ME 466
Computational Fluid Dynamics (Elective)

Dr. Ajith Kumar A

Assistant Professor,

Dept. of Mechanical Engineering,

Rajagiri School of Engineering and Technology, India

Module I & II

Introduction to

CFD,FVM, Discretization

What is CFD? Overview

- **Computational Fluid Dynamics (CFD) is the science of predicting fluid flow, heat and mass transfer, chemical reactions, and related phenomena by solving numerically the set of governing mathematical equations (GE)**
  - Conservation of mass
  - Conservation of momentum
  - Conservation of energy
  - Conservation of species
  - Effects of body forces

- **The results of CFD analyses are relevant in:**
  - Conceptual studies of new designs
  - Detailed product development
  - Troubleshooting
  - Redesign

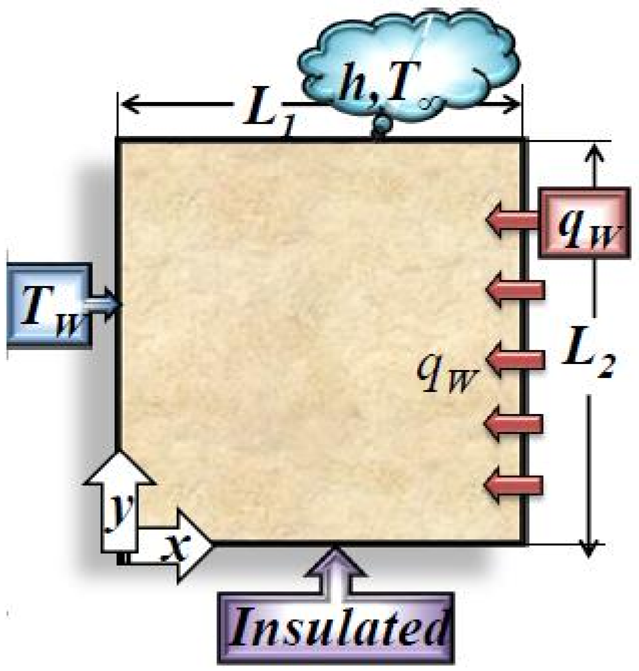
- **CFD doesn't negate the requirement of testing and experimentation.**
  - **It only reduces effort and cost required for experimentation and data acquisition**

CFD: Development Methodology

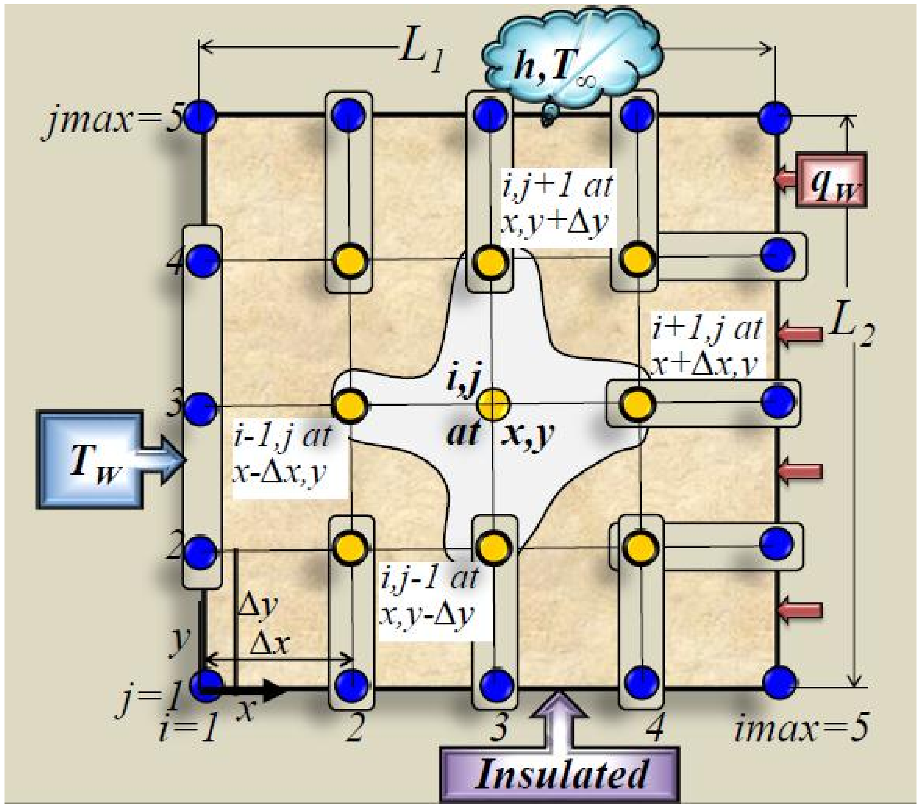
- 1. GRID GENERATION
  - The entire domain is divided into finite divisions
    - Finite locations (grids in FDM) OR Control Volumes (CV in FVM)
- 2. DISCRETIZATION METHOD
  - Algebraic formulation (LAEs) of Governing equations (PDEs)
- 3. SOLUTION METHODOLOGY (SOLVER)
  - Defining the methods to solve system of LAEs
    - Explicit and implicit methods for unsteady formulation
    - Iterative method for steady state formulation
  - Implementation details
    - Computer programming involved
  - Solution algorithm
    - Algorithm to call the various computer programmes involved
- 4. POST PROCESSING
    - Computing engineering parameters for inference
- 5. TESTING/VALIDATION/VERIFICATION

1. Grid Generation

2D heat conduction



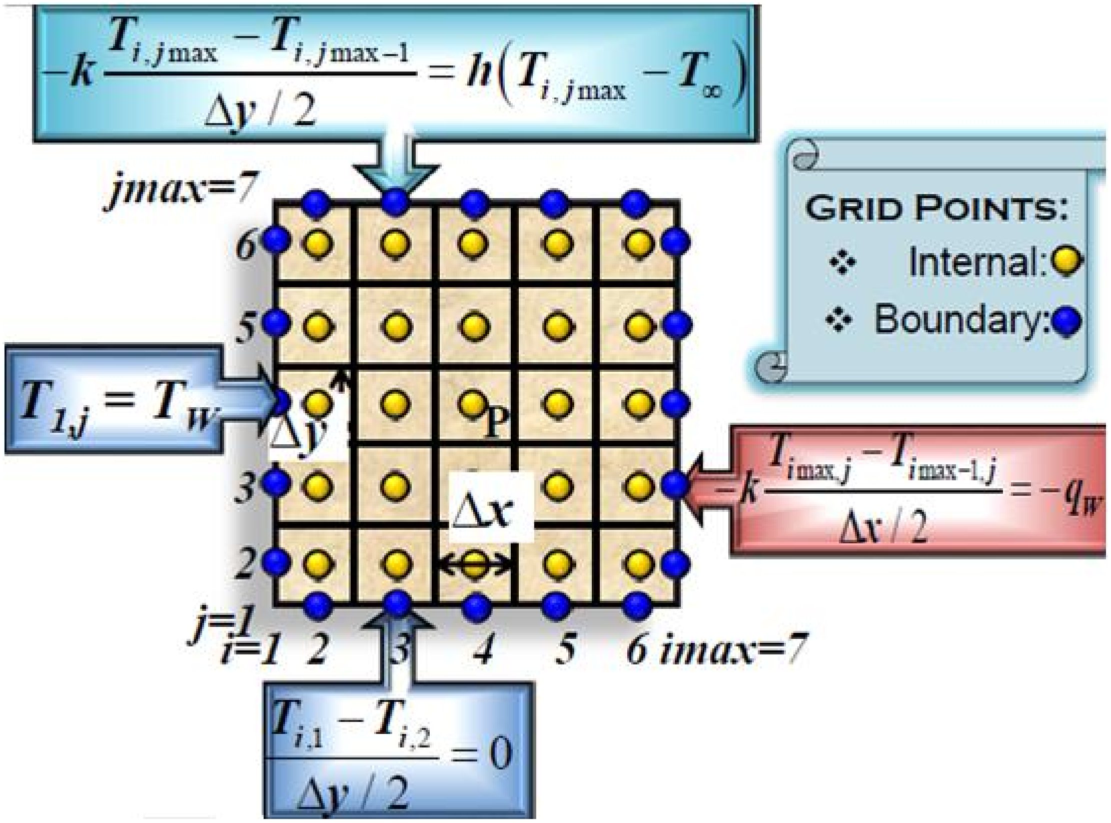
1. Grid Generation



- FDM
  - Grid points
  - GEs are solved based on Taylor series expansion
  - Robust
  - Disdv:
    - No account of flux conservation in formulation

1. Grid Generation

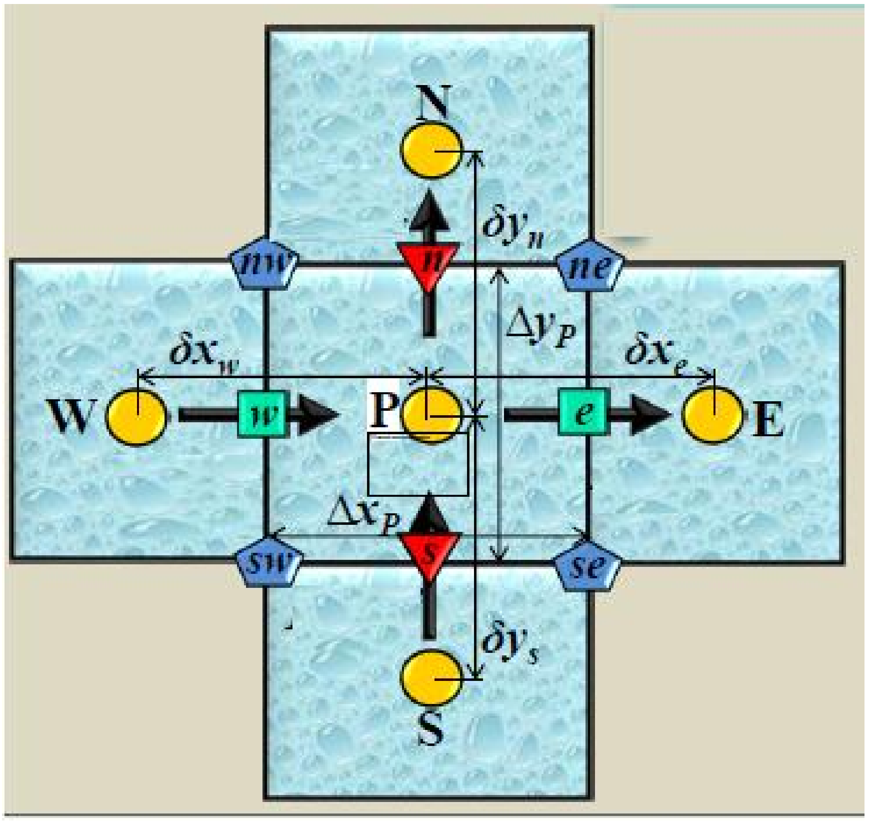
- **FVM**
  - Grid points and Control Volumes (CV)
  - GEs are solved based on Taylor series expansion
  - Physical law based (RTT, Gauss Divergence theorem).
  - Inherently conservative
  - More accurate for Fluid dynamics.
  - More rigorous than FDM



$-k\frac{T_{i,j\max}-T_{i,j\max-1}}{\Delta y/2}=h\left(T_{i,j\max}-T_\infty\right)$
jmax=7
GRID POINTS:
Internal: ●
Boundary: ●
$T_{1,j}=T_W$
Δy
P
Δx
$-k\frac{T_{i\max,j}-T_{i\max-1,j}}{\Delta x/2}=-q_W$
6
5
3
2
j=1
i=1 2 3 4 5 6 imax=7
$\frac{T_{i,1}-T_{i,2}}{\Delta y/2}=0$

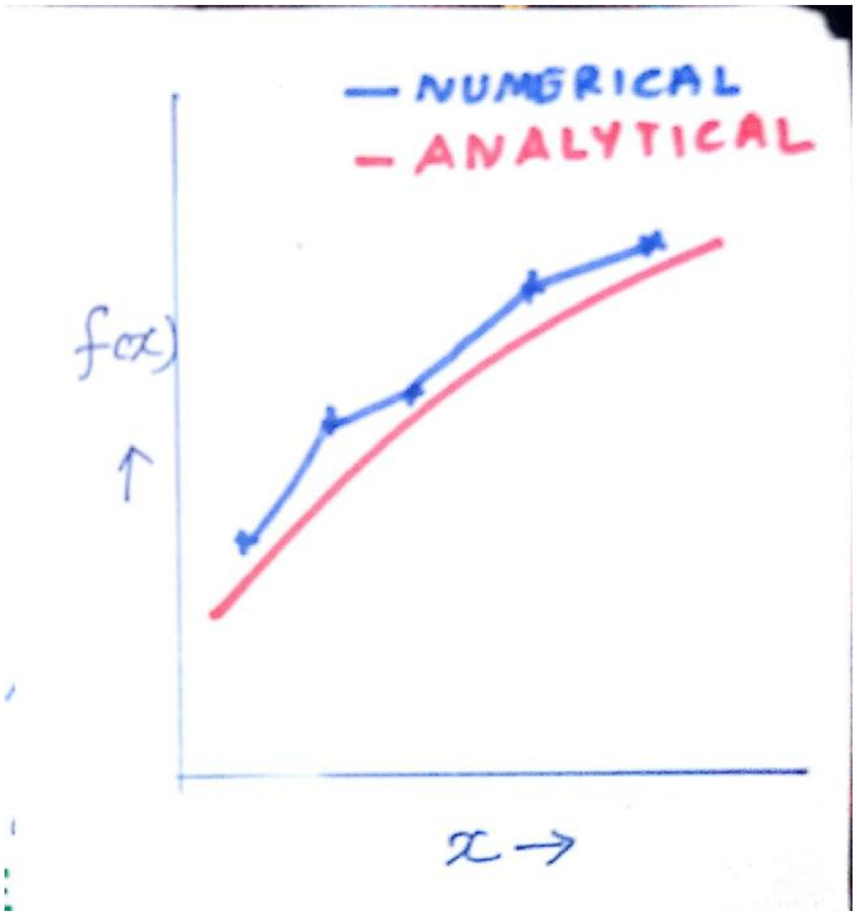
1. Grid Generation

FVM



2. Discretization

- Governing PDEs → LAEs
  - Calculus to algebra
  - PDEs valid everywhere including boundaries
  - Generates closed system of LAEs
  - As LAEs as no. of grid points



2. Discretization

- Discretization methods

1. Taylor Series Expansion (FDM)
2. Variational Method (Weighted Averaging)
3. Method of Weighted Residuals (MWR)
   - FVM, FEM originated from MWR
   - Assumes trial function over entire domain
     - Profile assumption for function

- Higher order polynomials for improved solution
- Tedious:
  - Avoid by subdividing domain. Apply lower order polynomial.

2. Discretization

- Eg. Discretizing 2D heat conduction eqn

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{Q}_{gen}}{k} = 0$$

- Interior grid points

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} + \frac{\dot{Q}_{gen_{i,j}}}{k} = 0$$

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4} + \frac{\left(\dot{Q}_{gen_{i,j}}\right) * (\Delta x)^2}{4k}$$

2. Discretization

- BCs

$$T_{x=0} = T_w \rightarrow T_{1,j} = T_w$$

$$-k\left(\frac{\partial T}{\partial x}\right)_{x=L1} = -q_w \rightarrow T_{imax,j} = T_{imax-1,j} + \frac{q_w \Delta x}{k}$$

$$-k\left(\frac{\partial T}{\partial y}\right)_{y=L2} = h\left(T_{y=L2} - T_\infty\right) \rightarrow T_{i,jmax} = \frac{kT_{i,jmax-1} + h\Delta y T_\infty}{(k + h\Delta y)}$$

$$\left(\frac{\partial T}{\partial y}\right)_{y=0} = 0 \rightarrow T_{i,1} = T_{i,2}$$

Module III & IV

Solution methodology

3. Solution methodology (Solver)

- System of linear algebraic equations (LAEs)

$$a_{11}\emptyset_1 + a_{12}\emptyset_2 + a_{13}\emptyset_3 + \cdots + a_{1n}\emptyset_n = b_1$$

$$a_{21}\emptyset_1 + a_{22}\emptyset_2 + a_{23}\emptyset_3 + \cdots + a_{2n}\emptyset_n = b_2$$

-

-

$$a_{n1}\emptyset_1 + a_{n2}\emptyset_2 + a_{n3}\emptyset_3 + \cdots + a_{nn}\emptyset_n = b_n$$

$$[\boldsymbol{A}][\emptyset] = [\boldsymbol{B}]$$

3. Solution methodology (Solver)

- Direct Method: Algebraic elimination
  - Matrix Inversion (TDMA)
  - Gauss Elimination
  - Used when no of eqns is less (<100), most coeffs are non-zero (system not diagonally dominant), system of eqns is ill conditioned

- Indirect Method: Iterative solution
  - Gauss-Siedel method, Jacobi Method
  - Commonly used in CFD (for steady state form)
  - Convergence criteria: $\left|\Delta \emptyset_{i=1,2,..n}\right|_{max} \leq \epsilon$
  - Large no of eqns, most coefficients are zero (sparse matrix)

3. Solution methodology (Solver)

- Explicit and implicit methods (unsteady formulation)
- Explicit:
  - Use of INFORMATION AT PREVIOUS TIME STEP TO UPDATE value for present time step
  - Proceeds point by point in domain
- Implicit:
  - Use of information from ONE POINT AT PREVIOUS TIME STEP TO UPDATE ALL VALUES for present time step
  - Solves for all points in the domain simultaneously

3. Solution methodology (Solver)

- 1D heat transient conduction eqn.

$$\frac{\partial T}{\partial t} + \alpha \frac{\partial^2 T}{\partial x^2} = 0$$

- Explicit:

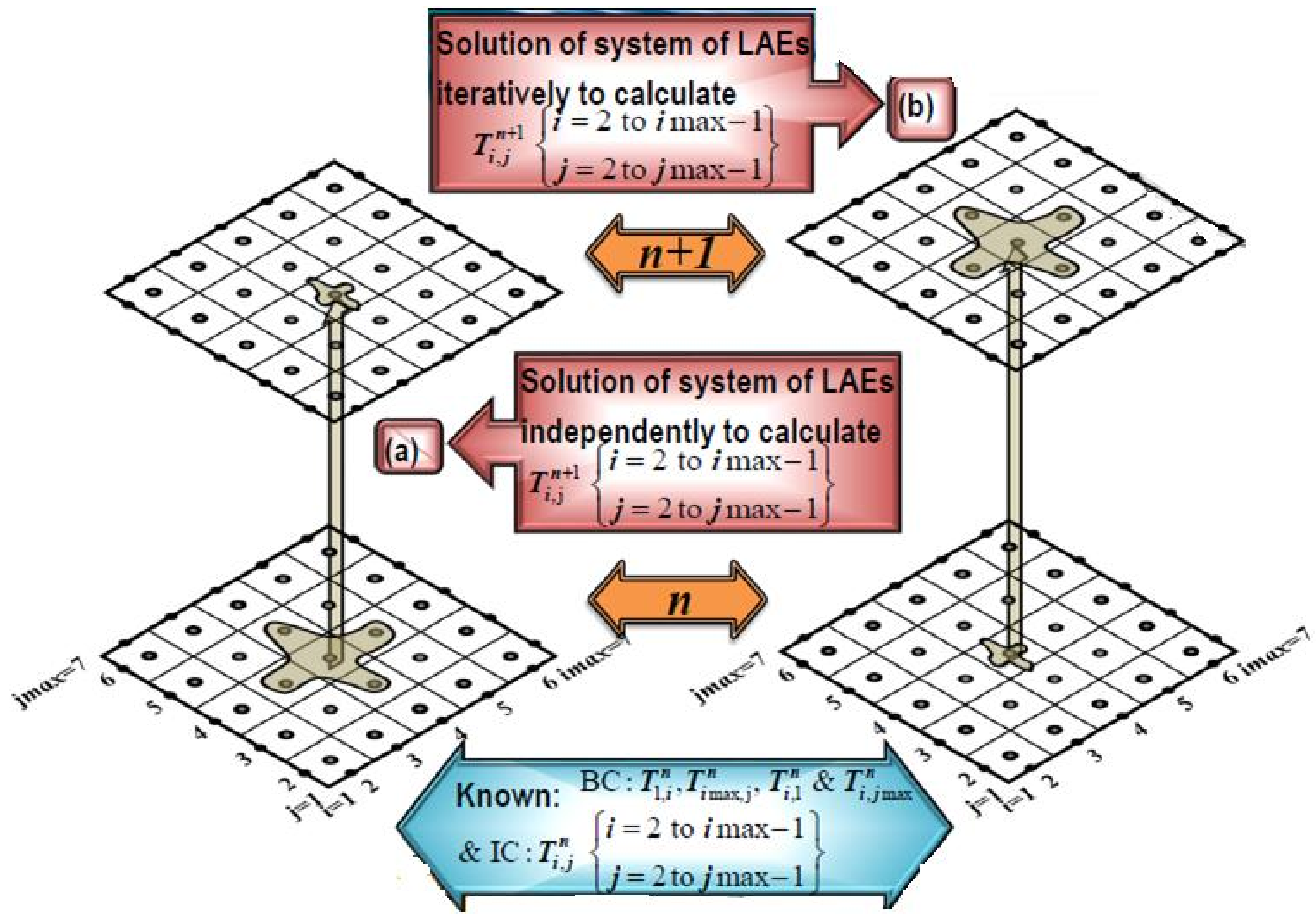
$$\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t} = \alpha \left(\frac{T_{i+1,j}{}^{n} - 2T_{i,j}{}^{n} + T_{i-1,j}{}^{n}}{(\Delta x)^2} \right) + O(\Delta t, (\Delta x)^2)$$

  – FDCS

- Implicit:

$$\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t} - \alpha \left(\frac{T_{i+1,j}{}^{n+1} - 2T_{i,j}{}^{n+1} + T_{i-1,j}{}^{n+1}}{(\Delta x)^2} \right) + O(\Delta t, (\Delta x)^2)$$

3. Solution methodology (Solver)



Solution of system of LAEs iteratively to calculate $T_{i,j}^{n+1} \begin{Bmatrix} i = 2 \text{ to } i\max-1 \\ j = 2 \text{ to } j\max-1 \end{Bmatrix}$
(b)
$n+1$
Solution of system of LAEs independently to calculate $T_{i,j}^{n+1} \begin{Bmatrix} i = 2 \text{ to } i\max-1 \\ j = 2 \text{ to } j\max-1 \end{Bmatrix}$
(a)
n
jmax=7
6 imax=7
j=1
i=1
Known: BC : $T_{1,i}^n, T_{i\max,j}^n, T_{i,1}^n$ & $T_{i,j\max}^n$
& IC : $T_{i,j}^n \begin{Bmatrix} i = 2 \text{ to } i\max-1 \\ j = 2 \text{ to } j\max-1 \end{Bmatrix}$

3. Solution methodology (Solver)

- Explicit

$$T_{i,j}{}^{n+1} - \left(1 - \frac{2\alpha\Delta t}{(\Delta x)^2}\right) T_{i,j}{}^{n} + \left(\frac{\alpha\Delta t}{(\Delta x)^2}\right) T_{i+1,j}{}^{n} + \left(\frac{\alpha\Delta t}{(\Delta x)^2}\right) T_{i-1,j}{}^{n}$$

  - Possibility for coefficient term to be negative
  - Causes “Instability”→ Solution “diverges”
  - Leads to Stability Criteria (“Conditional Stability”)
    - GRID FOURIER CRITERIA

$$\left(1 - \frac{2\alpha\Delta t}{(\Delta x)^2}\right) \geq 0 \rightarrow \Delta t \leq \frac{(\Delta x)^2}{2\alpha}$$

3. Solution methodology (Solver)

- Implicit

$$\left(1+\frac{2\alpha\Delta t}{(\Delta x)^2}\right)T_{i,j}{}^{n+1} = T_{i,j}{}^{n} + \left(\frac{\alpha\Delta t}{(\Delta x)^2}\right)T_{i+1,j}{}^{n+1} + \left(\frac{\alpha\Delta t}{(\Delta x)^2}\right)T_{i-1,j}{}^{n+1}$$

  – Inherently stable
  – Computationally costly (iterative method)

- Alternate scheme
  – Crank Nicolson (semi-implicit scheme)
    - Combines implicit and explicit schemes
    - Also an iterative, stable method, no stability criteria

3. Solution methodology (Solver)

- Measure of unsteadiness in iterations
  - Sets “Convergence Criteria”
  - Gives “residual plot” for convergence
  - Stoppage criteria for iterations

$$\left|\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t}\right|_{max} \leq \epsilon \sim 0.0001$$

    - Can cause false convergence→ Non-dimensionalize based on Fourier number

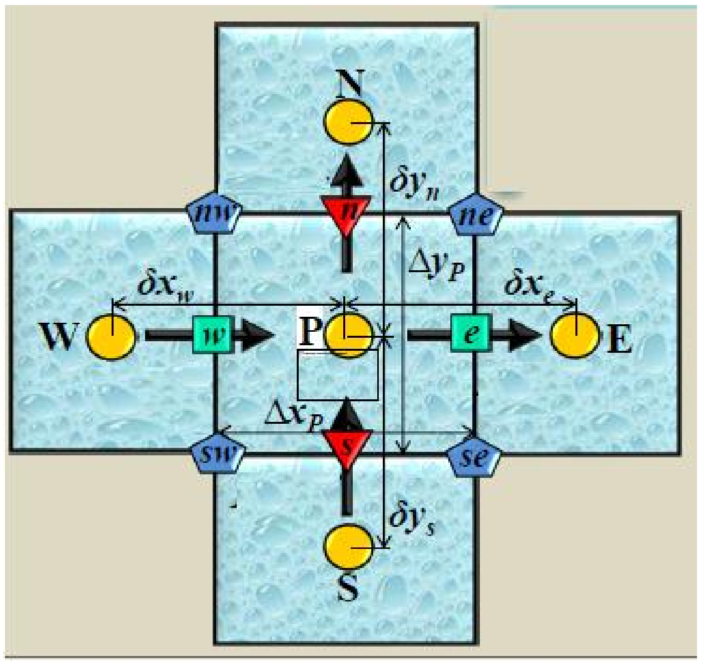
$$\left|\left(\frac{l_c{}^2}{\alpha \Delta T_c}\right)\left(\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t}\right)\right|_{max} \leq \epsilon \sim 0.0001$$

3. Solution methodology (Solver)

- Errors:
  - Consistency:
    - If FDE→PDE, for Δx, Δt →0, the scheme is consistent
  - Discretization error:
    - Truncation error + other errors of numerical scheme
  - Round off error:
    - Computer's tendency to round off few decimals
  - Convergence:
    - Stability + Consistency → convergence for FDM (Lax theorem). Eg. Suitable for heat conduction eqn, NOT for NS equations.

3. Solution methodology (Solver)

- 2D heat advection
  - Advection: bulk motion of fluid
  - Involves calculation of properties at face centres
  - Advection Schemes:
    - Based on profile assumptions
      - FOU
      - CD
      - SOU
      - QUICK



3. Solution methodology (Solver)

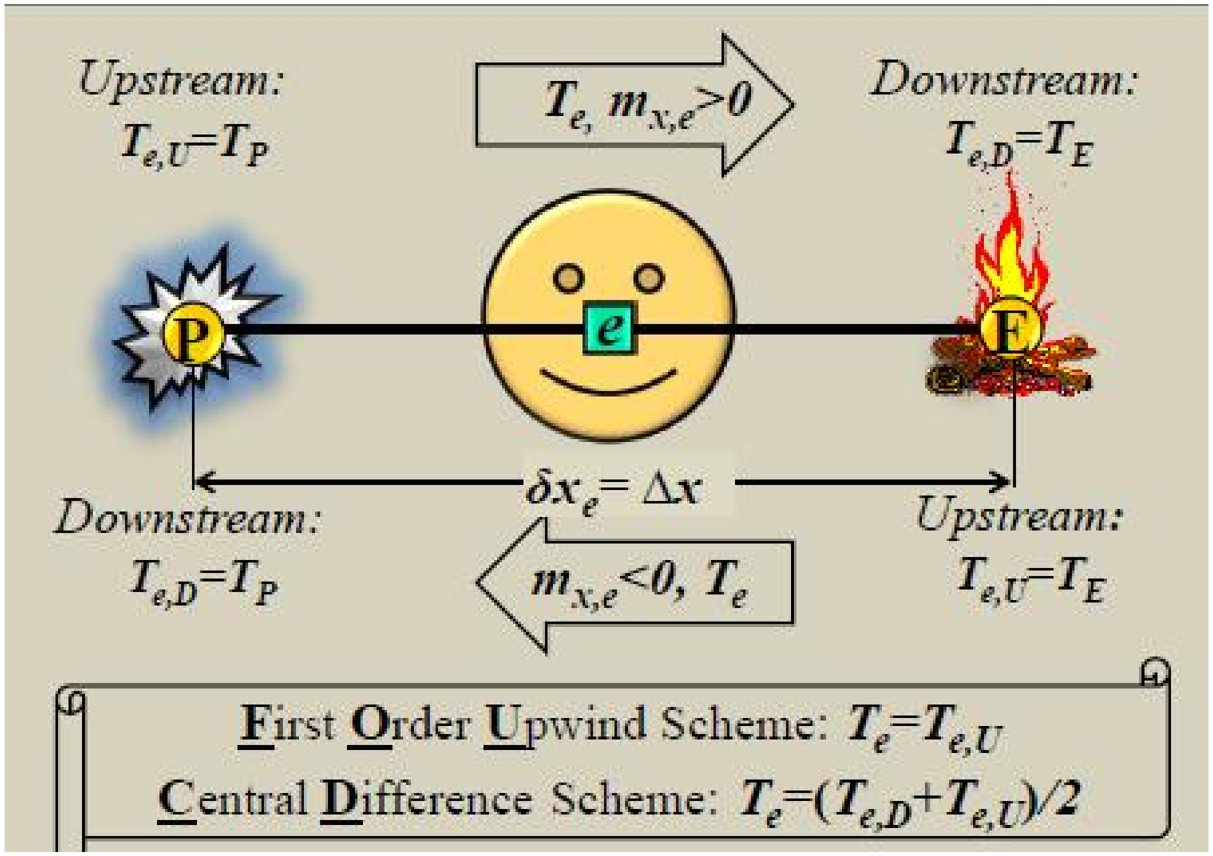
- Advection schemes
    - Based on profile assumptions for properties at cell faces:
  - 1. FOU (First Order Upwind)
  - 2. CD (Central Difference)
  - 3. SOU (Second Order Upwind)
  - 4. QUICK (Quadratic Upwind Interpolation for Convective Kinematics)
  - 5. POWER LAW

Industry preferred schemes

Academicians interest

3. Solution methodology (Solver)

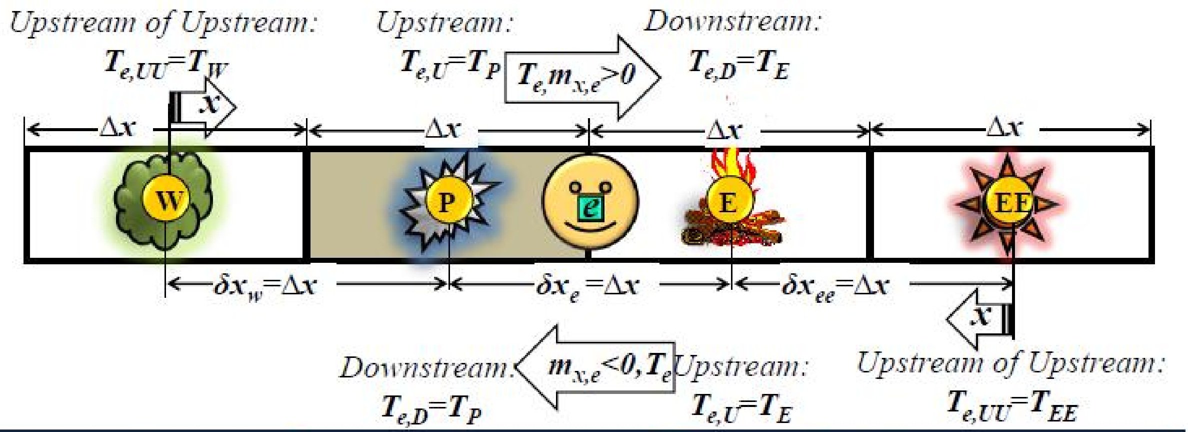
- Advection schemes: Profile assumption

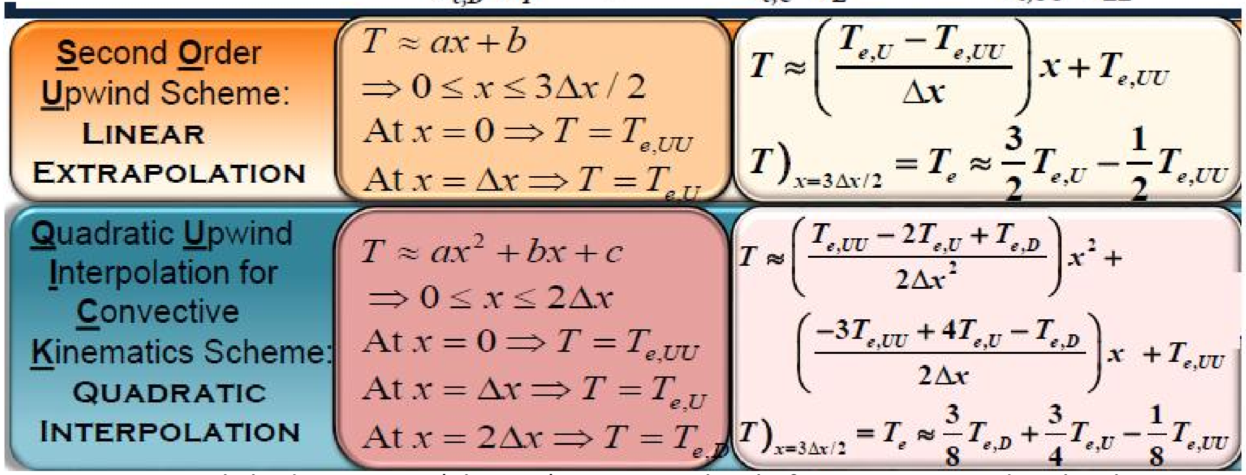


Upstream: $T_{e,U}=T_P$
T_e, $m_{x,e}>0$
Downstream: $T_{e,D}=T_E$
P
e
E
$\delta x_e=\Delta x$
Downstream: $T_{e,D}=T_P$
$m_{x,e}<0$, T_e
Upstream: $T_{e,U}=T_E$
First Order Upwind Scheme: $T_e=T_{e,U}$
Central Difference Scheme: $T_e=(T_{e,D}+T_{e,U})/2$

3. Solution methodology (Solver)

- Advection schemes: Profile assumption

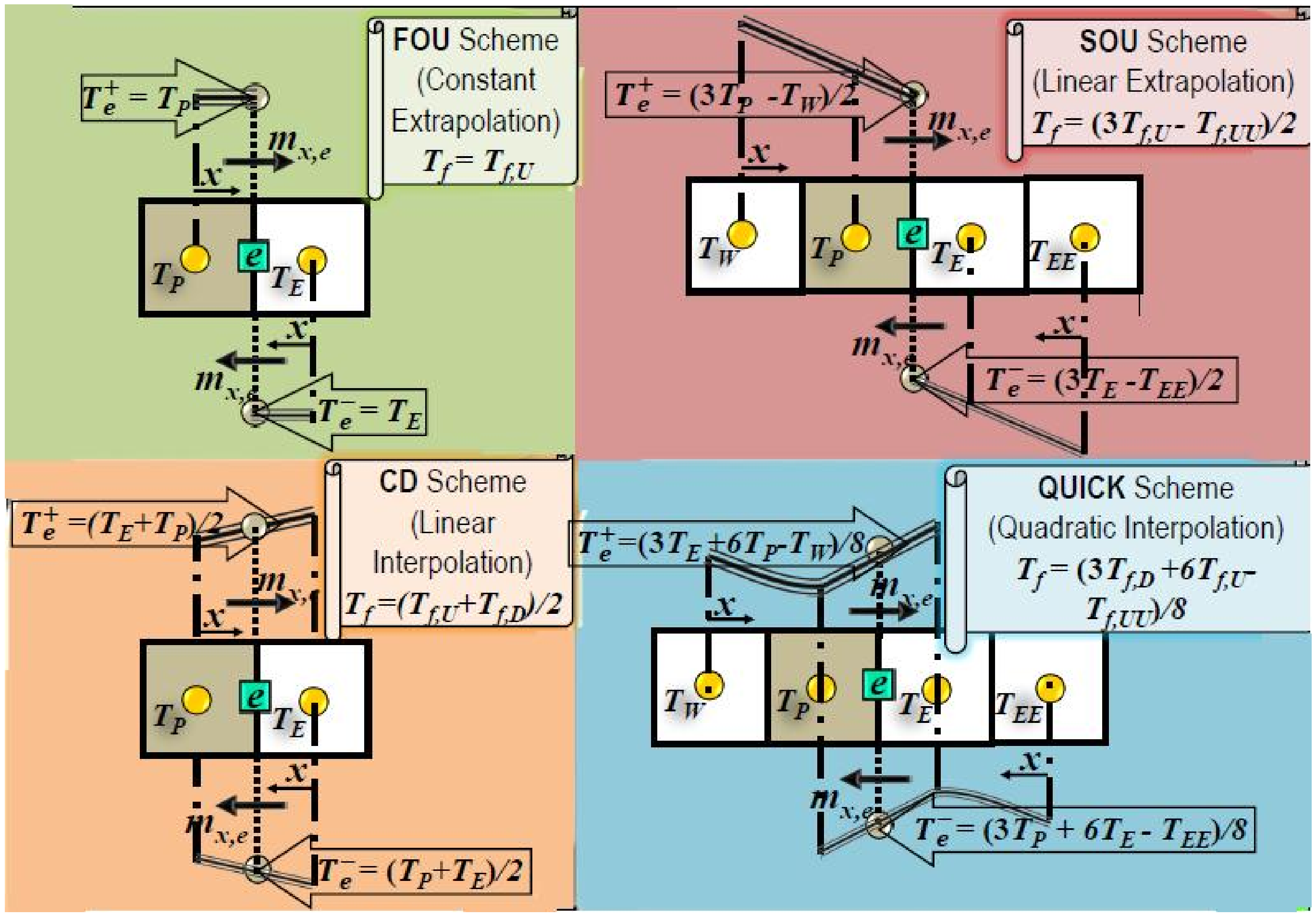




Second Order Upwind Scheme: LINEAR EXTRAPOLATION	$T \approx ax+b$ $\Rightarrow 0 \le x \le 3\Delta x/2$ At $x=0 \Rightarrow T=T_{e,UU}$ At $x=\Delta x \Rightarrow T=T_{e,U}$	$T \approx \left(\dfrac{T_{e,U}-T_{e,UU}}{\Delta x}\right)x+T_{e,UU}$ $T)_{x=3\Delta x/2}=T_e \approx \dfrac{3}{2}T_{e,U}-\dfrac{1}{2}T_{e,UU}$
Quadratic Upwind Interpolation for Convective Kinematics Scheme: QUADRATIC INTERPOLATION	$T \approx ax^2+bx+c$ $\Rightarrow 0 \le x \le 2\Delta x$ At $x=0 \Rightarrow T=T_{e,UU}$ At $x=\Delta x \Rightarrow T=T_{e,U}$ At $x=2\Delta x \Rightarrow T=T_{e,D}$	$T \approx \left(\dfrac{T_{e,UU}-2T_{e,U}+T_{e,D}}{2\Delta x^2}\right)x^2+$ $\left(\dfrac{-3T_{e,UU}+4T_{e,U}-T_{e,D}}{2\Delta x}\right)x+T_{e,UU}$ $T)_{x=3\Delta x/2}=T_e \approx \dfrac{3}{8}T_{e,D}+\dfrac{3}{4}T_{e,U}-\dfrac{1}{8}T_{e,UU}$

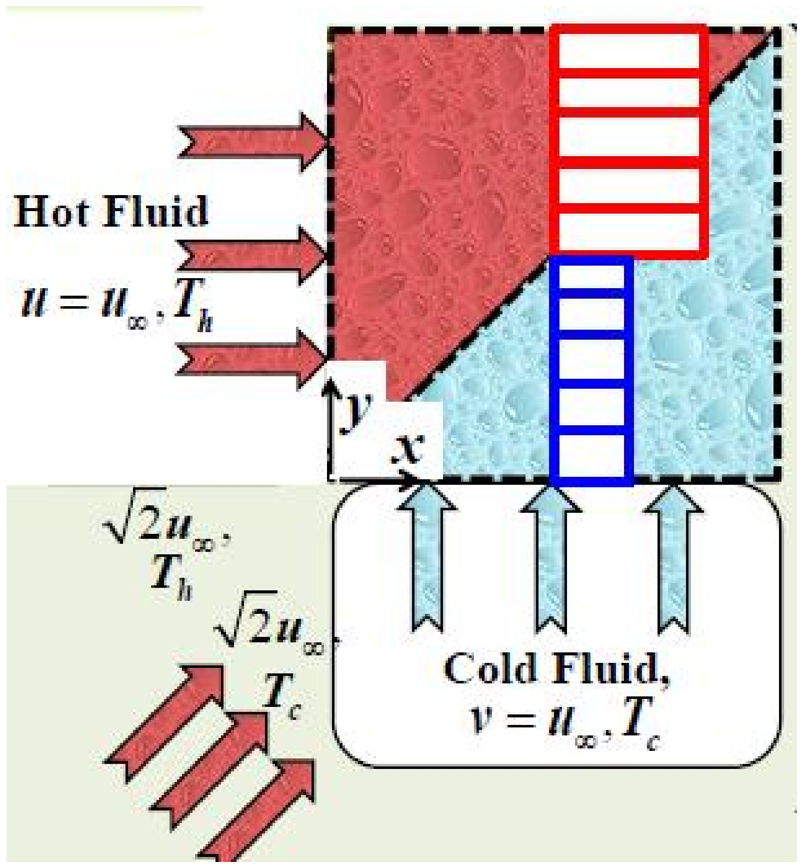
3. Solution methodology (Solver)

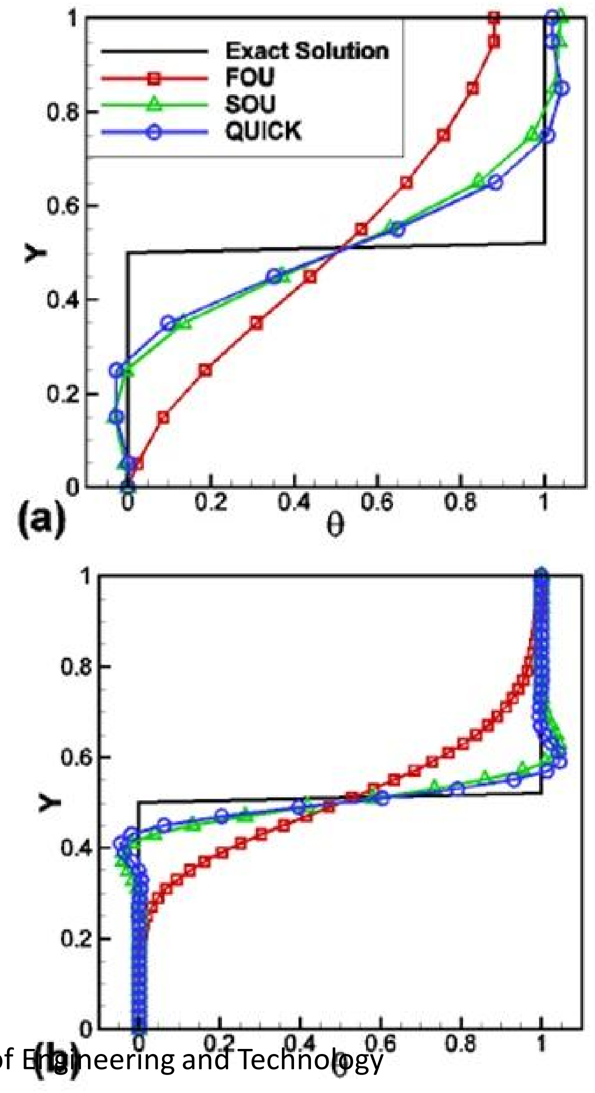
- Advection schemes: Summary



3. Solution methodology (Solver)

Advection schemes: Comparison





3. Solution methodology (Solver)

- 2D heat advection eqn

$$\frac{\partial(\rho C_p T)}{\partial t} + V \cdot \nabla(\rho C_p T) = 0$$

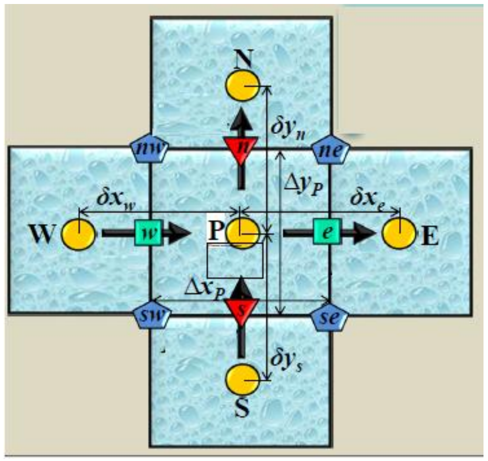
- FVM based discretization:

$$\rho C_p \frac{T_P^{n+1} - T_P^n}{\Delta t} \Delta x_P \Delta y_P + Q_{adv}^{\chi} = 0$$

where

$$Q_{adv}^{\chi} = (h_{x,e}^{\chi} - h_{x,w}^{\chi})\Delta y_P + (h_{y,n}^{\chi} - y_{y,s}^{\chi})\Delta x_P$$

$\chi = n \, or \, n+1 \, depending \, the \, scheme$



3. Solution methodology (Solver)

- FVM based discretized eqn

$$\rho C_p \frac{T_P^{n+1} - T_P^n}{\Delta t} \Delta x_P \Delta y_P + \lambda Q_{adv,P}^{n+1} + (1-\lambda) Q_{adv,P}^n = 0$$

$$\lambda = \begin{cases} 0, & Explicit \\ 1, & Implicit \\ 0.5, & Crank - N \end{cases}$$

- If following implicit or C-N
  - Avoid iterative methods unless using FOU scheme
    - Most coeffs may not be diagonally dominant

3. Solution methodology (Solver)

- Measure of unsteadiness in iterations (advection)
  - Stoppage criteria for iterations

$$\left|\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t}\right|_{max} \leq \epsilon \sim 0.0001$$

  - Can cause false convergence→ Non-dimensionalize based on CFL (Courant Frederich Lewis) number

$$\left|\left(\frac{l_c}{u_c \Delta T_c}\right)\left(\frac{T_{i,j}{}^{n+1} - T_{i,j}{}^{n}}{\Delta t}\right)\right|_{max} \leq \epsilon \sim 0.0001$$

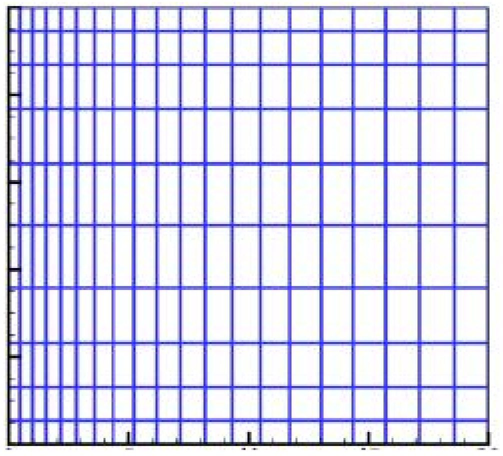
- Convergence Criteria for stability
  - CFL Criteria $\frac{u\Delta t}{\Delta x} \leq 1$

Module V &VI

Introduction to Grids, Pressure-velocity coupling, Algorithms and CFD packages

Overview on Grid types

- Structured
    - Neighbouring grids connected in similar pattern
  - Structured uniform
  - Structured non-uniform



Overview on Grid types

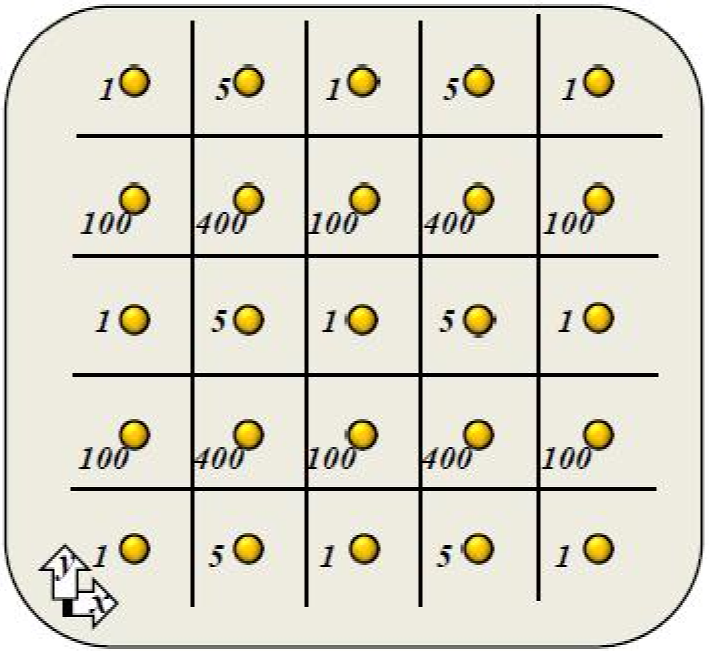
- Co-located grid
  - Coinciding temperature, velocity and pressure grid-points
    - No book keeping, Lesser storage, Can be easily extended to unstructured grid
    - However, with NS eqn, could result in false results
      - Checker-board field for velocity and pressure
      - Issue with pressure gradient term

$$\int_w^e \frac{\partial P}{\partial x} dx - P_w - P_e$$

$$P_e = \frac{1}{2}(P_E + P_P) \qquad P_w = \frac{1}{2}(P_P + P_W) \qquad P_w - P_e = \frac{1}{2}(P_W - P_E)$$

Issue with co-located grids

- Checker-board velocity-pressure field in a co-located grid → "Pressure-velocity decoupling"
  - (pressure and velocity evaluated at same locations in the grid)
  - Wavy field



Overview on Grid types

- Co-located grid
  - Solution to checker-board problem
    - Either use “staggered grid” OR
    - Add the pressure to source term in NS eqn
      - Control the iterations (“Convergence”) through a factor α
        - “UNDER RELAXATION” factor
        - “SUCCESSIVE OVER RELAXATION” factor SOR (in Gauss-Siedel)

$$a_p \emptyset_p = \sum(a_{nb}\emptyset_{nb}) + b$$

$$\emptyset_p = \frac{\sum(a_{nb}\emptyset_{nb}) + b}{a_p} - \emptyset_p + \emptyset_p$$

$$\emptyset_p{}^{n+1} = \emptyset_p{}^{n} + \alpha\left(\frac{\sum(a_{nb}\emptyset_{nb}) + b}{a_p} - \emptyset_p\right)^{n+1}$$

Overview on Grid types

- "UNDER/OVER RELAXATION" factor
  - Controls the convergence rate

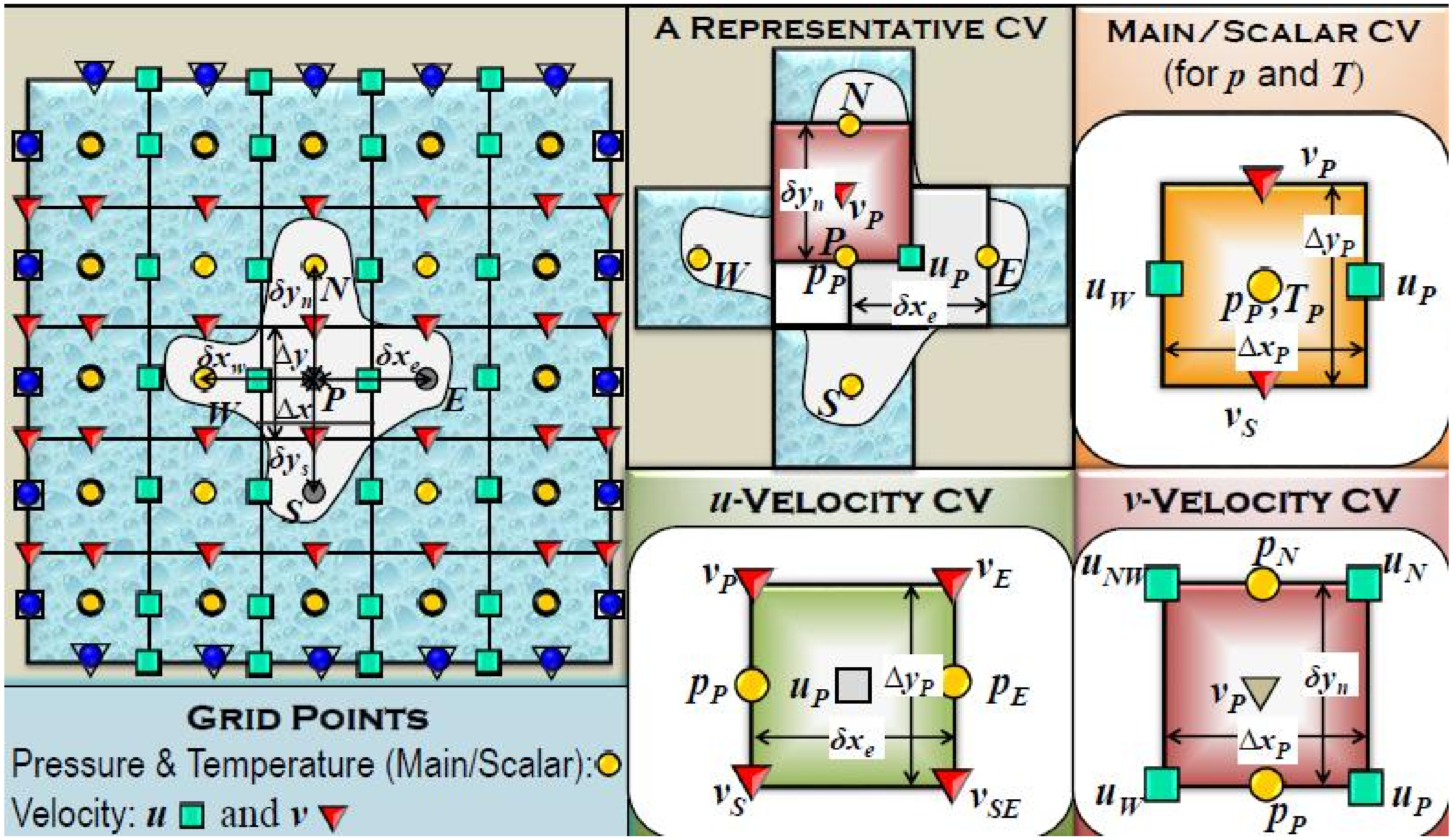
$$\alpha = \begin{cases} < 1, & NS\ equations \\ > 1, & Diffussion\ eqns \end{cases}$$

Overview on Grid types

- Staggered grid
  - Avoids Pressure-velocity decoupling
  - Separate control volumes for momentum flux, located at half CV distance from main CV
    - Pressure drives flow
    - No interpolation involved
    - No wavy fields
  - Solution to checker-board problem (false convergence) in regular grids
  - Computationally costly
    - high book keeping

Overview on Grid types

- Staggered grid



Overview on Grid types

- Staggered grid
  - LAE of the form

$$a_e u_e = \sum(a_{nb} u_{nb}) + (P_P - P_E)A_E + b$$

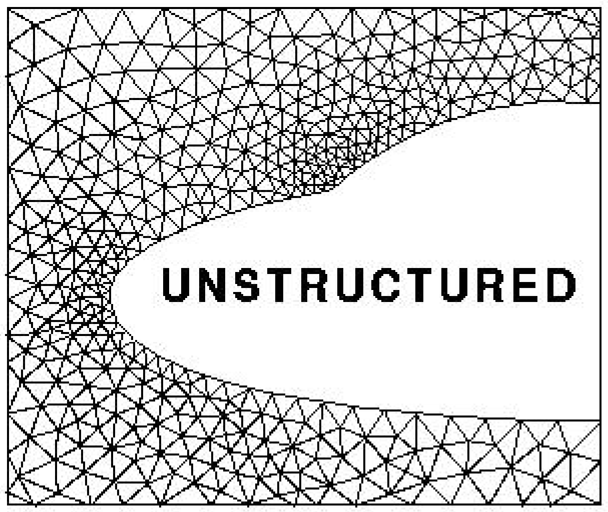
  - A "predictor-corrector algorithm" (Semi-implicit) is used to iteratively arrive at correct flow field
    - SIMPLE (Semi Implicit Method for Pressure Linked Eqns)
    - Assures fast convergence
    - More computational time
      - Additional eqns. and book keeping for correction values

Overview on Grid types

- Staggered grid
  - Improves SIMPLE
    - Corrected pressure term includes pseudo velocity
    - Better convergence
    - Less computational time than SIMPLE
      - However in FLUENT, SIMPLE preferred than SIMPLER since pseudo velocity field is very complex to handle.

Overview on Grid types

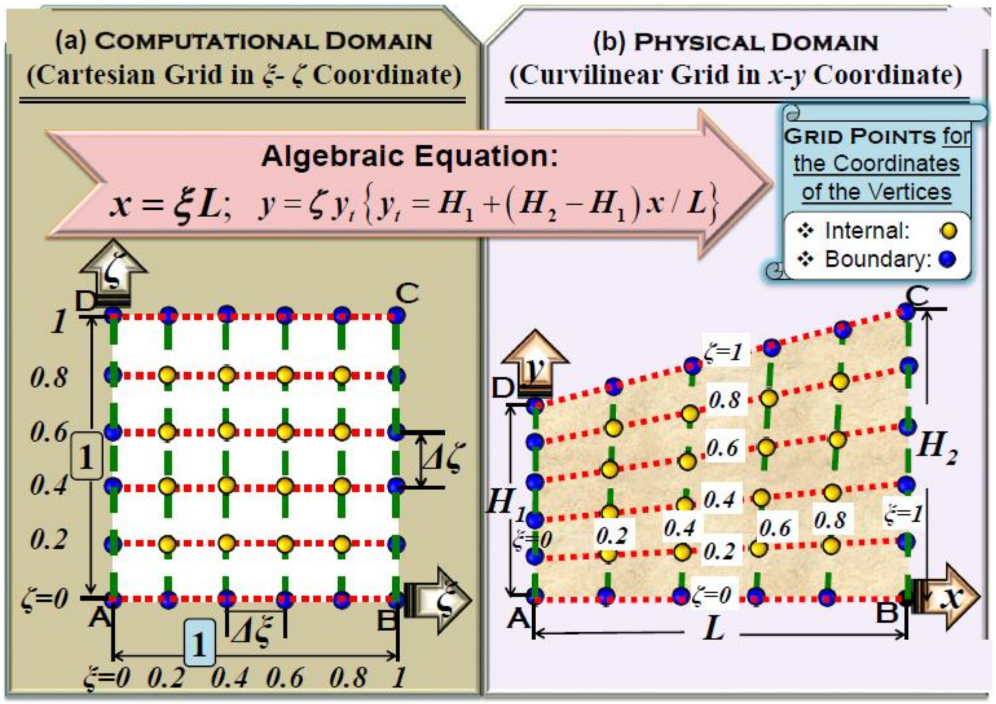
- Unstructured grid
  - Non-uniform
  - Delaunay triangulation algorithm for grid generation
  - Better approximation of irregular boundaries
    - Advantage of FVM over FDM
    - Complex geometries



Overview on Grid types

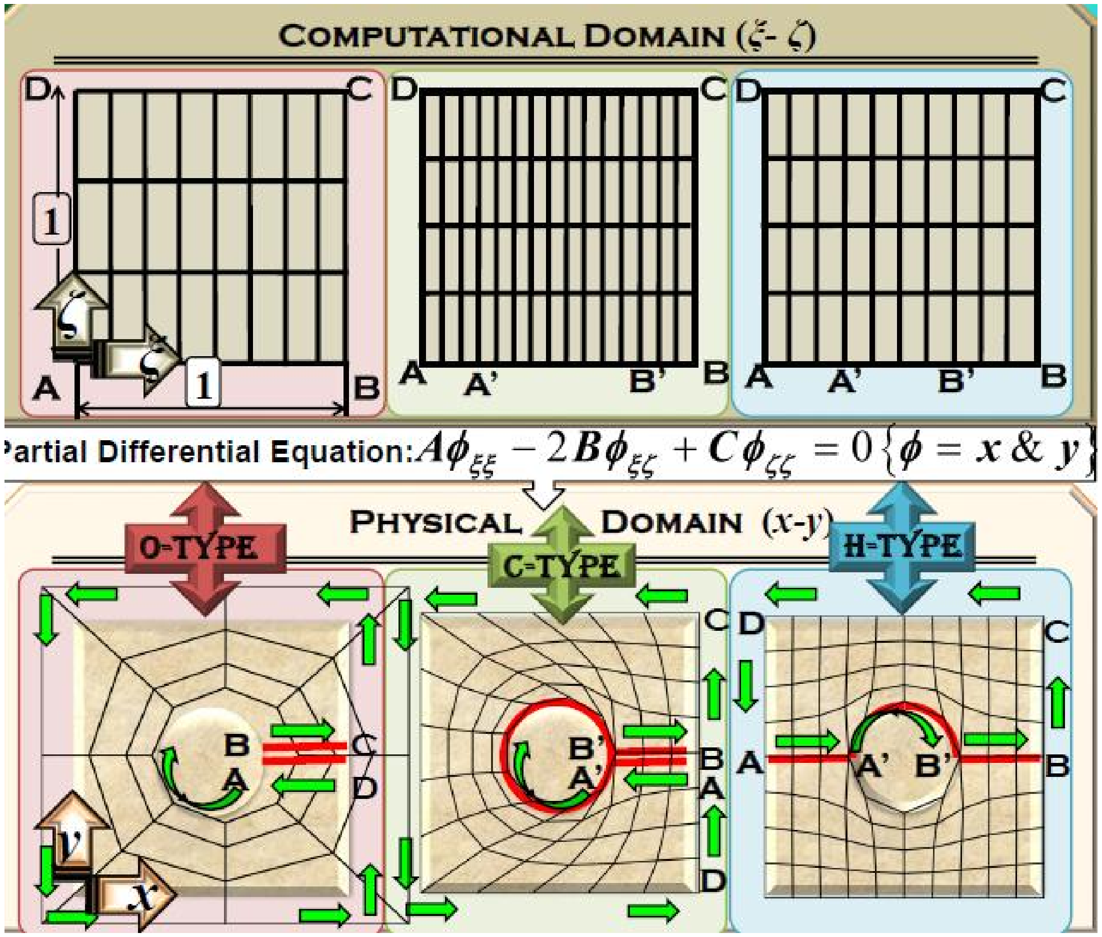
Curvilinear Grids

- 1. Algebraic Grid Generation
  - Coordinate transformation from "Computational domain" to "Physical domain" through algebraic eqns

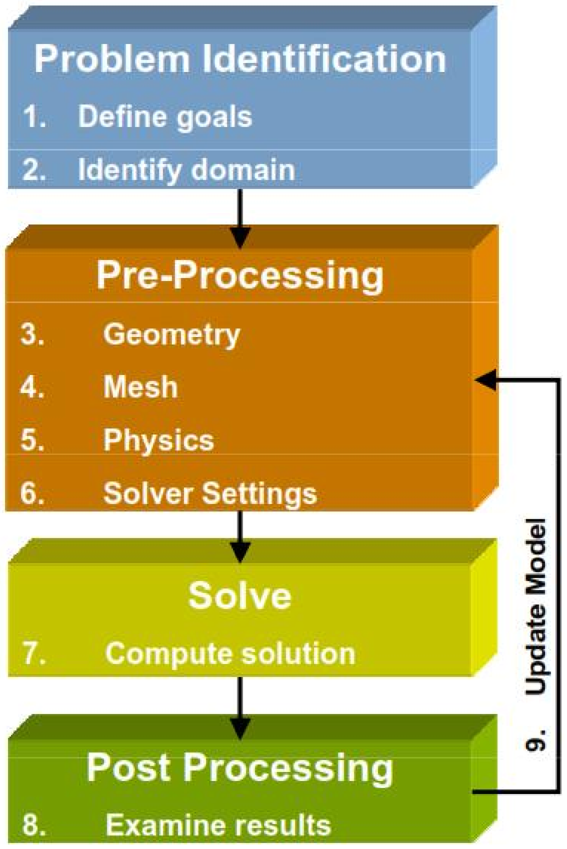

(a) Computational Domain
(Cartesian Grid in ξ- ζ Coordinate)
(b) Physical Domain
(Curvilinear Grid in x-y Coordinate)
Algebraic Equation:
$x = \xi L;\quad y = \zeta y_t \{y_t = H_1 + (H_2 - H_1)x/L\}$
Grid Points for the Coordinates of the Vertices
Internal: ●
Boundary: ●
A
B
C
D
ζ=0
0.2
0.4
0.6
0.8
1
ξ=0
Δξ
Δζ
ζ=1
H_1
H_2
L
x
y
ξ
ζ

Overview on Grid types

- Curvilinear Grids
  - 2. Elliptic Grid Generation
    - Based on PDE
- **‘O’ TYPE, ‘C’ TYPE, ‘H’ TYPE**


COMPUTATIONAL DOMAIN (ξ- ζ)
D
C
A
B
1
1
ζ
ξ
D
C
A
A'
B'
B
D
C
A
A'
B'
B
Partial Differential Equation: $A\phi_{\xi\xi} - 2B\phi_{\xi\zeta} + C\phi_{\zeta\zeta} = 0\{\phi = x \,\&\, y\}$
PHYSICAL DOMAIN (x-y)
O-TYPE
C-TYPE
H-TYPE
B
A
C
D
y
x
B'
A'
C
B
A
D
D
C
A
A'
B'
B

CFD Development in ANSYS



Boundary Conditions in ANSYS Fluent

- **Velocity inlet :**
  - used to define the velocity and scalar properties of the flow at inlet boundaries.
- **Pressure inlet:**
  - To define the total pressure and other scalar quantities at flow inlets.
- **Mass flow inlet**
  - used in compressible flows to prescribe a mass flow rate at an inlet.
  - It is not necessary to use mass flow inlets in incompressible flows because when density is constant, velocity inlet boundary conditions will fix the mass flow. Like pressure and velocity inlets, other inlet scalars are also prescribed.
- **Pressure outlet**
  - To define the static pressure at flow outlets (and also other scalar variables, in case of backflow). The use of a pressure outlet boundary condition instead of an outflow condition often results in a better rate of convergence when backflow occurs during iteration.
- **Pressure far-field**
  - To model a free-stream compressible flow at infinity, with free-stream Mach number and static conditions specified. This boundary type is available only for compressible flows.

Boundary Conditions in ANSYS Fluent

- **Outflow boundary**
  - used to model flow exits where the details of the flow velocity and pressure are not known prior to solution of the flow problem. They are appropriate where the exit flow is close to a fully developed condition, as the outflow boundary condition assumes a zero stream-wise gradient for all flow variables except pressure. They are not appropriate for compressible flow calculations.
- **Inlet vent**
  - To model an inlet vent with a specified loss coefficient, flow direction, and ambient (inlet) total pressure and temperature.
- **Intake fan**
  - Used to model an external intake fan with a specified pressure jump, flow direction, and ambient (intake) total pressure and temperature.
- **Outlet vent**
  - To model an outlet vent with a specified loss coefficient and ambient (discharge) static pressure and temperature.
- **Exhaust fan**
  - Used to model an external exhaust fan with a specified pressure jump and ambient (discharge) static pressure.

Thank You !!