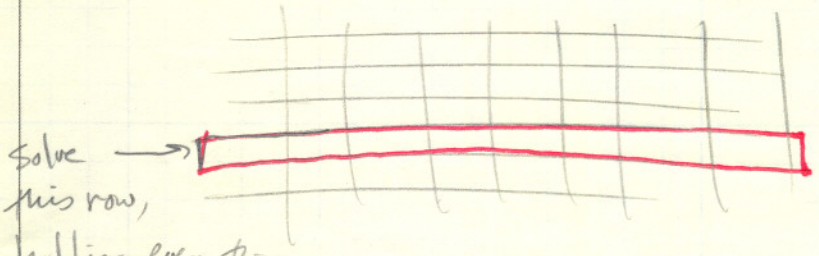
– note: upwind, power-law & hybrid differencing incorporated.

Matrix Solution Algorithm

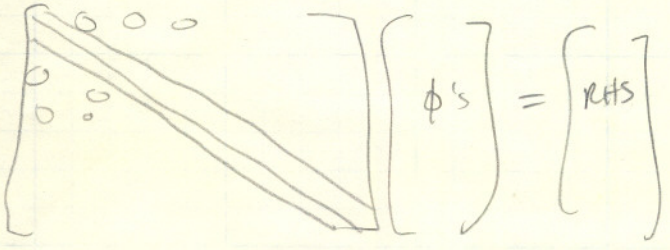


for each cell i,j, we have $\underbrace{\left(\sum_d A_d - S_p\right)}_{A_p}\phi_p = \sum_{N,S,E,W}(A_d \phi_d) + S_u$

Therefore, use successive tri-diagonal algorithms to iterate



Tridiagonal matrix



or

$$D_i \phi_i = A_i \phi_{i+1} + B_i \phi_{i-1} + C_i$$

$\Downarrow$

gets row (or column) of ϕ's calculated

<u>why?</u> $\Rightarrow$ large matrices are hard to invert (time; memory)

Convergence tricks

A. Flow chart

1. Pick one variable (i.e., u, v, γ_i, h, etc.)
2. Calculate A_P, A_d's, S_u for that variable based on latest guesses for other variables at each cell!
3. Do row by row & then column by column TDMA for that variable until residual for that variable is low.

$$RES_{ij} = \left(A_P \phi_P - \Sigma A_d \phi_d - S_u\right)_{ij}$$

$$Res_{tot} = \Sigma RES_{ij}$$

4. Go to next variable until all variables are solved
5. Start over with first variable, based on newest updates, until all residuals are low

B. Under-relaxation

Convention says we need to under-relax to try to get convergence

$$\phi^{new} = \lambda \phi^{new} + (1-\lambda)\phi^{old}$$

λ = under-relaxation factor between 0 and 1

1 = no under-relaxation

0 = no change (i.e., no solution other than 1st guess)

In practice, the source terms are under-relaxed

$$A_P^{new} = \frac{A_P^{old}}{\lambda}$$

$$S_u^{new} = S_u^{old} + \frac{(1-\lambda)\phi_P^{old} A_P^{new}}{\lambda}$$ or something similar