MEE4006- Computational Fluid Dynamics(CFD)

Dr. N.Sekarapandian

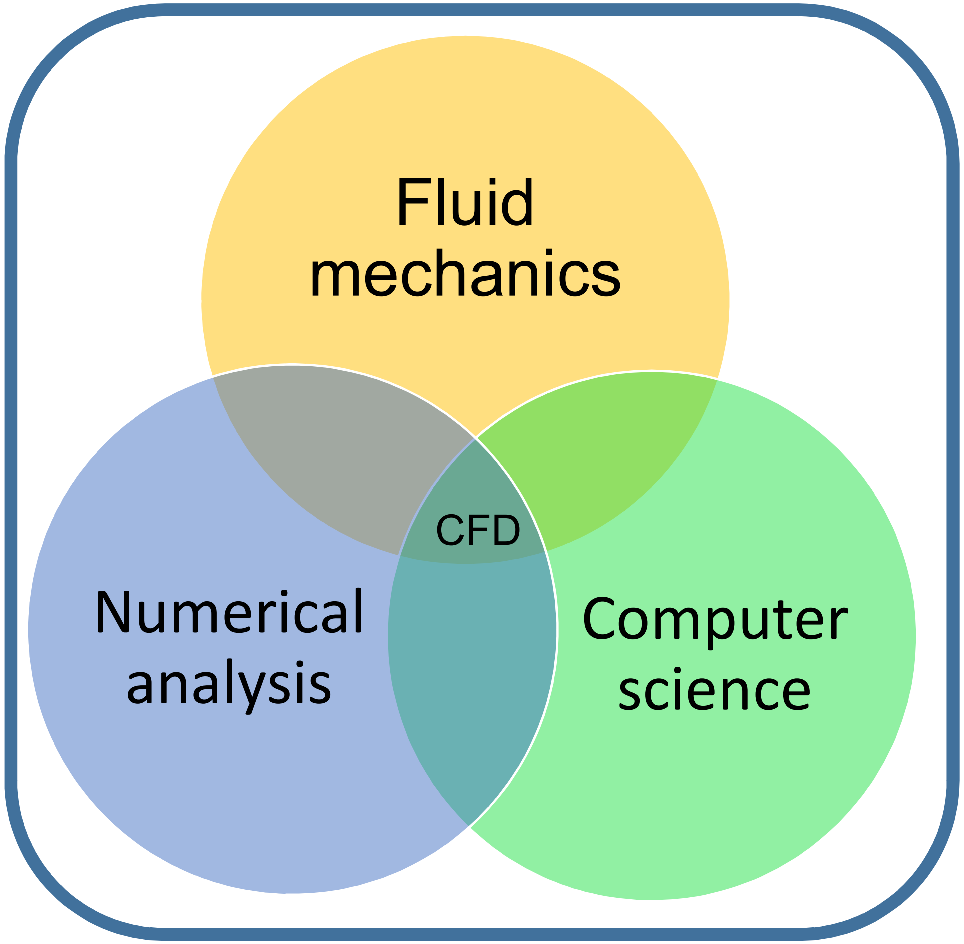
Associate Professor,

School of Mechanical Engineering (SMEC),

VIT University,

Vellore-632014, Tamilnadu, India

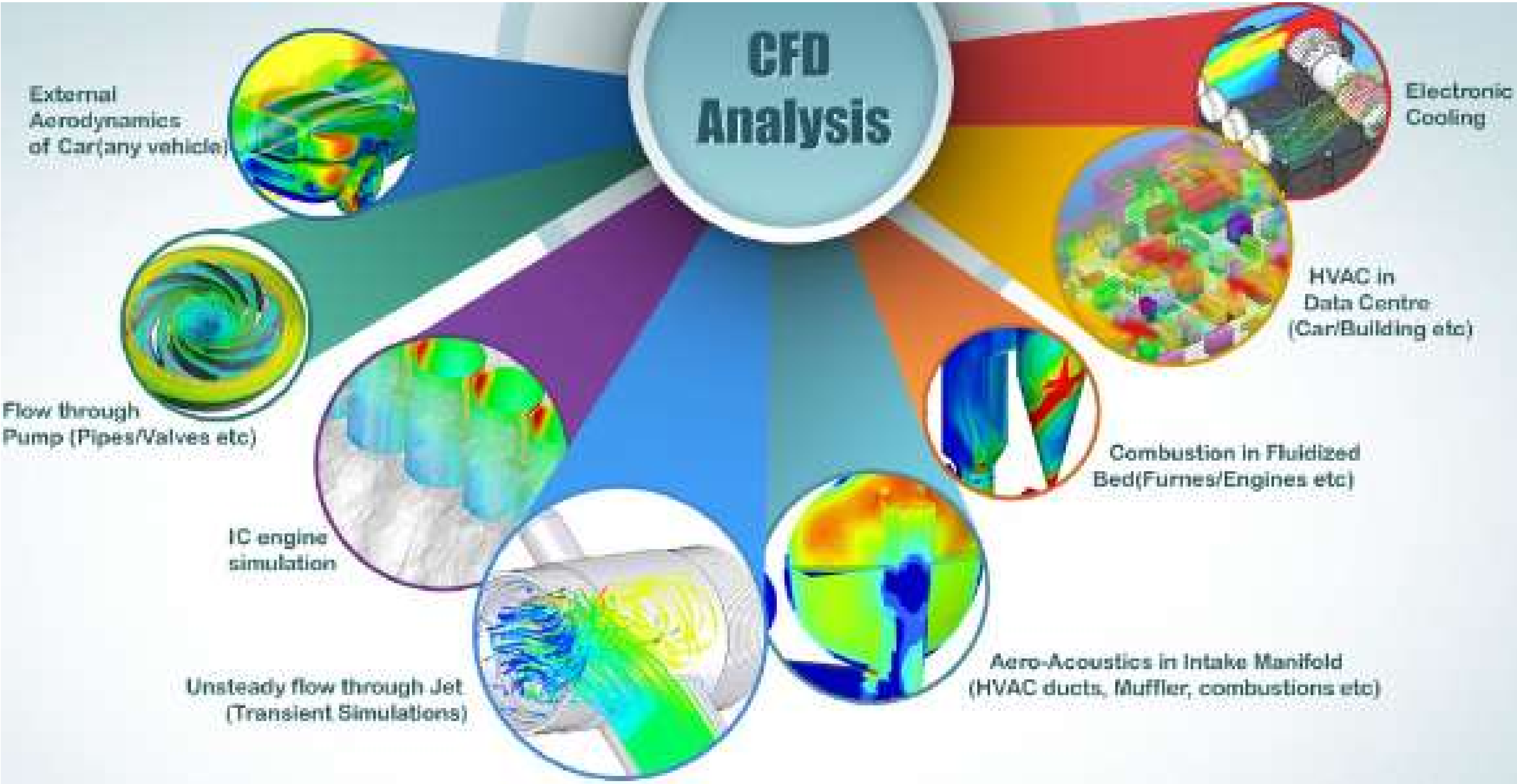
CFD Overview



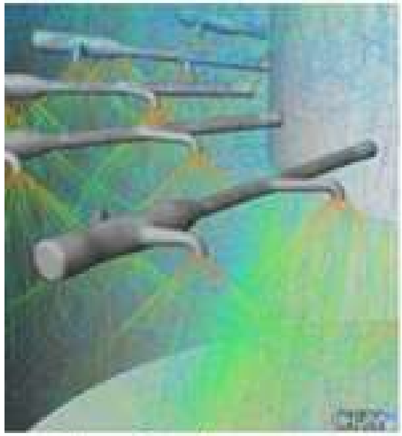
- Lots of university offer courses on CFD and it is an active area of research
- Number of software packages available (e.g. Ansys Fluent)
- Vast literature available on numerical methods for fluid mechanics.
- Widely accepted as a design tool by industrial users
- Even with incompressible flow – impossible to cover everything in single work.
- Based on the speed, the fluid flow is broadly classified into creeping, laminar and turbulent flows.
- Based on the Mach number, fluid flow can be classified into incompressible and compressible flows.
- Type of flow affects the mathematical nature of the problem and therefore the solution method.

CFD APPLICATIONS

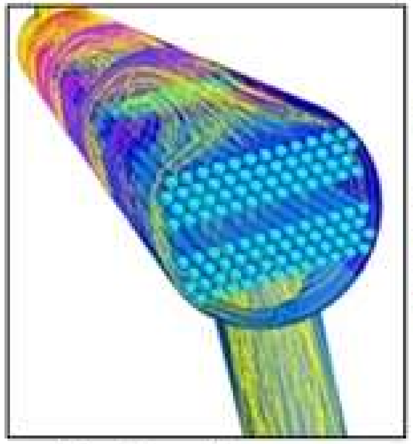
Wide Spectrum of Applications



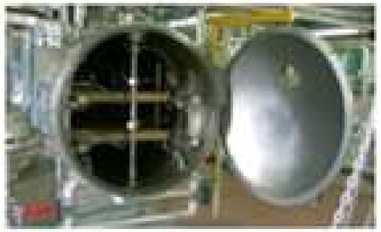
Materials & Chemical Processing



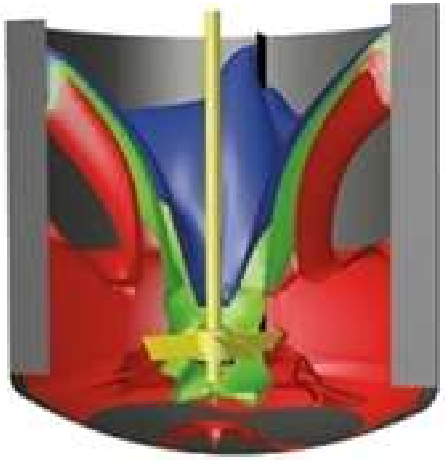
Chemical sprays



Heat exchangers

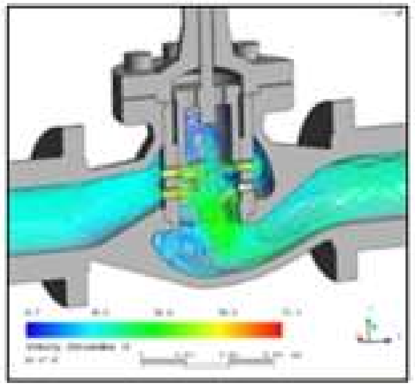


Dryers

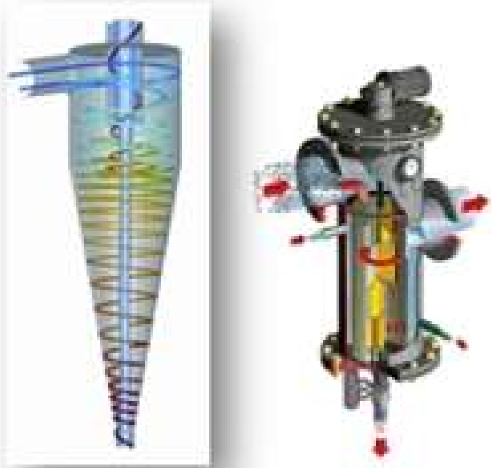


Mixing tanks

Metal processing

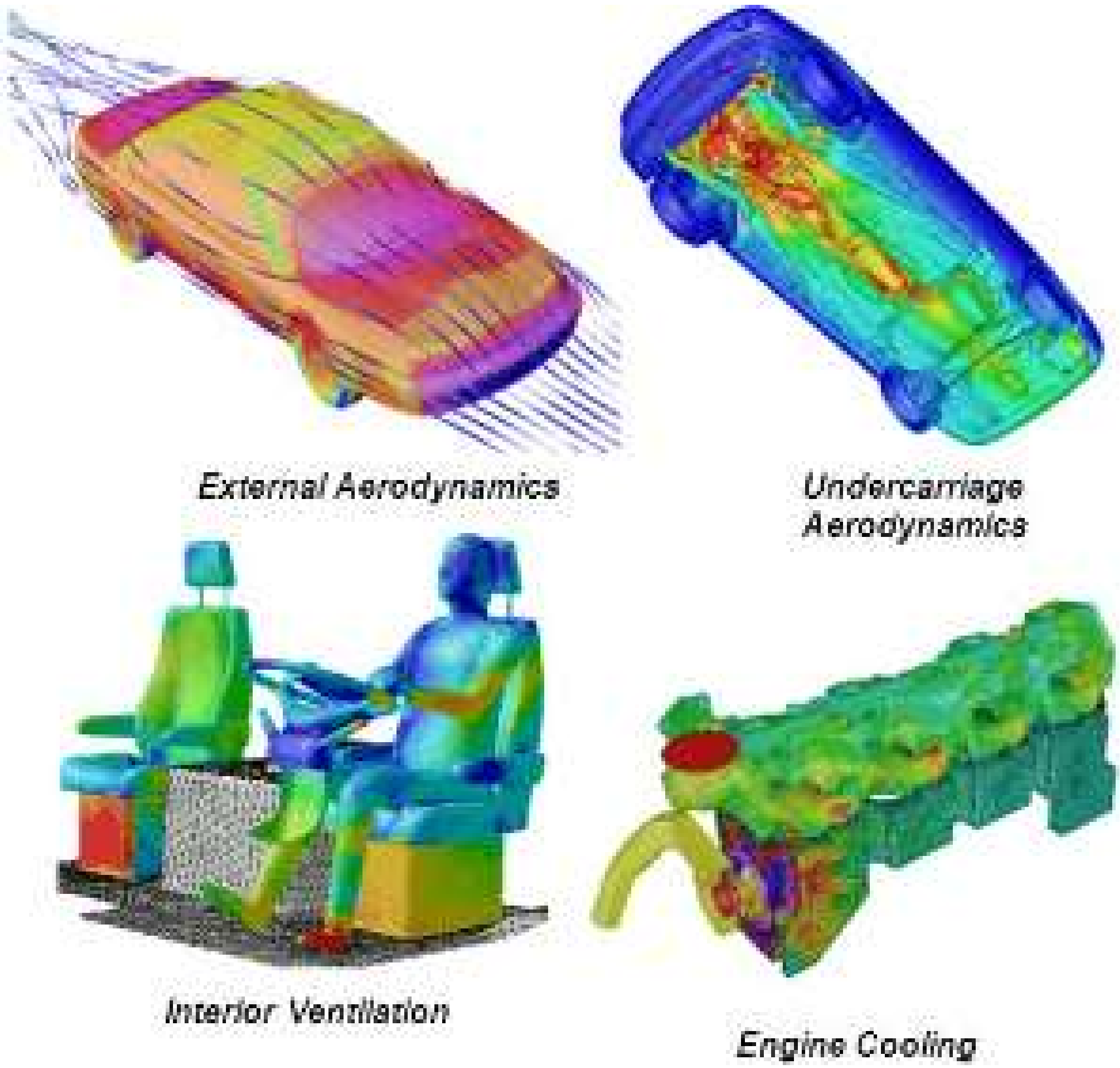


Valves, flow control

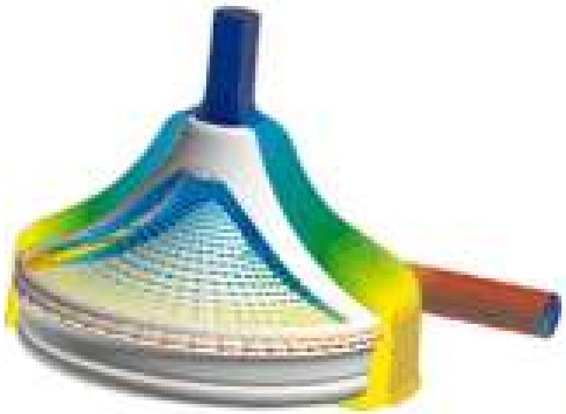


Separation and filtration

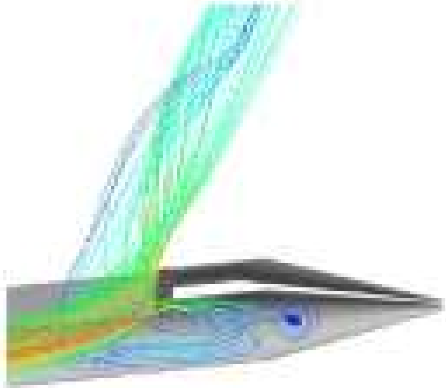
Automotive


External Aerodynamics
Undercarriage Aerodynamics
Interior Ventilation
Engine Cooling

Medical



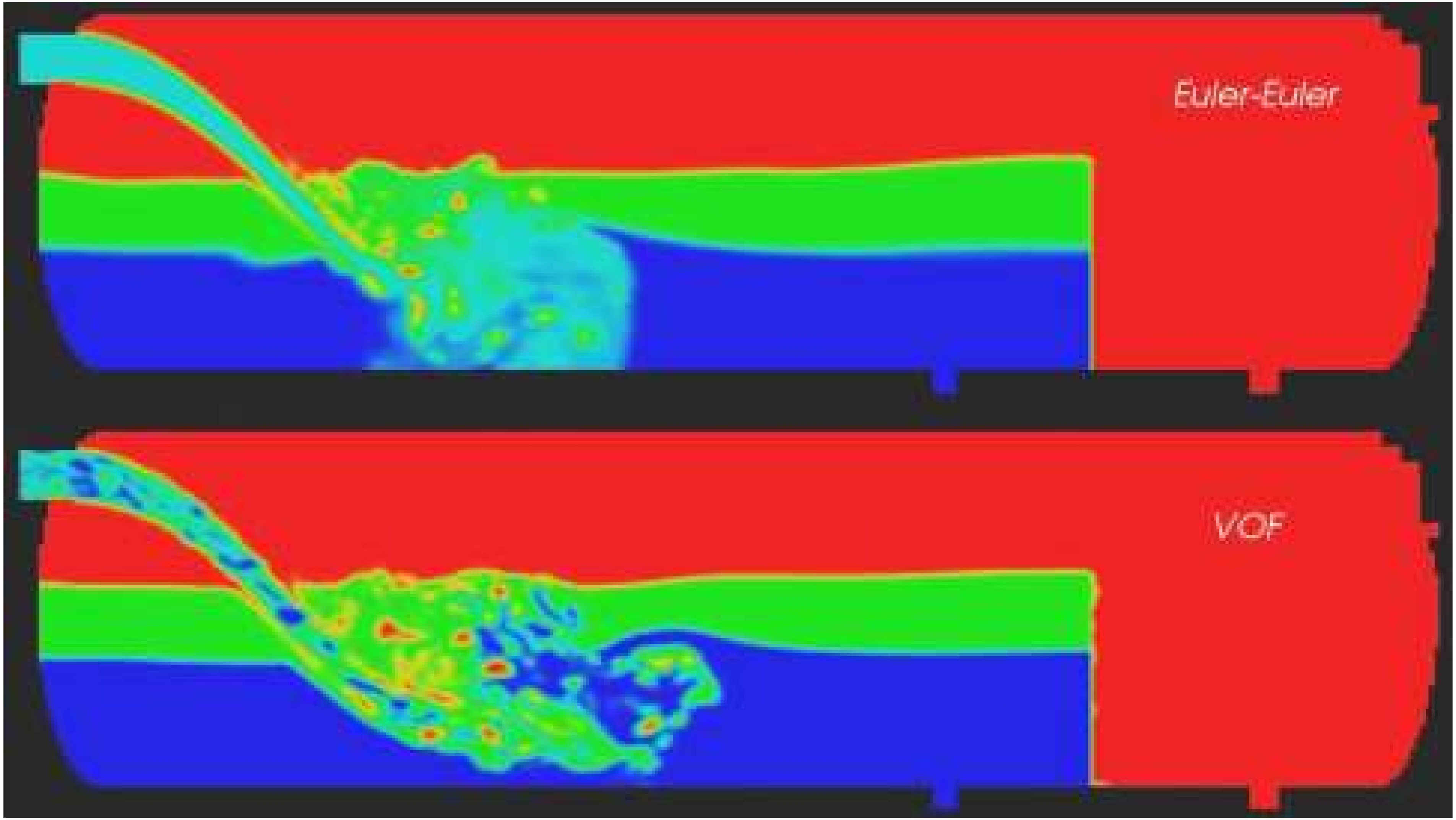
Medtronic Blood Pump



Spinal Catheter

Temperature and natural convection currents in the eye following laser heating.

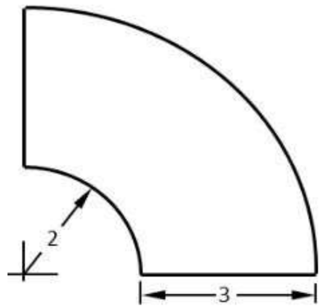
Multiphase flows



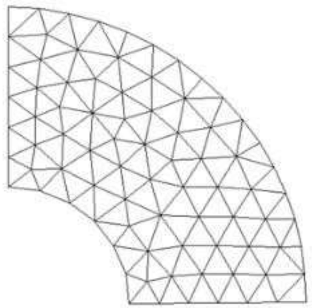
Oil- water separator

CFD SIMULATION PROCESS

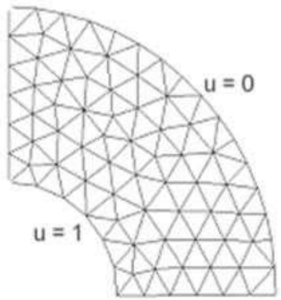
CFD Process- Illustration



1. Build geometry

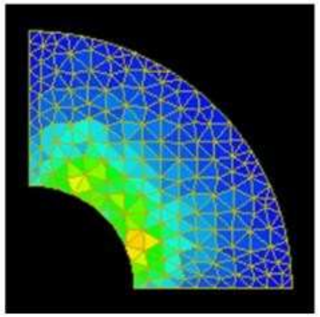


2. Mesh



3. Define boundary conditions

4. Compute



5. Visualize

CFD Process- Flow Chart

Pre processing

- CAD geometry import, clean-up
- Mesh generation

Numerical processing

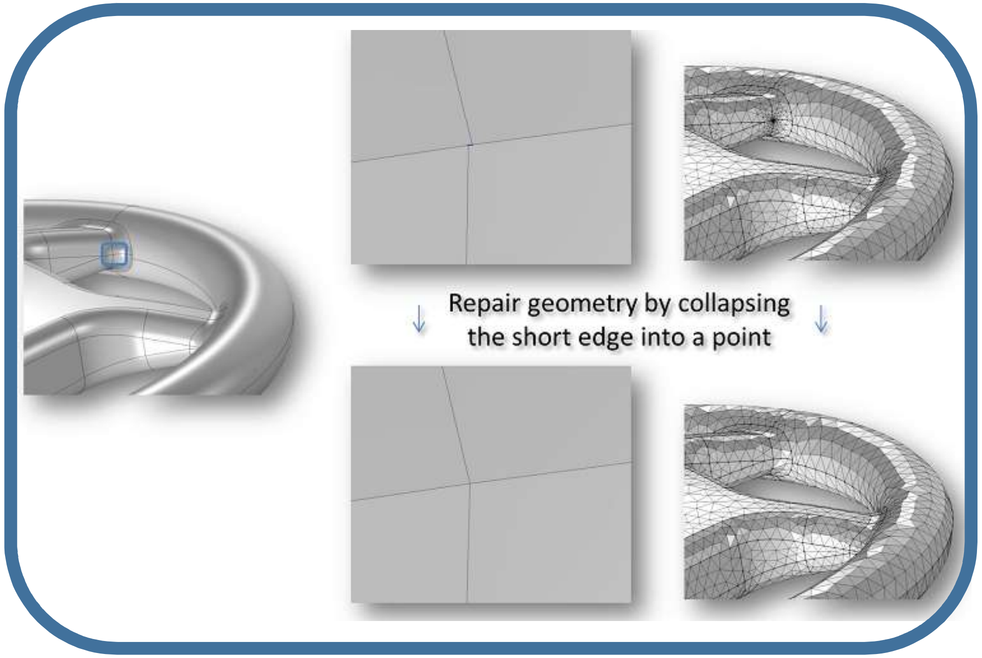
- Numerical solution of system of equations applying boundary / initial conditions
- Numerical Solver (e.g. Fluent, Open foam)

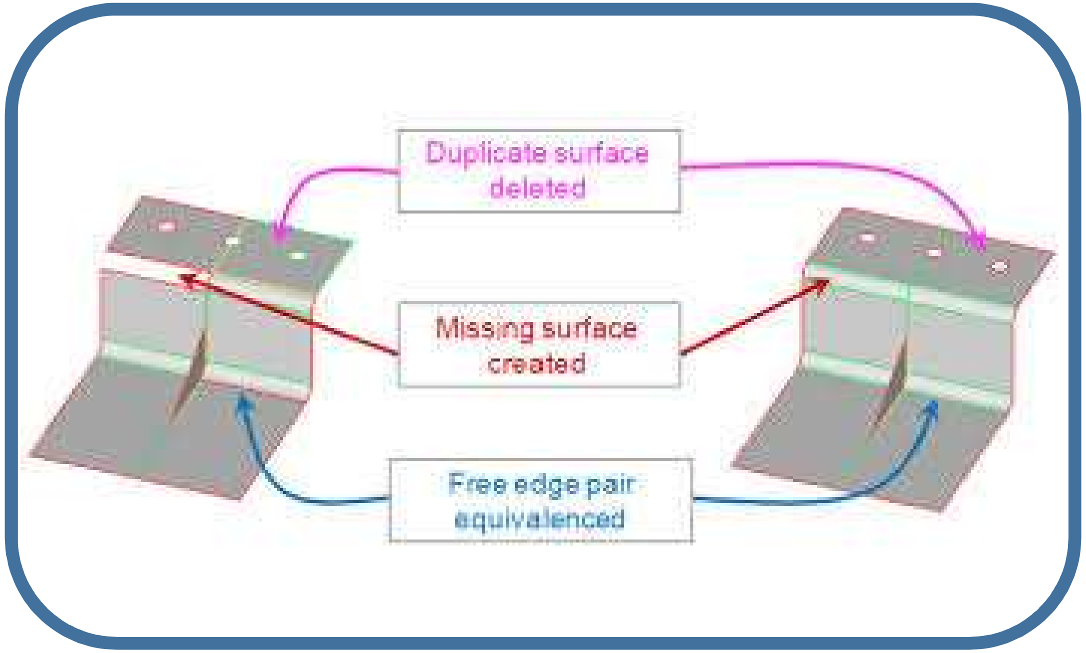
Post processing

- Data Analysis and visualization
- Fluid flow contour, streamlines etc.(e.g. paraview)

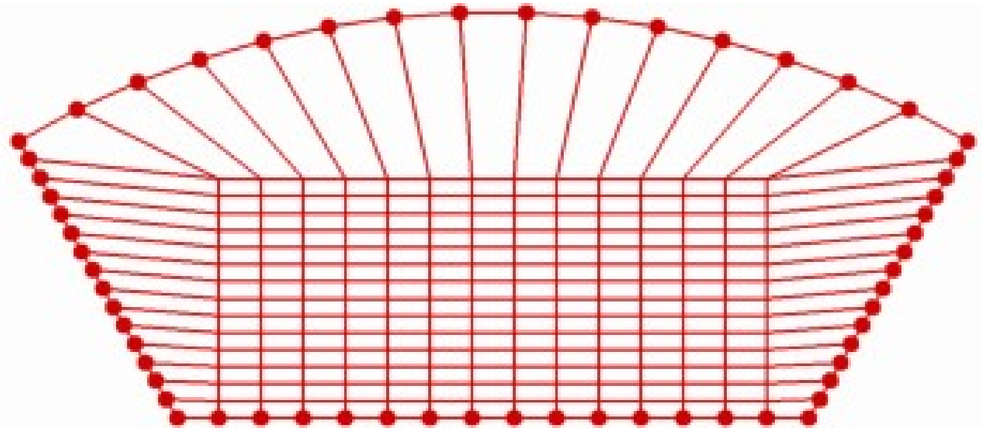
PRE-PROCESSING STAGE

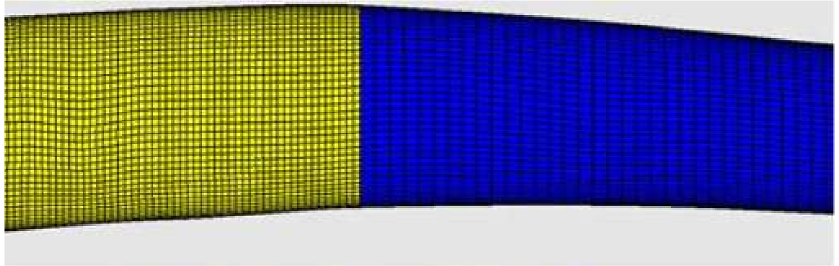
Geometry Import & Clean-up – An Illustration

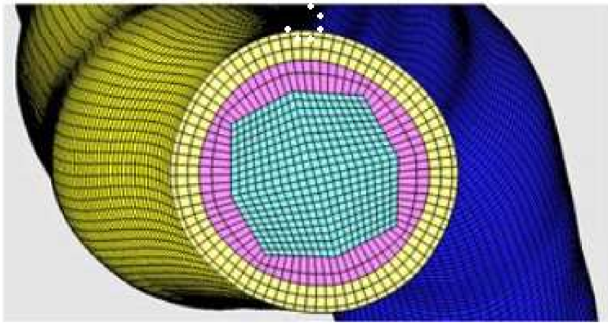


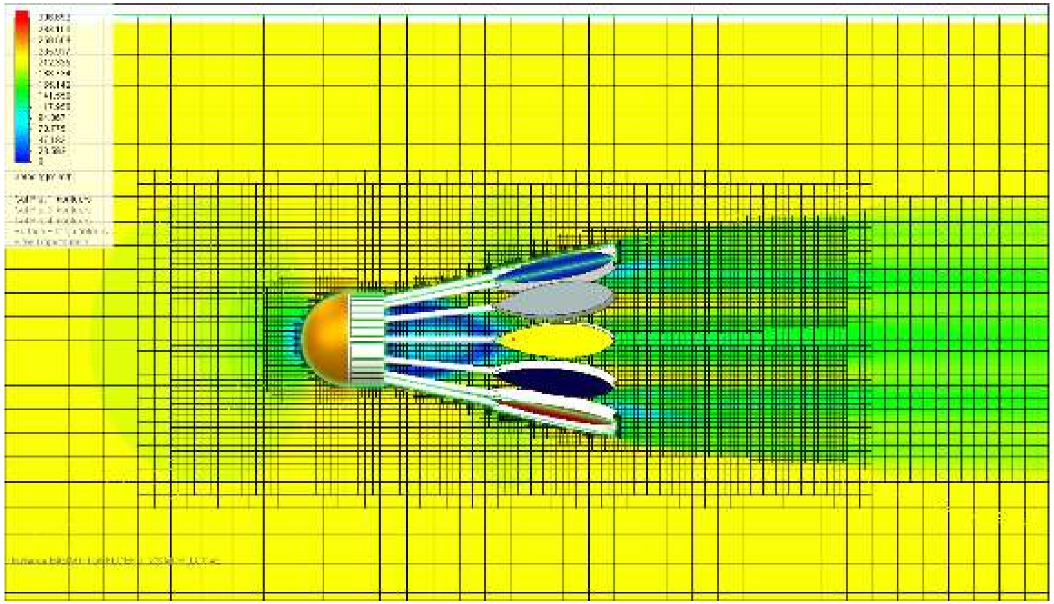


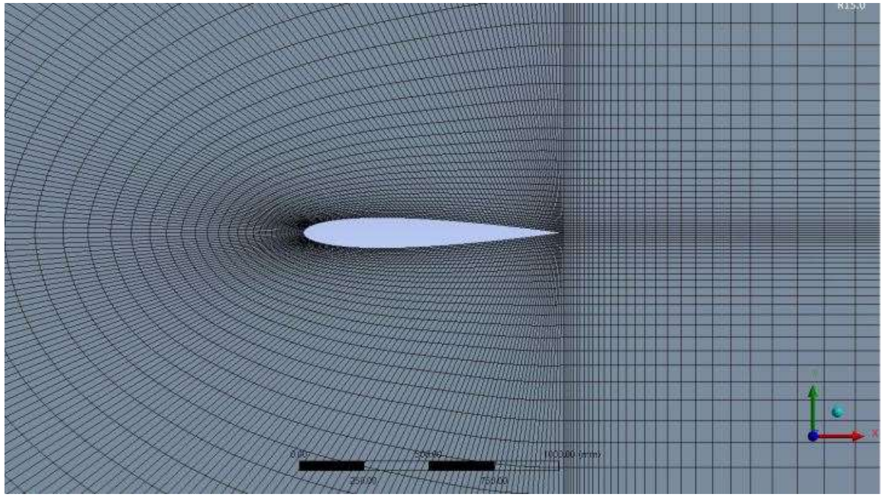
Mesh Generation – Structured Mesh



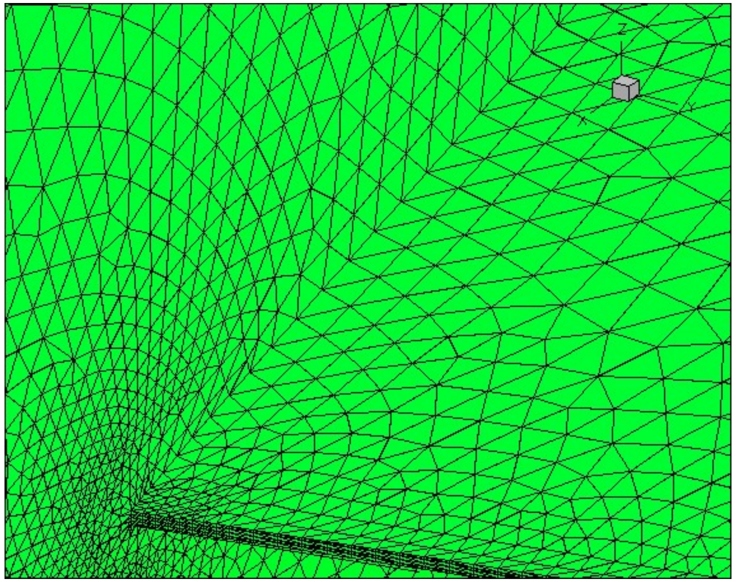


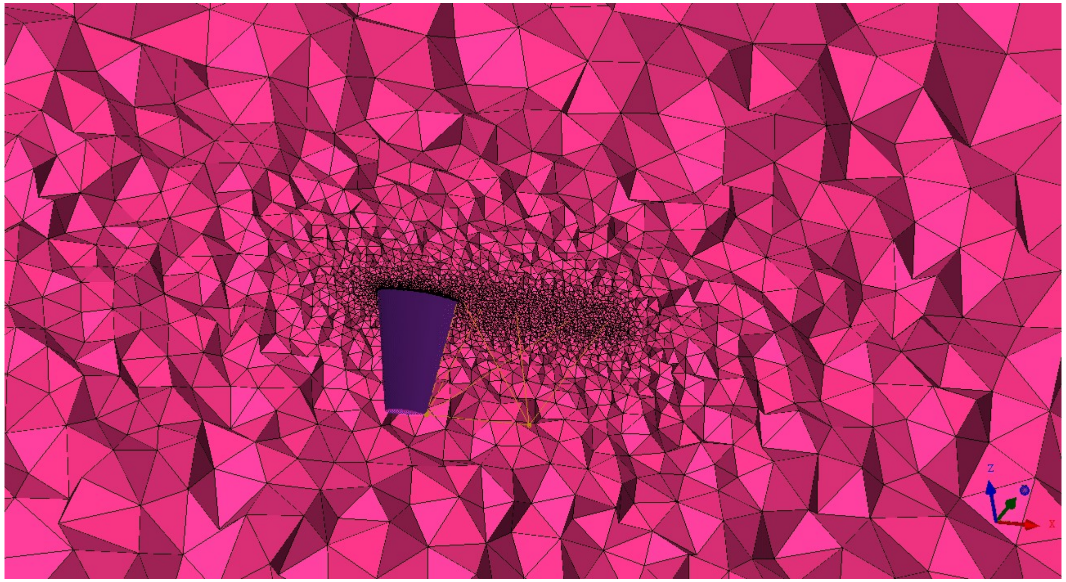


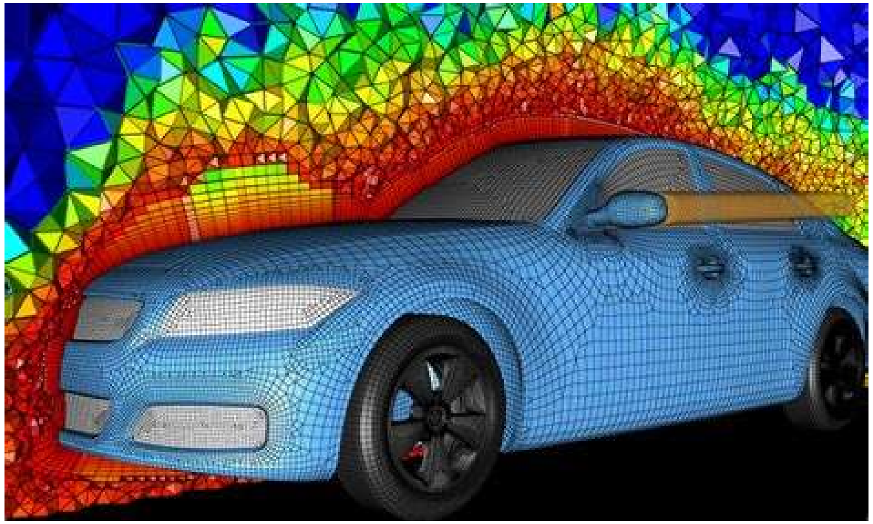


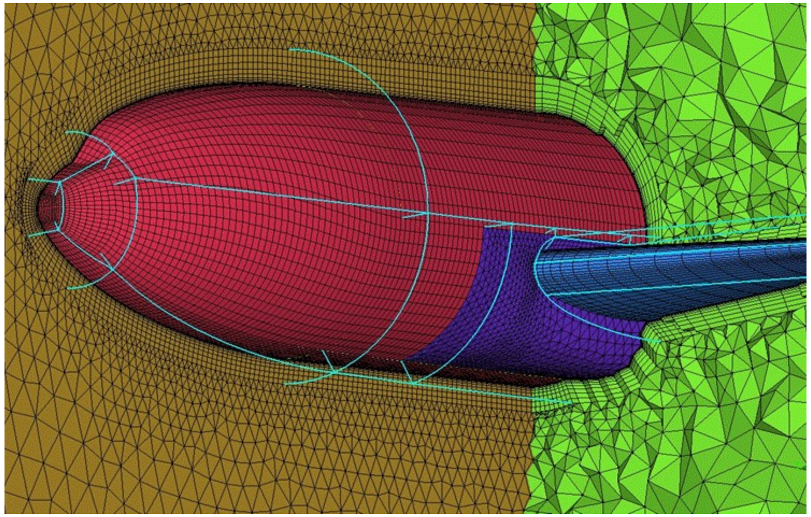


Mesh Generation – Unstructured Mesh









NUMERICAL PROCESSING STAGE

Unknowns in the Governing Equations

- In the CFD simulation, it is required to solve numerically a set of Non-linear partial differential equations called the Navier- Stokes Equations.

- For example the governing equations for incompressible flow is given as,

Continuity eq.:
$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0$$

x-mom.:
$$\rho(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z})=-\frac{\partial p}{\partial x}+\mu\nabla^{2}u+\rho g_{x}$$

y-mom.:
$$\rho(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z})=-\frac{\partial p}{\partial y}+\mu\nabla^{2}v+\rho g_{y}$$

z-mom.:
$$\rho(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z})=-\frac{\partial p}{\partial z}+\mu\nabla^{2}w+\rho g_{z}$$

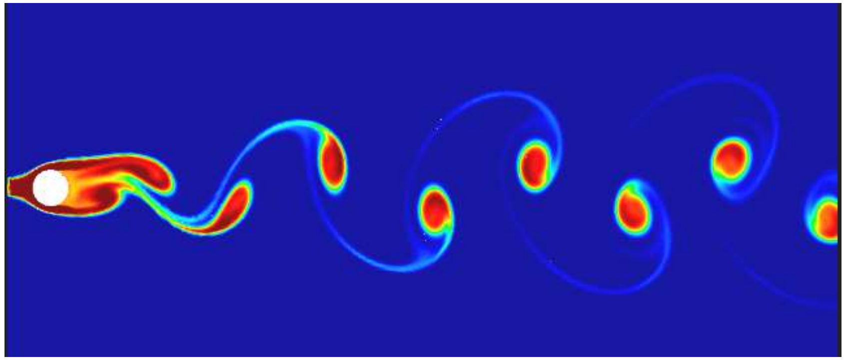
- These equations governs the laws of conservation of mass, momentum.

- The unknown includes the velocity and pressure of the fluid at several discrete points.

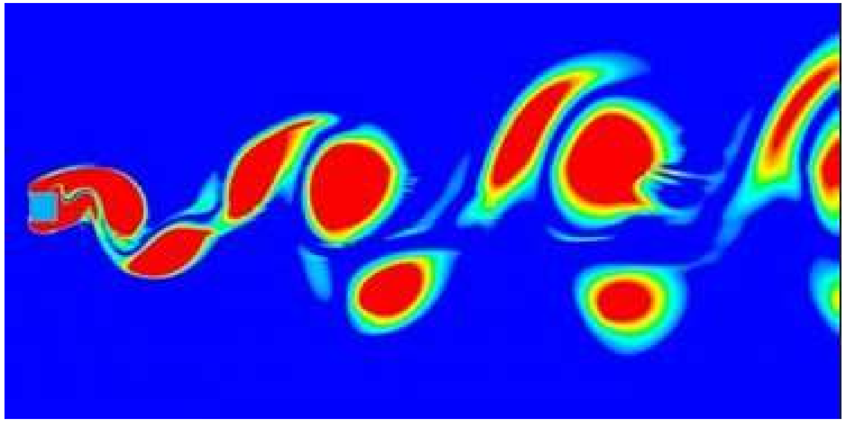
- There are several pressure and velocity correction based algorithms available to solve these equations (e.g. SIMPLE)

Types of Fluid Flow Problems

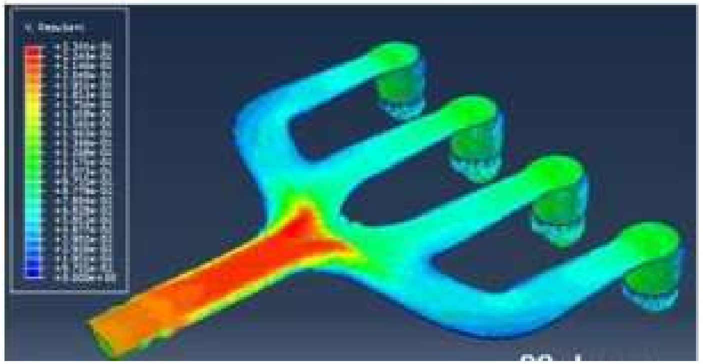
- In the CFD simulation, the fluid flow problems are broadly classified into external and internal flow problems.
- Further classification include steady or unsteady, compressible or incompressible, Laminar or Turbulent flow, one or two or three dimensional flows, natural or forced convection flows.



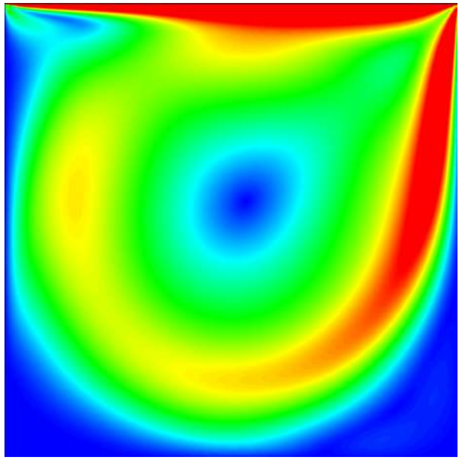
Unsteady external flow past a circular cylinder



Unsteady external flow past a square cylinder



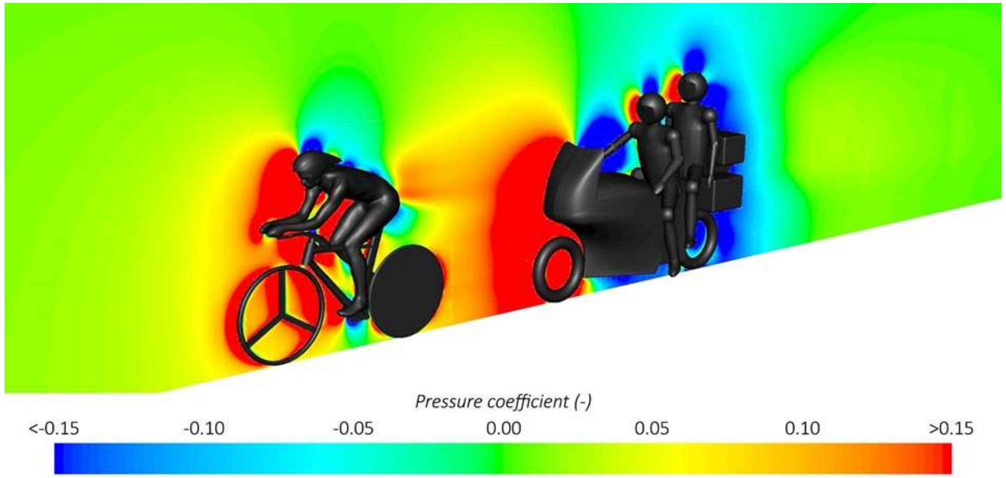
Internal flow in a pipe

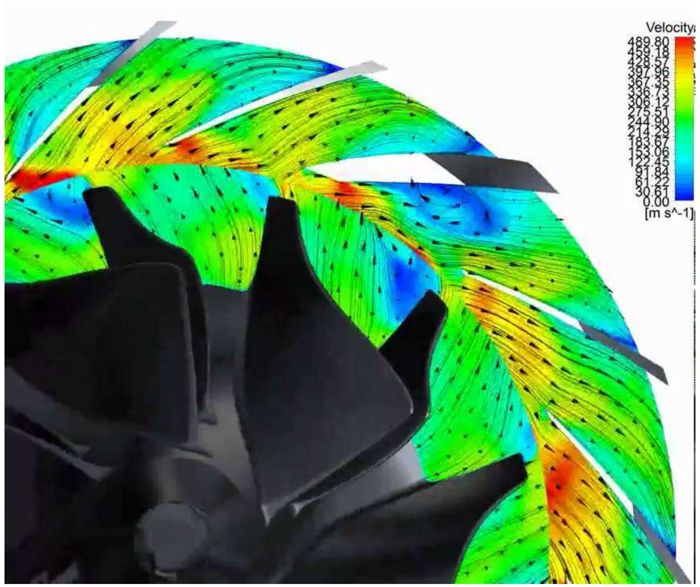


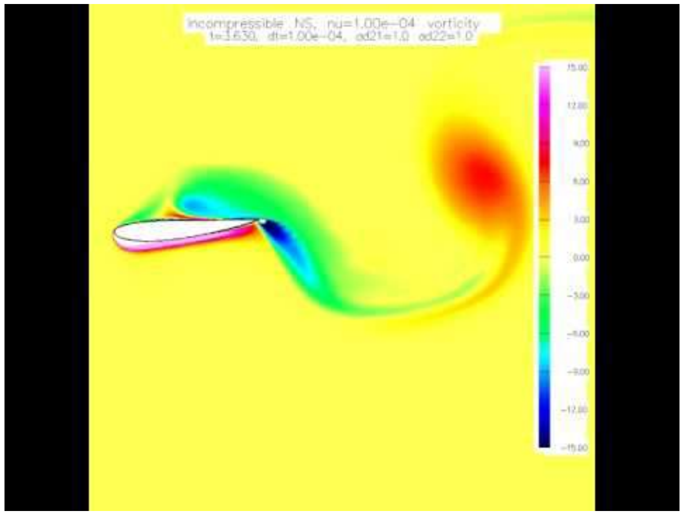
Flow inside a lid driven cavity

POST PROCESSING STAGE

Data Analysis & Visualization- An Illustration







CLASSIFICATION OF PARTIAL DIFFERENTIAL EQUATION CHARACTERISTICS

Characteristics of PDE Systems

Consider the linear PDE system

$$A\frac{\partial^2 \phi}{\partial x^2} + B\frac{\partial^2 \phi}{\partial x \partial y} + C\frac{\partial^2 \phi}{\partial y^2} = 0$$

This system is said to be elliptic for the case $B^2 - 4AC < 0.$

It is parabolic if $B^2 - 4AC = 0.$

It is hyperbolic when $B^2 - 4AC > 0.$

Elliptic PDE

- Consider steady two dimensional heat conduction governed by the equation

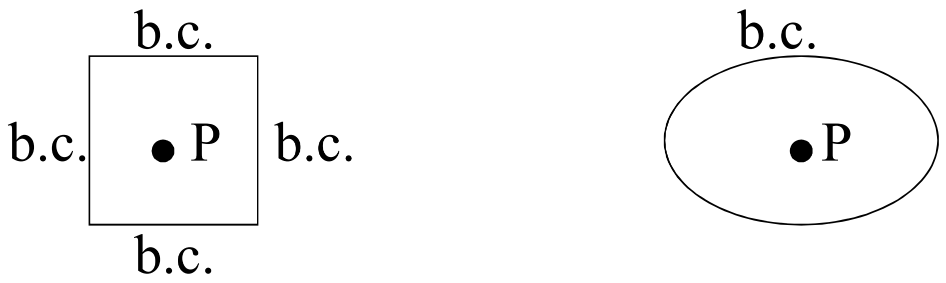
$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$

- Here, A = C = k and B = 0. Hence $B^2-4AC=-4k^2<0.$

- Therefore, the system is elliptic.

- For an elliptic PDE, the boundary conditions need to be given on a closed boundary.

- In other words, the boundary conditions all around influence the solution at a point



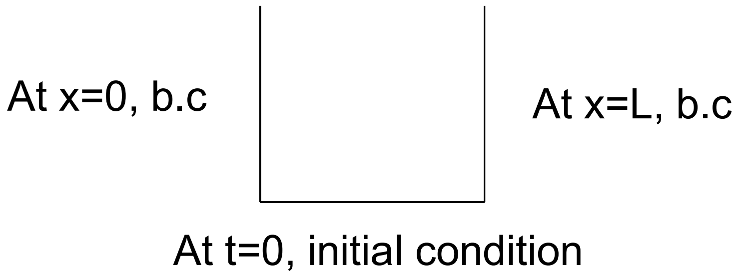
Boundary conditions for elliptic systems

Parabolic PDE

- Transient heat conduction problem which follows the governing equation

$$\frac{\partial T}{\partial t} = \frac{k}{\rho c}\frac{\partial^2 T}{\partial x^2} = \alpha \frac{\partial^2 T}{\partial x^2}$$

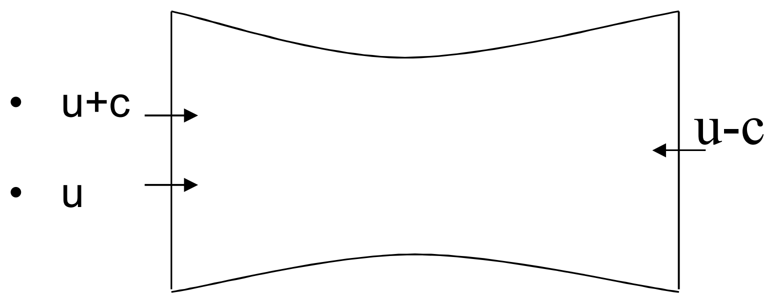
- Here, A = α, B= 0 and C = 0.
- Hence, $B^2 - 4AC = 0$
- It is a parabolic system.
- For a parabolic system the conditions need to specified as shown below.



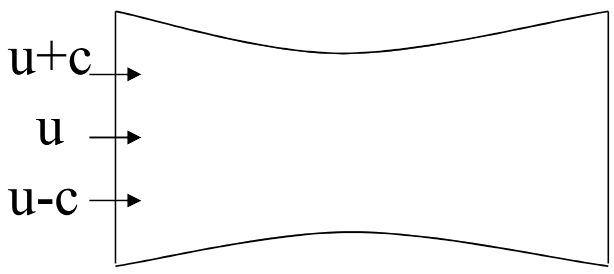
Hyperbolic PDE

- The wave equation $\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}$ is a hyperbolic system, with c denoting the acoustic speed.
- Here, B = 0 and A = 1, C = $-c^2$.
- Hence, $B^2 - 4AC = 0 - 4 x 1 x (-c^2) = 4 c^2 > 0$.
- For a hyperbolic system, there are characteristic variables which determine the number of boundary conditions to be given.
- In the above case, the two characteristics (x + ct) and (x– ct) represent the solutions corresponding to the backward-and forward- propagating waves.

Boundary Conditions for Hyperbolic PDE



Subsonic flow



Supersonic flow

- A compressible flow has three characteristic velocities i.e. u+c, u, u-c.
- Depending on the number of characteristics crossing into the domain at the boundary, the b.c. are decided.

BASIC ASPECTS OF NUMERICAL DISCRETIZATION METHODS FOR PARTIAL DIFFERENTIAL EQUATION (PDE)

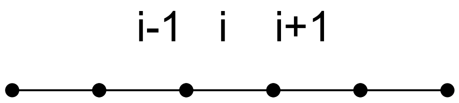
Types of Numerical Discretization Techniques

- Finite difference method
- Finite volume method
- Finite element method
- Boundary element method
- Spectral method

Finite Difference Method

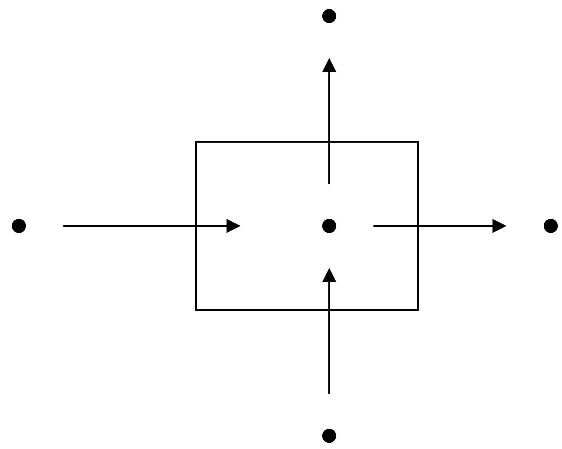
- In this method, differential equations are converted into difference expressions

$$\frac{dT}{dx} = \frac{T_i - T_{i-1}}{\Delta x} \qquad \text{or} \qquad \frac{T_{i+1} - T_i}{\Delta x}$$



Finite Volume Method

- Flux balance is applied for each cell.
- Heat flux in – Heat flux out = rate of thermal storage
- Fluxes are approximated using neighboring nodes



Finite Element Method

- While FDM & FVM are applied for flow/thermal problems, FEM was initially developed for structural problems.

- In this method, a large structure is divided into small elements and characteristic of each element is written as a matrix contribution.

- By adding contributions of all elements, wet matrix equation for the whole geometry.

APPLICATIONS OF FINITE DIFFERENCE METHOD

Taylor Series Expansions

$$T_{i-1} = T_i - \left(\frac{dT}{dx}\right)_i \Delta x + \left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x^2}{2!} - \left(\frac{d^3T}{dx^3}\right)\frac{\Delta x^3}{3!} + ... + \left(\frac{d^nT}{dx^n}\right)\frac{(-\Delta x)^n}{n!} + 0(\Delta x^{n+1})$$

$$T_{i+1} = T_i + \left(\frac{dT}{dx}\right)\Delta x + \left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x^2}{2!} + \left(\frac{d^3T}{dx^3}\right)_i \frac{\Delta x^3}{3!} + ... + \left(\frac{d^nT}{dx^n}\right)\frac{\Delta x^n}{n!} + O(\Delta x^{n+1})$$

$$\mathrm{T}_{i+2} = \mathrm{T}_i + \left(\frac{dT}{dx}\right)_i (2\Delta x) + \left(\frac{d^2T}{dx^2}\right)_i \frac{(2\Delta x)^2}{2!} + \left(\frac{d^3T}{dx^3}\right)\frac{(2\Delta x)^3}{3!}$$

$$....... + \left(\frac{d^nT}{dx^n}\right)\frac{(2\Delta x)^n}{n!} + 0\,(\Delta x^{n+1})$$

$$\mathrm{T}_{i-2} = \mathrm{T}_i - \left(\frac{dT}{dx}\right)_i (2\Delta x) + \left(\frac{d^2T}{dx^2}\right)_i \frac{(2\Delta x)^2}{2!} - \left(\frac{d^3T}{dx^3}\right)_i \frac{(2\Delta x)^3}{3!}$$

$$+ \left(\frac{d^nT}{dx^n}\right)_i \frac{(-2\Delta x)^n}{n!} + 0(\Delta x^{n+1})$$

First Derivative Approximation

Forward Difference

$$\left(\frac{dT}{dx}\right)_i = \frac{T_{i+1} - T_i}{\Delta x} - \left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x}{2!} - \left(\frac{d^3T}{dx^3}\right)_i \frac{\Delta x^2}{3!}$$

$$= \frac{T_{i+1} - T_i}{\Delta x} + 0(\Delta x)$$

Backward Difference

$$\left(\frac{dT}{dx}\right)_i = \frac{T_i - T_{i-1}}{\Delta x} - \left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x}{2!} + \left(\frac{d^3T}{dx^3}\right)_i \frac{\Delta x^2}{3!}$$

$$= \frac{T_i - T_{i-1}}{\Delta x} + 0(\Delta x)$$

Central Difference

$$\left(\frac{dT}{dx}\right)_i = \frac{T_{i+1} - T_{i-1}}{2\Delta x} + O(\Delta x^2)$$

One Sided Difference

$$4T_{i+1} - T_{i+2} = 3T_i + 2\left(\frac{dT}{dx}\right)_i \Delta x + 0(\Delta x^3)$$

$$\left(\frac{dT}{dx}\right)_i = \frac{4T_{i+1} - T_{i+2} - 3T_i}{2\Delta x} + 0(\Delta x^2)$$

Second Derivative Approximation

Central Difference

$$T_{i+1} + T_{i-1} = 2T_i + 2\left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x^2}{2!} + 2\left(\frac{d^4T}{dx^4}\right)_i \frac{\Delta x^4}{4!} + ...$$

$$\left(\frac{d^2T}{dx^2}\right)_i = \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2} + 0(\Delta x^2)$$

Estimation of Error

$$\varepsilon_i^k = T(x_i, t^k) - T^*(x_i, t^k)$$

$$\varepsilon_i^k \propto \Delta x_i^2 \quad \text{and } \varepsilon_i^k \propto \Delta t^k$$

$\varepsilon = O\ (\Delta x^2, \Delta t)$

FDM For One-D Heat Conduction

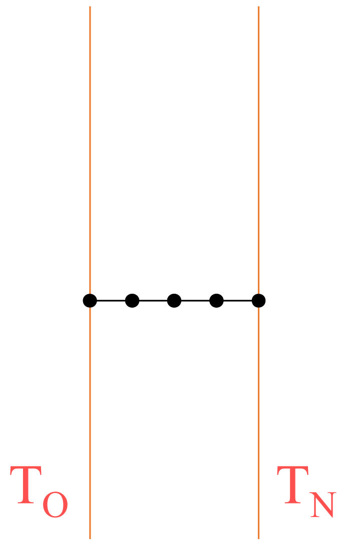
$$k\frac{d^2T}{dx^2} + Q = 0$$

$$k\left(\frac{d^2T}{dx^2}\right)_i + Q = k\frac{(T_{i+1} + T_{i-1} - 2T_i)}{\Delta x^2} + Q + 0(\Delta x^2)$$

$$T_{i+1} + T_{i-1} - 2T_i = -Q\Delta x^2/k$$

AT X = 0, T = T_O

AT X = L, T = T_N



Flux Type Boundary Condition- Method 1

$$\frac{dT}{dx} = 0 \qquad \text{at } x = L$$

$$\left(\frac{\mathrm{dT}}{\mathrm{dx}}\right)_{\mathrm{i=N+1}} = \frac{\mathrm{T_{N+2} - T_N}}{2\Delta \mathrm{x}} = 0$$

$$\frac{k(T_{N+2} + T_N - 2T_{N+1})}{\Delta x^2} + Q = 0$$

$$\frac{2k(T_N - T_{N+1})}{\Delta x^2} + Q = 0$$

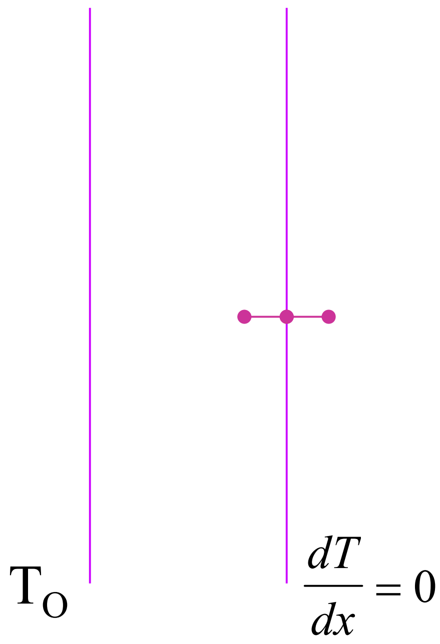


IMAGE POINT METHOD

Flux Type Boundary Condition- Method 2

Applying Taylor's series expansion at boundary point

$$T_{i+1} = T_i + \left(\frac{dT}{dx}\right)\Delta x + \left(\frac{d^2T}{dx^2}\right)_i \frac{\Delta x^2}{2!} + \left(\frac{d^3T}{dx^3}\right)_i \frac{\Delta x^3}{3!} + \ldots + \left(\frac{d^nT}{dx^n}\right)\frac{\Delta x^n}{n!} + O(\Delta x^{n+1})$$

$dT/dx = 0$ and $d^2T/dx^2 = -Q/k$ and higher order terms are zero. Hence

$T_{N+1} = T_N – Q\ Dx^2/2k$

Polynomial Expansion

It is possible to use local Polynomial expansions of the form

$$T = A x^2 + B x + C$$

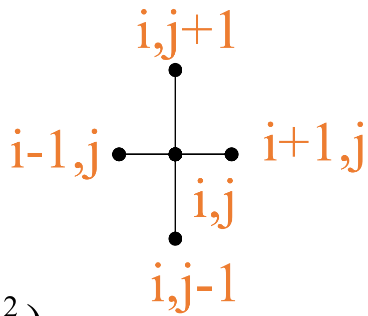
and use three nodes to fit a quadratic expression for the variable. From such an expansion the required derivatives at boundary can be evaluated for implementing the flux type BC

Matrix Form For Flux Type BC

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix} = \begin{bmatrix} T_b \\ -Q\Delta x^2 / k \\ -Q\Delta x^2 / k \\ -Q\Delta x^2 / k \\ -Q\Delta x^2 / k \\ -Q\Delta x^2 / 2k \end{bmatrix}$$

Two-D Heat Conduction

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+Q=0$$



$$\left(\frac{\partial^2 T}{\partial x^2}\right)_{i,j}=\frac{T_{i+1,j}+T_{i-1,j}-2T_{i,j}}{\Delta x^2}+0(\Delta x^2)$$

$$\left(\frac{\partial^2 T}{\partial y^2}\right)_{i,j}=\frac{T_{i,j+1}+T_{i,j-1}-2T_{i,j}}{\Delta y^2}+0(\Delta y^2)$$

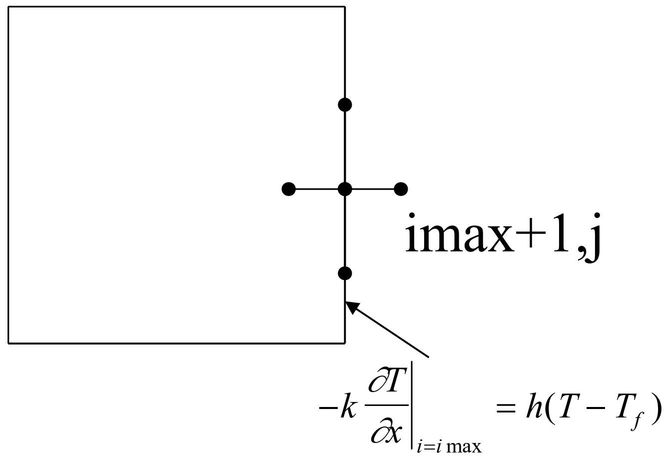
$$\frac{k(T_{i+1,j}+T_{i-1,j}-2T_{i,j})}{\Delta x^2}+\frac{k(T_{i,j+1}+T_{i,j-1}-2T_{i,j})}{\Delta y^2}+Q=0$$

Implementation of BC

$$T_{i+1,j} + T_{i-1,j} + \beta^2 (T_{i,j+1} + T_{i,j-1}) - 2(1+\beta^2) T_{i,j} = \frac{-Q\Delta x^2}{k}$$

where the grid aspect ratio $\beta = \Delta x/\Delta y$. Consider the boundary condition

$$-k \left.\frac{\partial T}{\partial x}\right|_{i=i\max} = h(T - T_f)$$



$$T_{i-1,j} = T_{i,j} - \left(\frac{\partial T}{\partial x}\right)_{i,j} \Delta x + \left(\frac{\partial^2 T}{\partial x^2}\right)_{i,j} \frac{\Delta x^2}{2!} + 0(\Delta x^3)$$

$$T_{i-1,j} = T_{i,j} + \{ h\,(T_{i,j} - T_f)/k\}\Delta x \quad - \frac{Q\Delta x^2}{2k} - \frac{\beta^2 (T_{i,j+1} + T_{i,j-1} - 2T_{i.j})}{2}$$

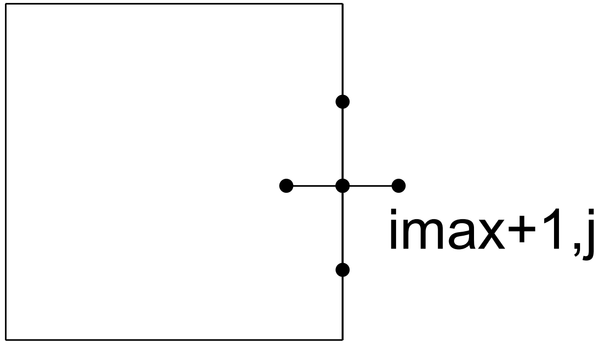
The same expression is obtained by image point method

Convective Boundary Condition

At i=imax, $-k\dfrac{\partial T}{\partial x} = h(T - T_f)$

$$\frac{\partial T}{\partial x} = \frac{T_{i\max+1,j} - T_{i\max-1,j}}{\Delta x} = \frac{-h(T_{i\max,j} - T_f)}{k}$$

$$T_{i\max+1,j} = T_{i\max-1,j} - \frac{h\Delta x(T_{i\max,j} - T_f)}{k}$$



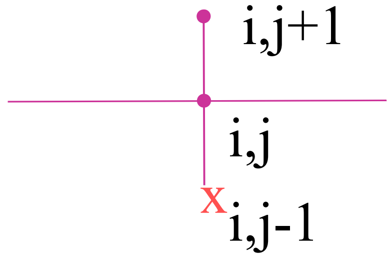
Applying heat balance at node imax, we have

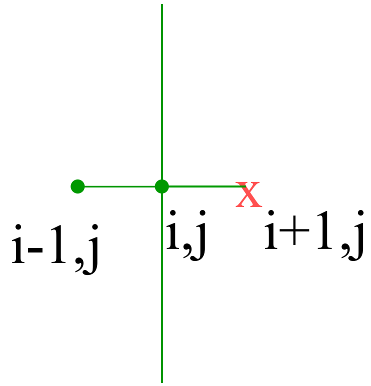
$$T_{i\max+1,j} + T_{i\max-1,j} + \beta^2(T_{i\max,j+1} + T_{i\max,j-1}) - 2(1+\beta^2)T_{i\max,j} = -\frac{Q\Delta x^2}{k}$$

Substituting for the image point temperature, we get:

$$2T_{i\max-1,j} - \frac{h\Delta x(T_{i\max,j} - T_f)}{k} + \beta^2(T_{i\max,j+1} + T_{i\max,j-1}) - 2(1+\beta^2)T_{i\max,j} = -\frac{Q\Delta x^2}{k}$$

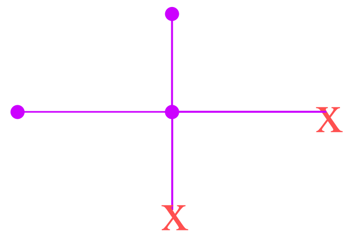
Image Point Method





Using image point, discretize the boundary condition and substitute in governing equation

For corner points with two flux type bc



Solution Methods

Point –by-Point Method

$$2(1+\beta^2)\, T_{i,j} = T^*_{i+1,j} + T^*_{i-1,j} + \beta^2(T^*_{i,j+1} + T^*_{i,j-1}) + Q\Delta x^2/k$$

Line-by-Line Method

$$T_{i+1,j} + T_{i-1,j} - 2(1+\beta^2)T_{i,j} = \frac{-Q\Delta x^2}{k} - \beta^2(T^*_{i,j+1} + T^*_{i,j-1})$$

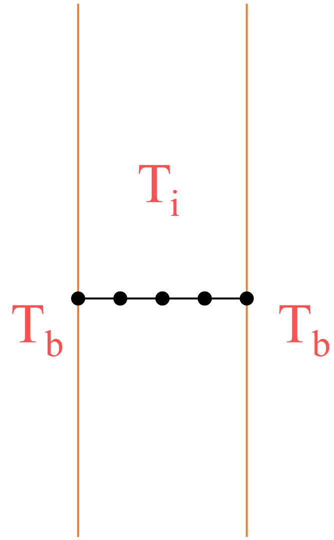
$$\beta^2(T_{i,j+1} + T_{i,j-1}) - 2(1+\beta^2)T_{i,j} = \frac{-Q\Delta x^2}{k} - T^*_{i+1,j} - T^*_{i-1,j}$$

Under-relaxation/ Over-relaxation

$$T_{i,j}^{k+1} = W \times T_{i,j} + (1-W)T_{i,j}^{k}$$

Transient Heat Conduction

$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + Q$$



Boundary Conditions:

$T = T_b$ at $x = 0$ and $x = L$

Initial Condition:

$T = T_i$ for all $0 < x < L$

Methods For Transient Marching

- Explicit method
- Implicit method
- Semi-implicit method (Crank- Nicolson technique)

Explicit Method

$$\left(\frac{\partial T}{\partial t}\right)_i^n = \alpha\left(\frac{\partial^2 T}{\partial x^2}\right)_i^n$$

$$T_i^{n+1} = T_i^n + \Delta t\left(\frac{\partial T}{\partial t}\right)_i^n$$

$$T_i^{n+1} = T_i^n + (\alpha\Delta t / \Delta x^2)(T_{i+1}^n + T_{i-1}^n - 2T_i^n)$$

Implicit Method

$$T_i^{n+1} = T_i^n + \Delta t \left(\frac{\partial T}{\partial t} \right)_i^{n+1}$$

$$\left(\frac{\partial T}{\partial t} \right)_i^{n+1} = \alpha \left(\frac{\partial^2 T}{\partial x^2} \right)_i^{n+1}$$

$$T_i^{n+1} - (\alpha \Delta t / \Delta x^2)(T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1}) = T_i^n$$

Semi-Implicit method

$$T_i^{n+1} = T_i^{n} + (\Delta t / 2)\left\{\left(\frac{\partial T}{\partial t}\right)_i^{n} + \left(\frac{\partial T}{\partial t}\right)_i^{n+1}\right\}$$

$$T_i^{n+1} - (\alpha \Delta t / 2\Delta x^2)(T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1}) = T_i^{n} + (\alpha \Delta t / 2\Delta x^2)(T_{i+1}^{n} + T_{i-1}^{n} - 2T_i^{n})$$

Comparison of Implicit/ Explicit methods

- Explicit method involves pointwise updating & requires no matrix inversion. Implicit Scheme needs Matrix inversion

- Computational time per time step is more for Implicit method than the Explicit.

- From stability considerations, explicit scheme may require very small time steps and hence several thousand steps to obtain steady state solution. Large time steps can be used in implicit scheme

- Both explicit & implicit methods are $O(\Delta t)$ while Semi-implicit scheme is second order accurate

Alternating Direction Implicit Method

$$\rho C_p \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) + Q$$

X-Dir. Implicit

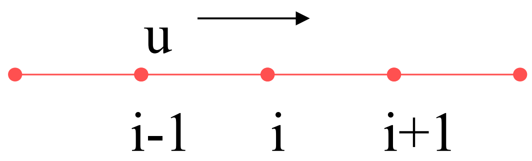
$$\rho C_p \frac{\partial T}{\partial t} = k\left[\left(\frac{\partial^2 T}{\partial x^2}\right)^{n+1} + \left(\frac{\partial^2 T}{\partial y^2}\right)^{n}\right] + Q$$

Y-Dir. Implicit

$$\rho C_p \frac{\partial T}{\partial t} = k\left[\left(\frac{\partial^2 T}{\partial x^2}\right)^{n+1} + \left(\frac{\partial^2 T}{\partial y^2}\right)^{n+2}\right] + Q$$

One-D Convection Diffusion Equation

$$u\frac{dT}{dx} = \alpha\frac{d^2T}{dx^2}$$



Using Central Difference Scheme

$$u\frac{T_{i+1} - T_{i-1}}{2\Delta x} = \alpha\frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2}$$

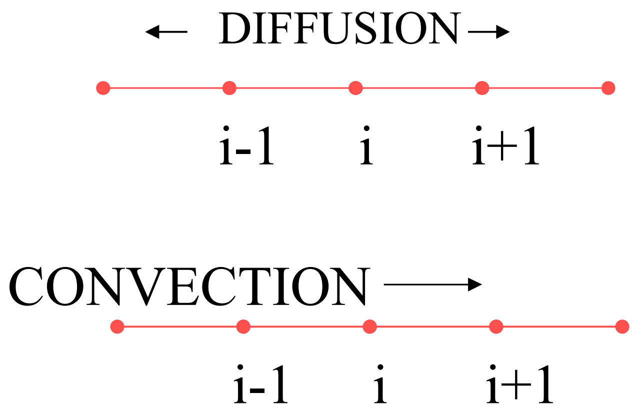
$$2T_i - (1 - Pe_c/2)T_{i+1} - (1 + Pe_c/2)T_{i-1} = 0$$

$$\mathrm{Pe_c} = \mathrm{u}\Delta\mathrm{x}/\alpha$$

Cell Pe < 2 for spatial stability, when central difference is used

Upwind Differencing

$$u\frac{T_i - T_{i-1}}{\Delta x} = \alpha \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2}$$



For U>0

$$2(1 + Pe_c)T_i - T_{i+1} - (1 + Pe_c)T_{i-1} = 0$$

For U<0

$$2(1+|Pe_c|)T_i - (1+|Pe_c|)T_{i+1} - T_{i-1} = 0$$

Artificial Diffusion

Central Difference:

$$2T_i - (1 - Pe_c / 2)T_{i+1} - (1 + Pe_c / 2)T_{i-1} = 0$$

Upwind Difference:

$$2(1 + Pe_c)T_i - T_{i+1} - (1 + Pe_c)T_{i-1} = 0$$

$$DIFFERENCE = (Pe_c / 2)(T_{i+1} + T_{i-1} - 2T_i)$$

Artificial Diffusion

$$u\frac{dT}{dx} = \alpha\frac{d^2T}{dx^2} + \alpha_a\frac{d^2T}{dx^2}$$

The last term on the right is the artificial diffusion term

$$(u\Delta x/2\alpha)(T_{i+1} - T_{i-1}) = (1 + \frac{\alpha_a}{\alpha})(T_{i+1} + T_{i-1} - 2T_i)$$

By setting $(\alpha_a/\alpha) = Pe_c/2$, one can get the upwind form from central difference form

Upwinding & Artificial Diffusion

- Upwinding can be done with higher order accuracy.

- For node i, we can consider the nodes (i-2), (i-1) and (i) to get second order accurate expression for convective term. Even nodes (i-2), (i-1), (i) and (i+1) can be taken for third order accuracy.

- For artificial diffusion 2^{nd} order, or 4^{th} order or 6^{th} order expressions etc. can be used.

Higher order Artificial Diffusion

$$u\frac{dT}{dx} = \alpha\frac{d^2T}{dx^2} + \alpha_a^{II}\frac{d^2T}{dx^2} + \alpha_a^{IV}\frac{d^4T}{dx^4} + \alpha_a^{VI}\frac{d^6T}{dx^6}$$

$$\frac{d^2T}{dx^2} = \frac{T_{i+1} + T_{i-1} - 2T_i}{\Delta x^2}$$

$$\frac{d^4T}{dx^4} = \frac{T_{i+2} - 4T_{i+1} + 6T_i - 4T_{i-1} + T_{i-2}}{\Delta x^4}$$

Artificial Diffusion

- Can be used in flow direction for high speed flows to avoid numerical oscillations; need not be used in cross- flow direction

- Can be used to smoothen the solution at shocks & high gradient regions

NUMERICAL ALGORTHIM TO SOLVE NAVIER STOKES EQUATION- PRESSURE CORRECTION APPROACH

Velocity-Pressure Formulation

Continuity Equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

X-Momentum Eq. (For Updating U Velocity):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\mu}{\rho}\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\}$$

Y-Momentum Eq. (For Updating V Velocity) :

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{\mu}{\rho}\{\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\}$$

SIMPLE Method

Semi- IMplicit Pressure Linked Equation Solver-- SIMPLE

X-mom.:

$$\frac{\partial u}{\partial t} = -\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right)\Bigg|^{n} - \frac{1}{\rho}\frac{\partial p}{\partial x}\Bigg|^{n+1} + \frac{\mu}{\rho}\left\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right\}\Bigg|^{n}$$

$$u^{n+1} = u^{n} - \Delta t.\left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right)\Bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial x}\Bigg|^{n+1} + \Delta t.\frac{\mu}{\rho}\left\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right\}\Bigg|^{n}$$

Y-mom.:

$$\frac{\partial v}{\partial t} = -\left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right)\Bigg|^{n} - \frac{1}{\rho}\frac{\partial p}{\partial y}\Bigg|^{n+1} + \frac{\mu}{\rho}\left\{\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right\}\Bigg|^{n}$$

$$v^{n+1} = v^{n} - \Delta t.\left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right)\Bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial y}\Bigg|^{n+1} + \Delta t.\frac{\mu}{\rho}\left\{\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right\}\Bigg|^{n}$$

Velocity Correction Equation – X Momentum

$$u^{n+1} = u^{n} - \Delta t.(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y})\Bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial x}\Bigg|^{n+1} + \Delta t.\frac{\mu}{\rho}\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\}\Bigg|^{n}$$

$$u^{*} = u^{n} - \Delta t.(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y})\Bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial x}\Bigg|^{*} + \Delta t.\frac{\mu}{\rho}\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\}\Bigg|^{n}$$

$$u^{n+1} - u^{*} = -\Delta t.\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)^{n+1} + \Delta t.\left(\frac{1}{\rho}\frac{\partial p}{\partial x}\right)^{*}$$

Velocity Correction Equation – Y Momentum

$$v^{n+1} = v^{n} - \Delta t.(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y})\bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial y}\bigg|^{n+1} + \Delta t.\frac{\mu}{\rho}\{\frac{\partial^{2} v}{\partial x^{2}} + \frac{\partial^{2} v}{\partial y^{2}}\}\bigg|^{n}$$

$$v^{*} = v^{n} - \Delta t.(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y})\bigg|^{n} - \Delta t.\frac{1}{\rho}\frac{\partial p}{\partial y}\bigg|^{*} + \Delta t.\frac{\mu}{\rho}\{\frac{\partial^{2} v}{\partial x^{2}} + \frac{\partial^{2} v}{\partial y^{2}}\}\bigg|^{n}$$

$$v^{n+1} - v^{*} = -\Delta t.\left(\frac{1}{\rho}\frac{\partial p}{\partial y}\right)^{n+1} + \Delta t.\left(\frac{1}{\rho}\frac{\partial p}{\partial y}\right)^{*}$$

Pressure Corrections

Define

$$u' = u^{n+1} - u^* \qquad v' = v^{n+1} - v^* \qquad p' = p^{n+1} - p^*$$

It can be shown that

$$u' = -\frac{\Delta t}{\rho}\frac{\partial p'}{\partial x} \qquad v' = -\frac{\Delta t}{\rho}\frac{\partial p'}{\partial y}$$

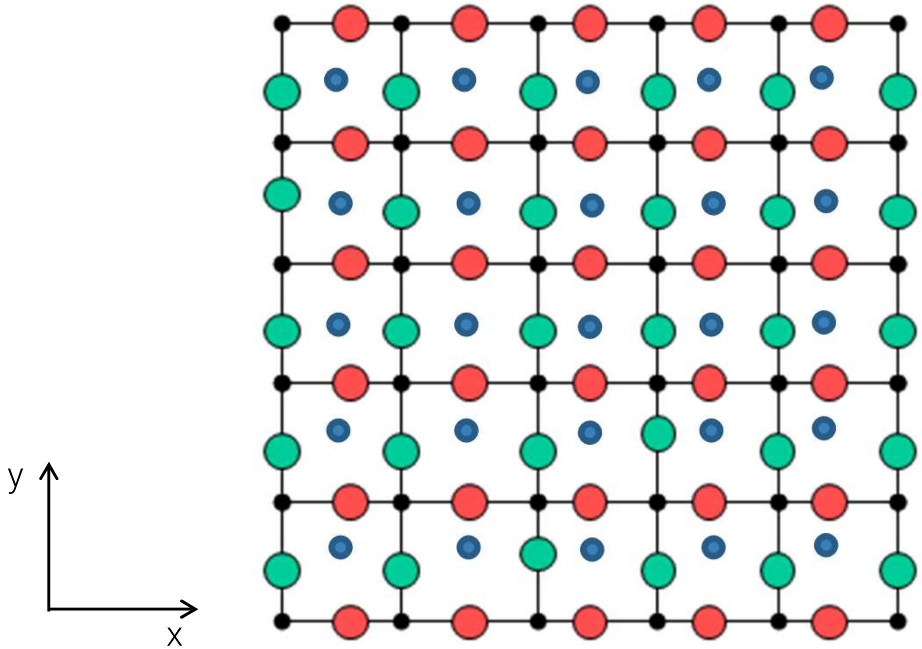
Substituting for velocity & pressure corrections, we get

$$\frac{\partial^2 p'}{\partial x^2} + \frac{\partial^2 p'}{\partial y^2} = -\frac{\rho}{\Delta t}\left(\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y}\right) = \frac{\rho}{\Delta t}\left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y}\right)$$

Steps Involved In SIMPLE

- At the start of a time step, assume a guess pressure field p^*
- Solve momentum equations to get guess velocities u^* and v^* at each node
- Using u^* and v^* calculate continuity residue at each point
- From continuity equation residue, solve for pressure correction p' at each node
- Using p' solve for velocity corrections
- Update variables as $p^{n+1}=p^*+p'$, $u^{n+1}=u^*+u'$, $v^{n+1}= v^*+v'$
- And go to next time step

Staggered & Collocated Mesh

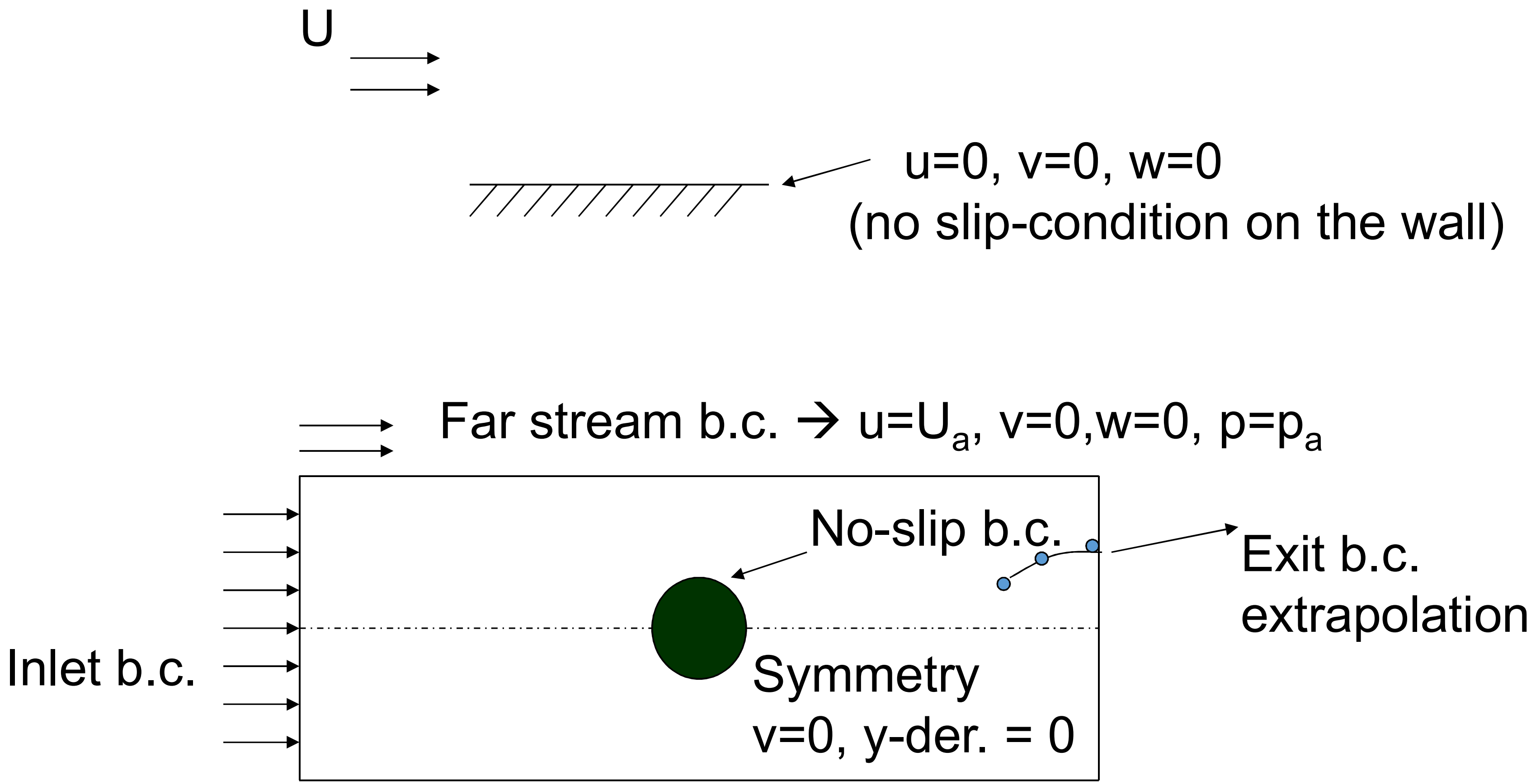


	Staggered	Semi-Staggered	Collocated
	V- velocity	V- velocity	-
	U- velocity	U- velocity	-
	Cell vertices	Pressure (Cell vertices)	-
	Pressure (Cell centers)	Cell centers	U,V- velocities, Pressure

Staggered Mesh Procedure

- Pressure nodes are taken as the main nodes.
- x-velocity (u) nodes are shifted by dx/2 with reference to pressure nodes .
- and y-velocity (v) nodes are shifted by dy/2 with reference to pressure nodes.
- Such a staggered mesh avoids odd-even decoupling (chequer-board configuration) between velocities & pressures .

Typical Flow Boundary Conditions



Typical Thermal Boundary Conditions