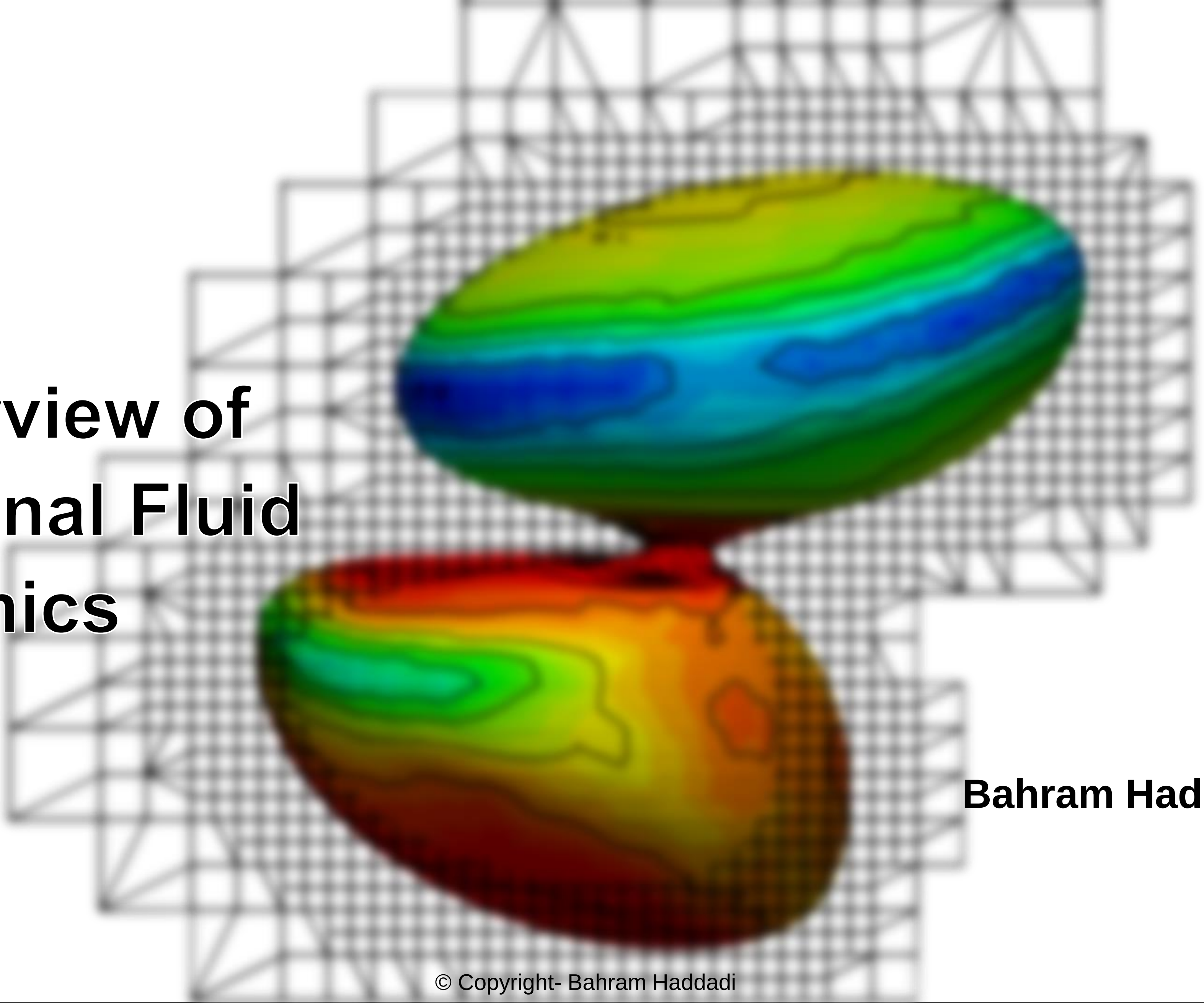
ENGINEERING FLUID DYNAMICS

Brief Overview of Computational Fluid Dynamics



Bahram Haddadi

What is Computational Fluid Dynamics

- Behavior of fluidic systems
  - Experimental investigations
  - Numerical approaches
    - Analytical solutions
    - Numerical representation → CFD
      - Combination of Physics and Numeric
- Computer based analysis of systems
  - fluid flow
  - Heat transfer
  - Mass transfer, …
- Application range
  - Industrial
  - non-industrial

COMPUTATIONAL
FLUID
DYNAMICS

- **Not a tool, but a science**
- The technique is very powerful
  - If combined with fluid dynamics and numerics knowledge

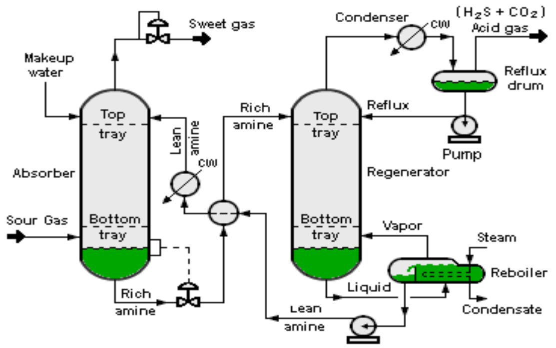
CFD vs. Other Numerical Methods

Details and computational cost

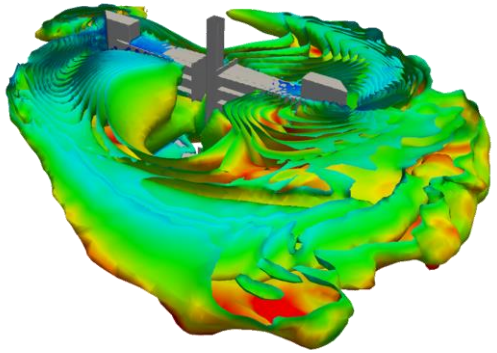
Size of the system

- Black box modeling
  - Just time resolved
  - E.g. balance calculations in a system
  - Very fast but least information provided
- Process simulation
  - Time resolved
  - One space dimension also resolved
  - Resonably fast and moderate details about the system
  - Suitable for complex industial plants and pilots
- Computational fluid dynamics
  - Time resolved
  - Fully space resolved
  - Sutable for detailed systems analysis

https://en.wikipedia.org

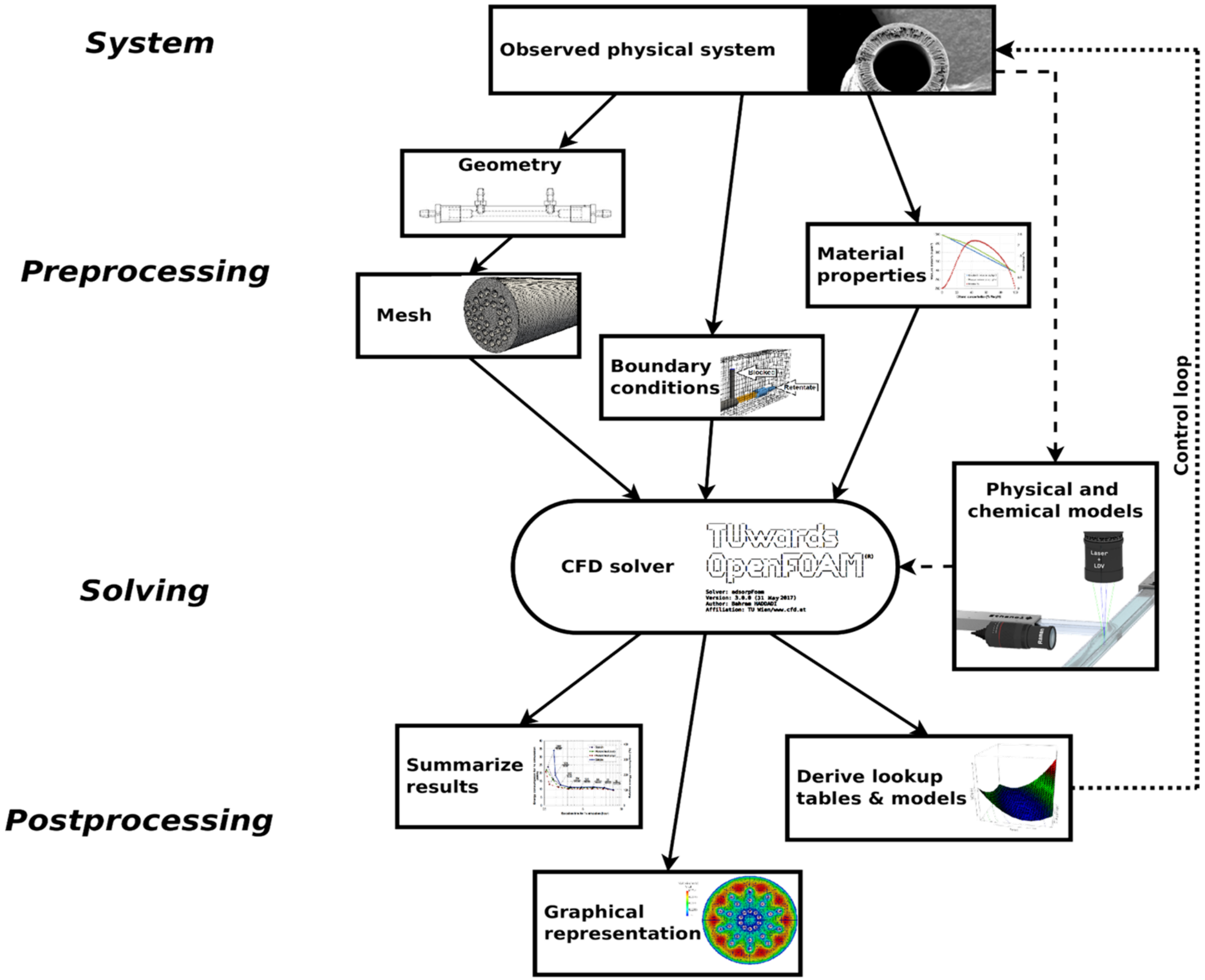


https://en.wikipedia.org



@Bahram Haddadi

CFD Workflow



@Bahram Haddadi

COMPUTATIONAL

EXPERIMENTAL

ENGINEERING FLUID DYNAMICS

CFD Tools

- Common CFD code categories
  - Commercial
    - Annual license cost
    - No/limited access to the code
    - Mostly user friendly interface and official support
    - E.g. Ansys Fluent, Ansys CFX, Star CCM+, COMSOL, …
  - Free
    - No license cost
    - Mainly open-source
    - Missing official support or paid support
    - E.g. OpenFOAM®
- **All are built upon the same concept**

www.comsol.com

www.ansys.com

www.star-ccm.com

www.openfoam.org

OpenFOAM®

- Open Field Operation And Manipulation
  - Original development: 1980s – Imperial College (UK)
  - Open source CFD toolbox – written in C++
  - Operating system:
    - **Linux** – original development and best compatibility
    - **Windows** – using Docker
    - **OSX** – direct compilation or using Docker
  - Different flavours:
    - **ESI version**: openfoam.com – v2212
    - **Foundation version**: openfoam.org – openfoam10
    - **OpenFOAM extend**: Hevoje Jasak – foam-extend-4.1
  - Released for free download (source and binary)
    - Under General Public License

www.openfoam.com

www.openfoam.org

Why OpenFOAM®

- Advantages over commercial codes:
  - Open source and free → No license fee
  - Highly parallel
  - Users can inspect, alter, expand the source code
  - Users have to understand and know what he/she is doing, not just "clicking"!!!
- Advantages over open source codes:
  - One of the most versatile open source CFD programs
  - All standard finite volume algorithms implemented.
  - Industrial interest
  - Large community on the internet: cfd-online.com and stackoverflow.com

@Bahram Haddadi

Mass Conservation – Continiuty Equation

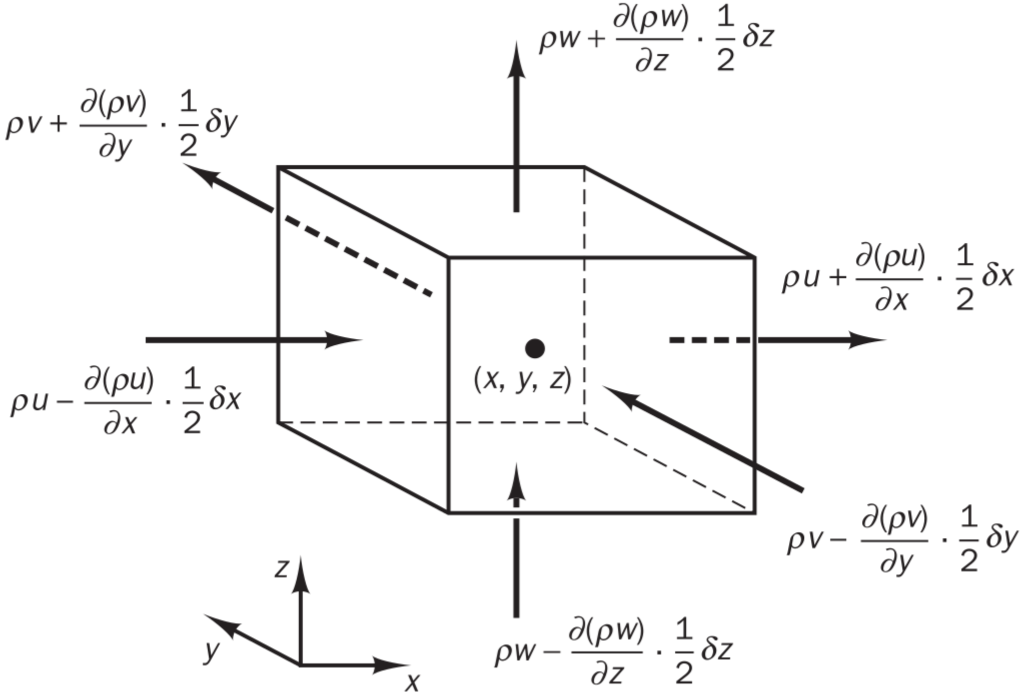
Accumulation = Input - Output

$$\left(\rho u - \frac{\partial(\rho u)}{\partial x}\frac{1}{2}\delta x\right)\delta y\,\delta z - \left(\rho u + \frac{\partial(\rho u)}{\partial x}\frac{1}{2}\delta x\right)\delta y\,\delta z$$

$$+\left(\rho v - \frac{\partial(\rho v)}{\partial y}\frac{1}{2}\delta y\right)\delta x\,\delta z - \left(\rho v + \frac{\partial(\rho v)}{\partial y}\frac{1}{2}\delta y\right)\delta x\,\delta z$$

$$+\left(\rho w - \frac{\partial(\rho w)}{\partial z}\frac{1}{2}\delta z\right)\delta x\,\delta y - \left(\rho w + \frac{\partial(\rho w)}{\partial z}\frac{1}{2}\delta z\right)\delta x\,\delta y$$

$$= \frac{\partial \rho}{\partial t}\delta x\,\delta y\,\delta z$$



$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad \Rightarrow \quad \frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0$$

An introduction to computational fluid dynamics – The finite volume method – 2nd Edition, Versteeg and Malalasekera 2007

Governing Equations

Continuity	$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0$
x-momentum	$\frac{\partial (\rho u)}{\partial t} + \mathrm{div}(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + \mathrm{div}(\mu \text{ grad } u) + S_{Mx}$
y-momentum	$\frac{\partial (\rho v)}{\partial t} + \mathrm{div}(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + \mathrm{div}(\mu \text{ grad } v) + S_{My}$
z-momentum	$\frac{\partial (\rho w)}{\partial t} + \mathrm{div}(\rho w \mathbf{u}) = -\frac{\partial p}{\partial z} + \mathrm{div}(\mu \text{ grad } w) + S_{Mz}$
Energy	$\frac{\partial (\rho i)}{\partial t} + \mathrm{div}(\rho i \mathbf{u}) = -p \text{ div } \mathbf{u} + \mathrm{div}(k \text{ grad } T) + \Phi + S_i$
Equations of state	$p = p(\rho, T)$ and $i = i(\rho, T)$
	e.g. perfect gas $p = \rho R T$ and $i = C_v T$

$$\frac{\partial (\rho \varphi)}{\partial t}$$

Time derivative

$$\nabla \cdot (\rho \varphi \boldsymbol{u})$$

Convection term

$$\nabla \cdot (\Gamma \nabla \varphi)$$

Diffusion term

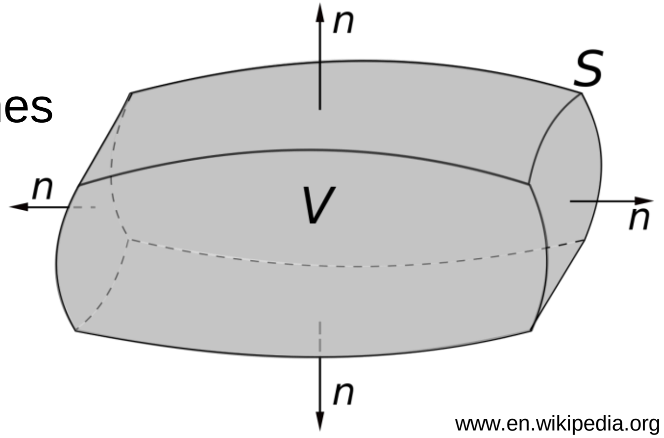
$$S_\varphi$$

Source term

An introduction to computational fluid dynamics – The finite volume method – 2nd Edition, Versteeg and Malalasekera 2007

Finite Volume Approach

- **Mesh (grid):** Converting the domain into discrete domains
- **Grid cell:** The small volume surrounds each node of the mesh
- **Key step:** integration of the transport equation over a three-dimensional control volume
- **Gauss divergence theorem:** Replacing volume integral of the divergence term by surface integral
  - Terms evaluated as fluxes at the surfaces
  - Ensures the conservation of fluxes entering and exiting the grid
  - Allows for easy formulation of the balances on unstructured meshes



www.en.wikipedia.org

- Time-dependency: integration with respect to time

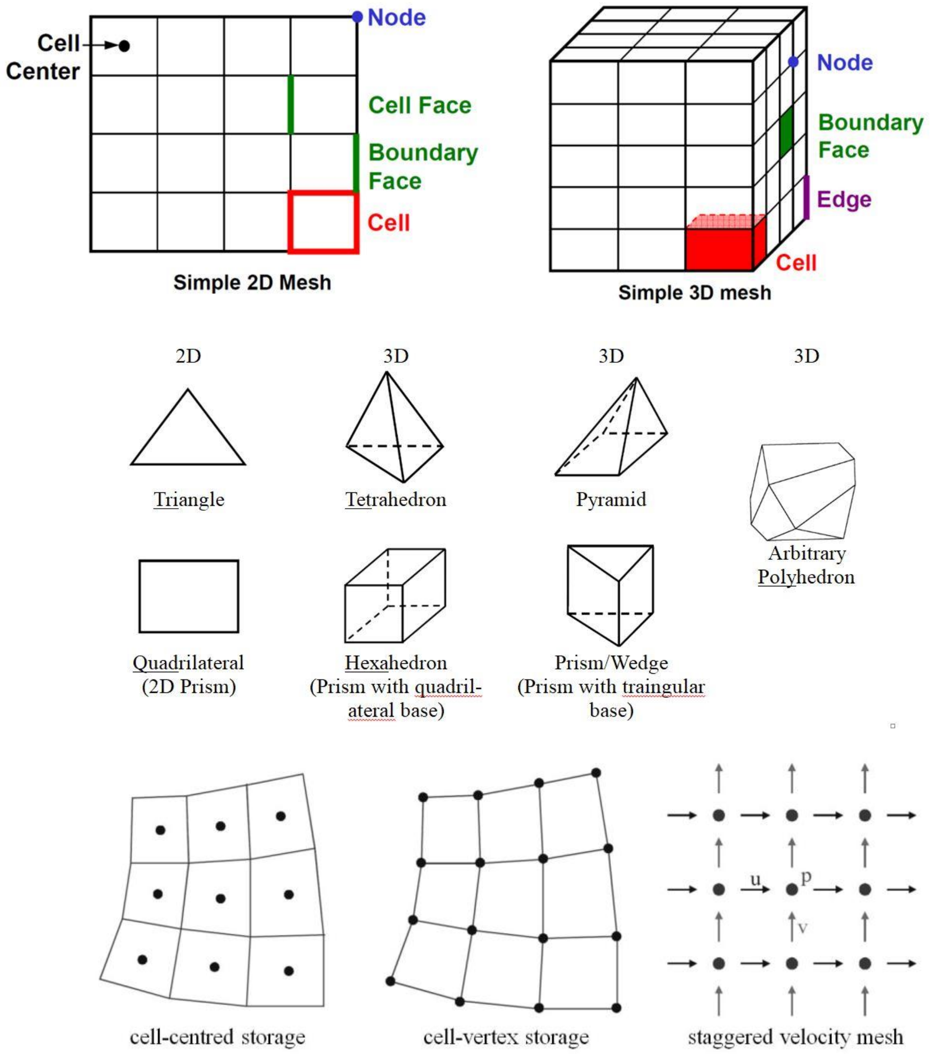
$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\varphi\boldsymbol{u}) = \nabla \cdot (\Gamma \nabla \varphi) + S_\varphi$$

$$\int_{\Delta t} \frac{\partial}{\partial t}\left(\int_{CV} \rho\varphi \, dV\right) dt + \int_{\Delta t} \int_{A} \boldsymbol{n} \cdot (\rho\varphi\boldsymbol{u}) \, dA dt = \int_{\Delta t} \int_{A} \boldsymbol{n} \cdot (\Gamma \nabla \varphi) \, dA dt + \int_{\Delta t} \int_{CV} S_\varphi dV \, dt$$

An introduction to computational fluid dynamics – The finite volume method – 2nd Edition, Versteeg and Malalasekera 2007

Mesh Terminology

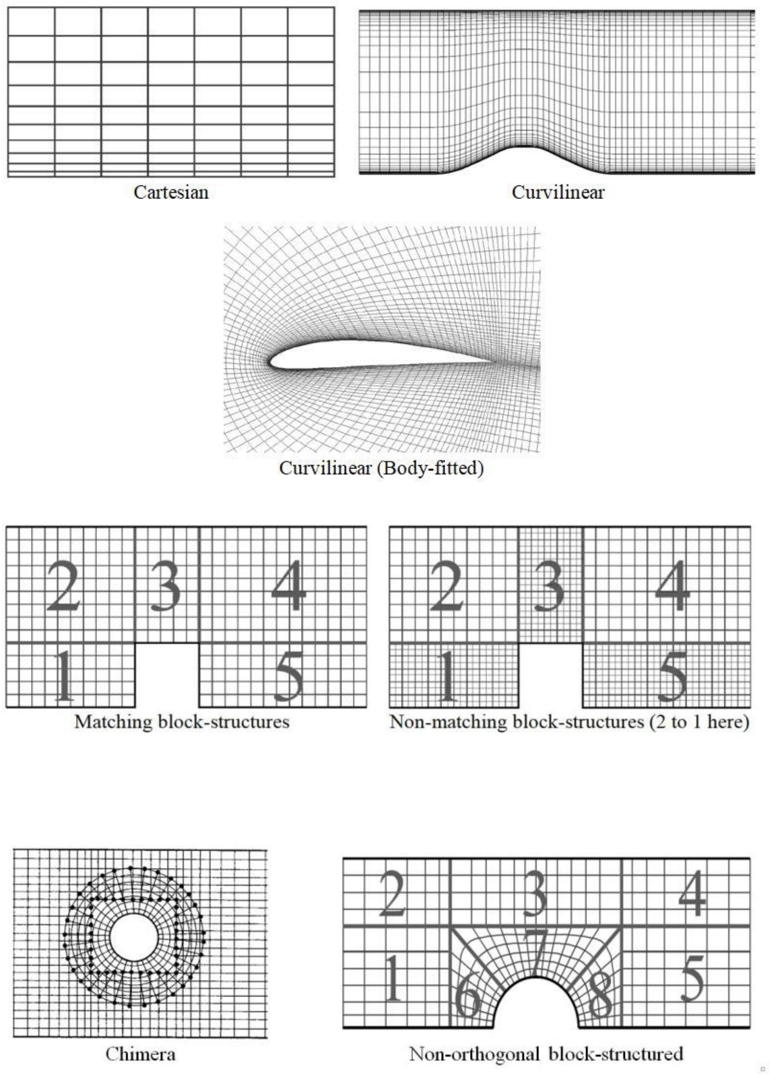
- Cell: control volume created by domain discretization
- Node: grid point
- Cell center: center of a cell
- Edge: boundary of a face
- Face: boundary of a cell
  - Internal
  - Boundary
- Zone: grouping of nodes, faces, cells
- Domain: group of node, face and cell zones
- Nodes positions relative to the vertices
  - Cell centered
  - Cell vertex
  - Staggered



www.manchestercfd.co.uk/post/all-there-is-to-know-about-different-mesh-types-in-cfd

Mesh Types

- **Structured Grids**
  - Cartesian
    - Grid lines are always parallel to the coordinate axes
  - Curvilinear
    - Coordinate surfaces are curved to fit boundaries
  - *Orthogonality: all grid lines cross at 90º*
- **Block-structured Grids**
  - Matching
    - The grid on the common boundaries between the regions is the same
  - Non-matching
  - Chimera (overset)

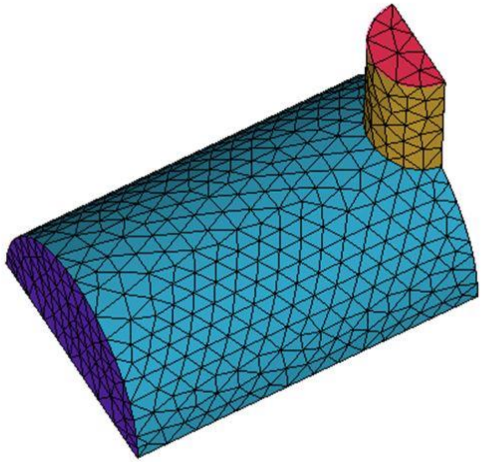


www.manchestercfd.co.uk/post/all-there-is-to-know-about-different-mesh-types-in-cfd

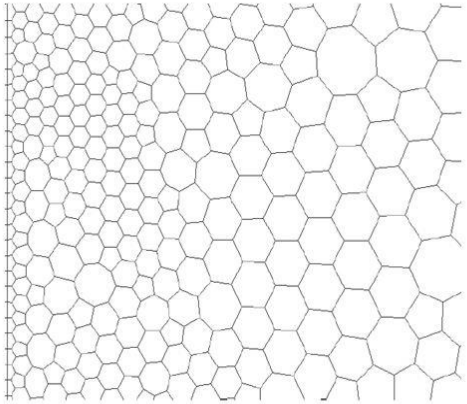
Mesh Types

- **Unstructured Grids**
  - Triangular (tetrahedral)
    - Robust meshing algorithms
  - Quadrilateral (hexahedral)
    - Best for CFD calculations
  - Polygon (polyhedral)
    - Similar to tet, with less computational overhead
  - Hybrid

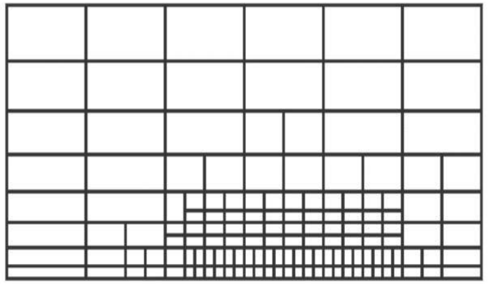
- **Quad/Hex advantages vs Tri/Tet**
  - Higher quality solutions for flow-aligned problems
  - Fewer cells/nodes than a comparable tri/tet mesh.
  - Reduced numerical diffusion



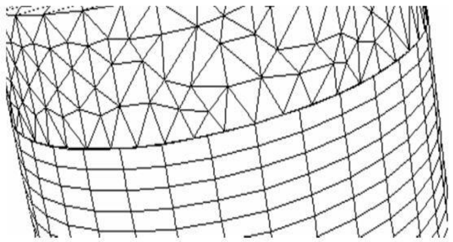
Triangular/tetrahedral



Polygon (polyhedral)



Quadrilateral (hexahedral)

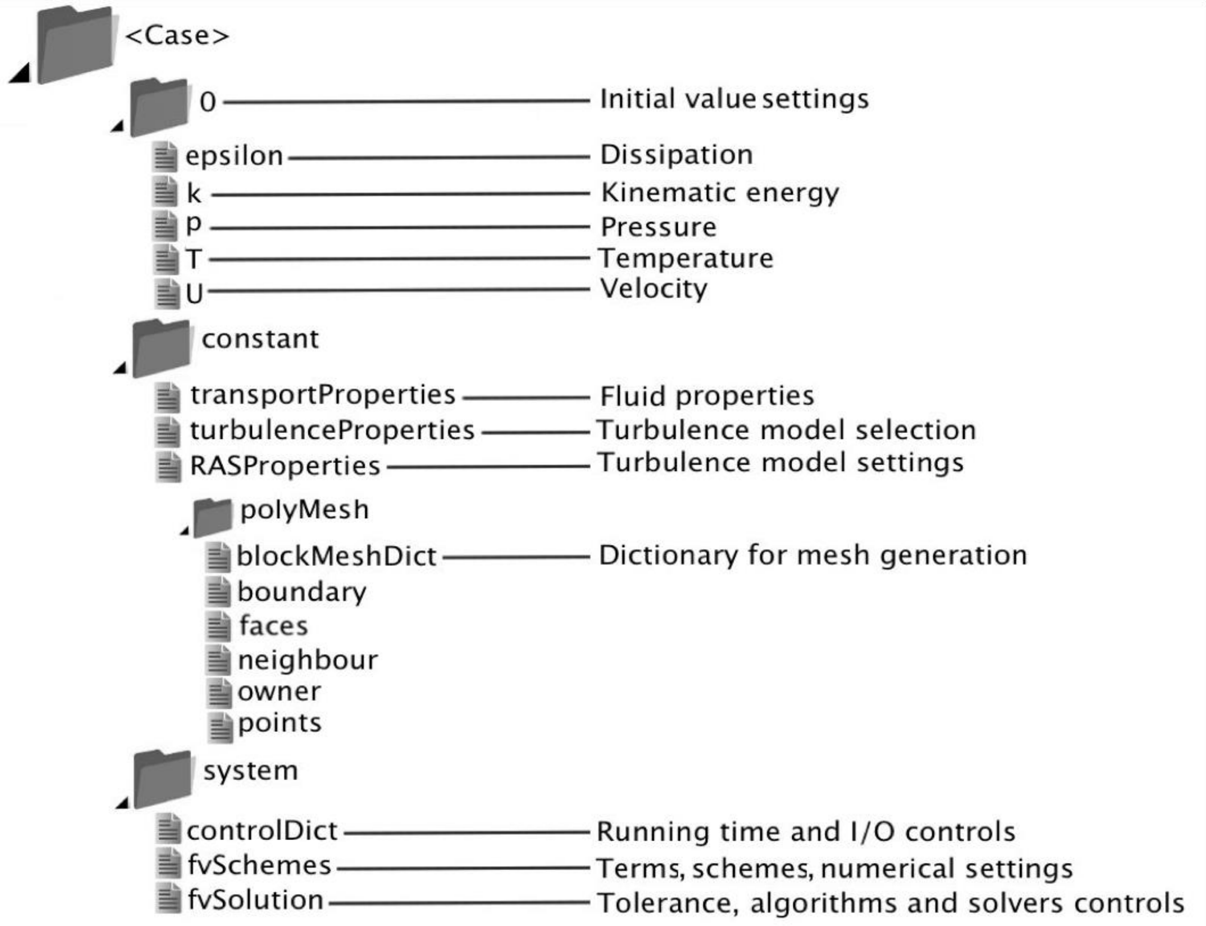


Hybrid Tetrahedral

www.manchestercfd.co.uk/post/all-there-is-to-know-about-different-mesh-types-in-cfd

OpenFOAM® Applications and Case Structure

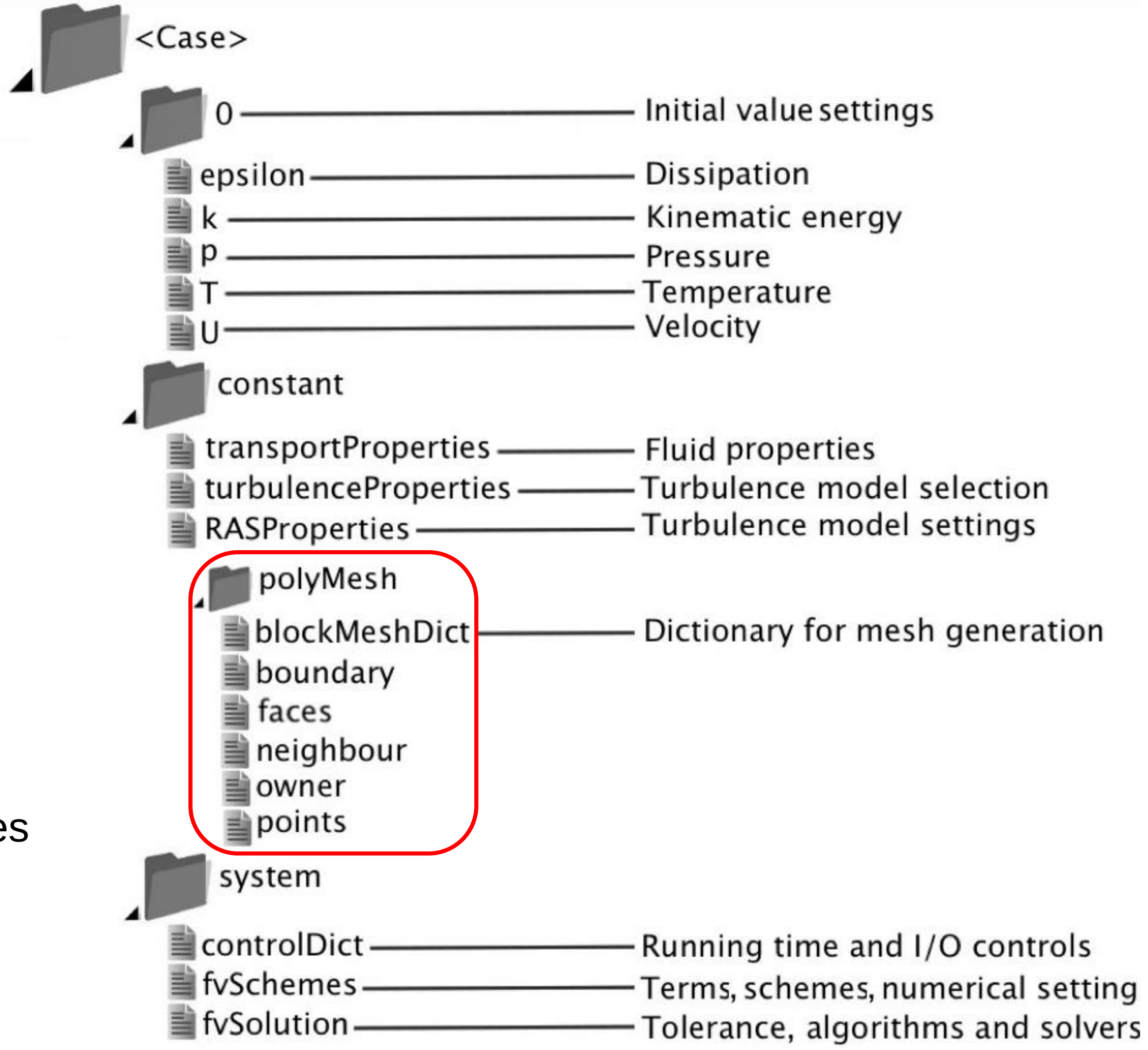
- Solvers
  - Solving a specific continuum mechanics problem
  - icoFoam
  - simpleFoam
  - …
- Utilities
  - Performing pre and post processing
  - Mesh preparation: blockMesh, snappyHexMesh
  - Simulation set-up: topoSet, setFields
  - Processing the results: postProcess
  - …



@Bahram Haddadi

Mesh in OpenFOAM®

- Unstructured mesh design
  - Capable of handling structured
- Basic mesh generation
  - *blockMesh*
  - Block structured meshes
  - Curved internal and external boundaries.
  - The mesh is setup in a script (no GUI)
  - Good for simple geometries
- Advanced mesh generation
  - *snappyHexMesh, cfMesh*
  - Automatic mesh generator for complex geometries
  - Possibility of mesh refinement at desired regions
  - Parallel
- Mesh import
  - `gambitToFoam, ansysToFoam, cfxToFoam,` …



@Bahram Haddadi

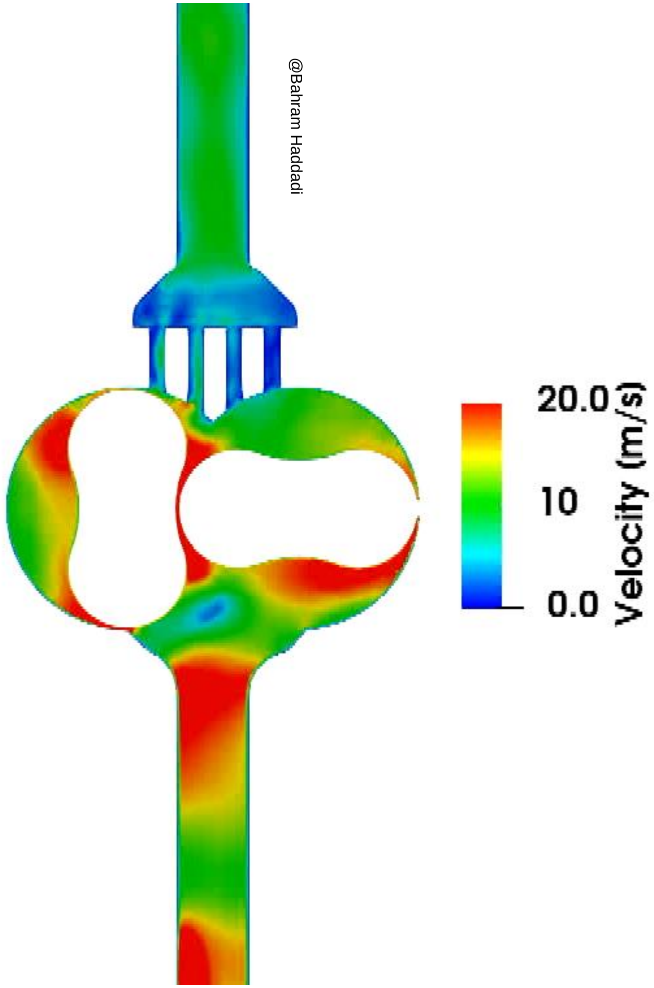
Initial Conditions

- Starting value for the solver and once specified
  - Transient simulations
    - Initial state of the system
    - Next time step values are calculated based on these
  - Steady state simulations
    - Initial guess for the numerical system
    - Initial values will be replaced by newly calculated values
    - Better the initial values → the faster the convergence
- Value is assigned to the center of every cell
  - A uniform value for the whole field
  - Individual value per cell
    - Patching fields
    - OpenFOAM®: *setFields*
    - *Multi-solver solutions*

@Bahram Haddadi

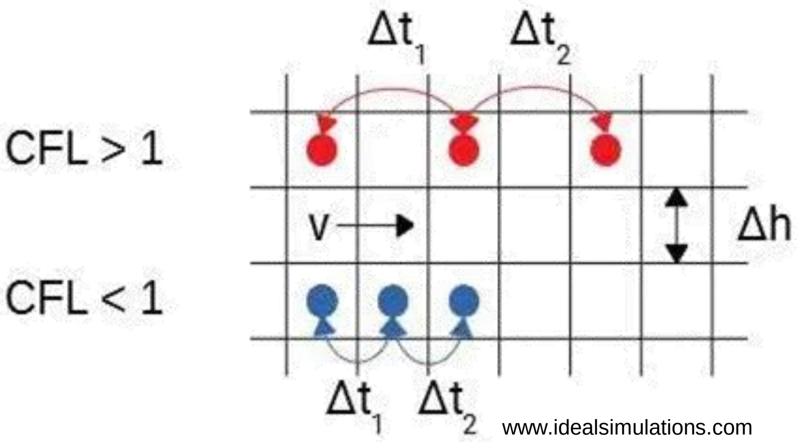
Boundary Conditions

- Boundary conditions will connect the simulation domain with its surroundings
  - Each property interaction with outside the domain
- The values specified are located at the boundary faces of the domain
- Three main types of boundary conditions
  - **Dirichlet**: fixed value on the boundary
  - **Neumann**: fixed gradient on the boundary
  - **Mixed**: combined fixed value and gradient
- Most boundary conditions are either steady state or transient
- Ill-defined boundary conditions
  - Non-convergence
  - Incorrect results



Time Stepping and CFL condition

- Time step
  - How much the information travels across a computational grid cell
  - Too big time step
    - Information propagates through more than one grid cell
    - Numerically: Inaccurate solution - divergence
    - Physically: Nonphysical results
  - Too small time step
    - Computationally expensive



- Information transport should not “overtake” the physical transport
  - Courant-Friedrichs-Lewy (CFL) condition – Co
    - One-dimensional: $\mathrm{Co} = \mathrm{u}\frac{\Delta t}{\Delta x}$
    - Less or equal to one
    - Finer mesh → smaller time step
    - Initial time step: maximum velocity and smallest mesh size

Discretization

- General transport equation

$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\varphi\boldsymbol{u}) = \nabla \cdot (\Gamma\nabla\varphi) + S_\varphi$$

- Time derivative
  - Integrating over volume
  - Euler implicit scheme

$$\int_V \frac{\partial\rho\varphi}{\partial t} dV \approx \frac{\rho_P^n\varphi_P^n - \rho_P^0\varphi_P^0}{\Delta t} V_P$$

- Convection term
  - Integrating over volume
  - Applying Gauss‘s theorem
  - φ_f to be calculated → schemes

$$\int_A \boldsymbol{n} \cdot (\rho\varphi\boldsymbol{u})\, dA \approx \sum_f \boldsymbol{n} \cdot (A\rho\boldsymbol{u})_f \varphi_f = \sum_f F\varphi_f$$

- Diffusion term
  - Similar to convection term
  - $\nabla_f\varphi$ gradient at the face
  - Second order accurate → d orthogonal between P and N

$$\int_A \boldsymbol{n} \cdot (\Gamma\nabla\varphi)\, dA = \sum_f \Gamma_f(\boldsymbol{n} \cdot \nabla_f\varphi)A_f$$

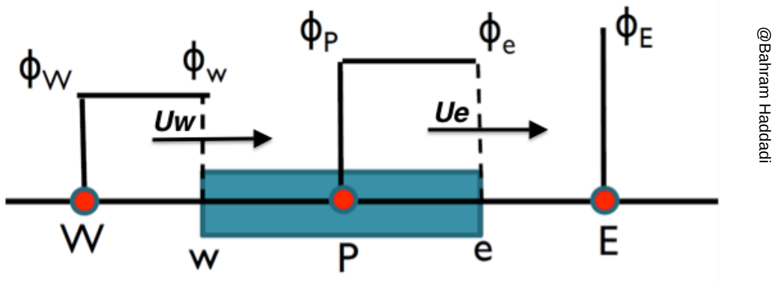
$$\boldsymbol{n} \cdot \nabla_f\varphi = \frac{\varphi_N - \varphi_P}{|\boldsymbol{d}|}$$

Common Discritization Schemes

- **First Order Upwind**

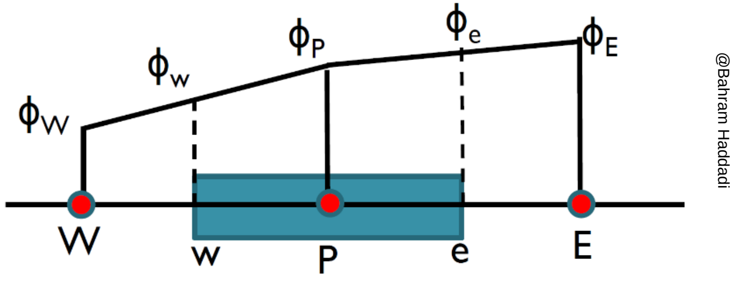
$$\varphi_e = \varphi_P \qquad if, F_e > 0$$

$$\varphi_e = \varphi_E \qquad if, F_e < 0$$



- **Central Differencing Scheme**

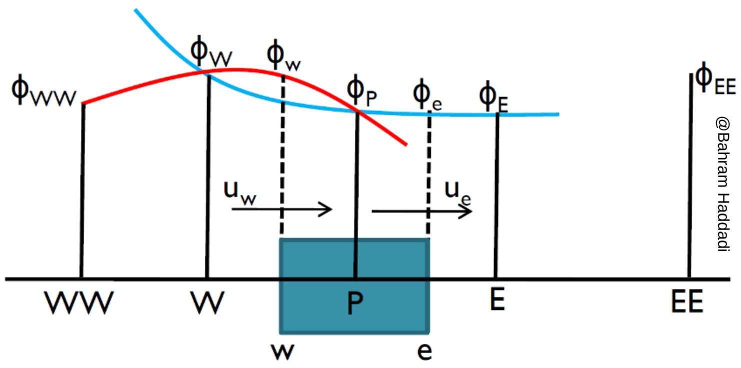
$$\varphi_e = \frac{\varphi_E + \varphi_P}{2}, \qquad \varphi_w = \frac{\varphi_P + \varphi_W}{2}$$



- **Quadratic Upwind Interpolation for Convective Kinetics (QUICK)**

$$\text{When } F_e > 0, \qquad \varphi_e = \frac{6}{8}\varphi_P + \frac{3}{8}\varphi_E - \frac{1}{8}\varphi_W$$

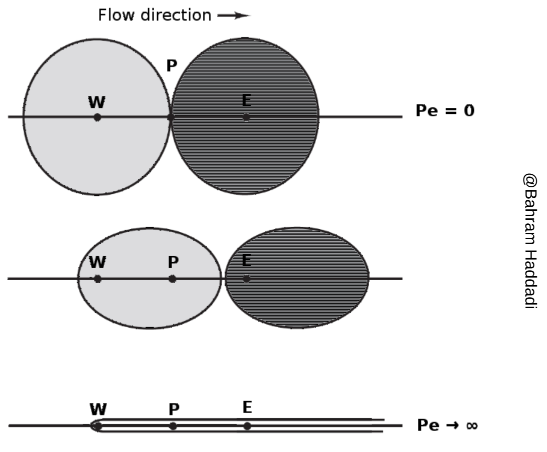
$$\text{When } F_w > 0, \qquad \varphi_w = \frac{6}{8}\varphi_W + \frac{3}{8}\varphi_P - \frac{1}{8}\varphi_{WW}$$



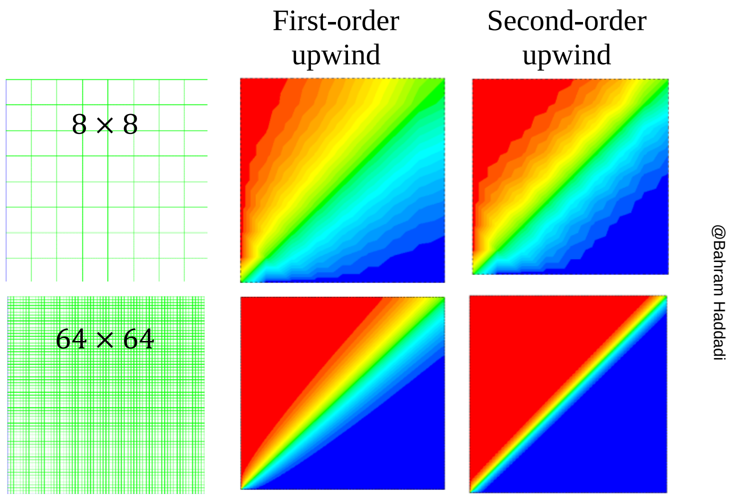
Discritization Schemes Properties

Prerequisite for results to be physically realistic

- **Conservativeness:** flux across a certain face must be equal the adjacent control volume flux through the same face
- **Boundedness:** ensures the solution remains within certain bound limits and guarantees the solution does not exhibit unphysical or unrealistic behavior
- **Transportiveness:** Peclet number, Pe. It measures the relative strengths of convection, N_{conv} and diffusion, N_{diff}.



> ***False diffusion:*** a multidimensional phenomenon and it occurs when the flow is not perpendicular to the grid lines. It is a numerically introduced diffusion and arises in convection dominated flows

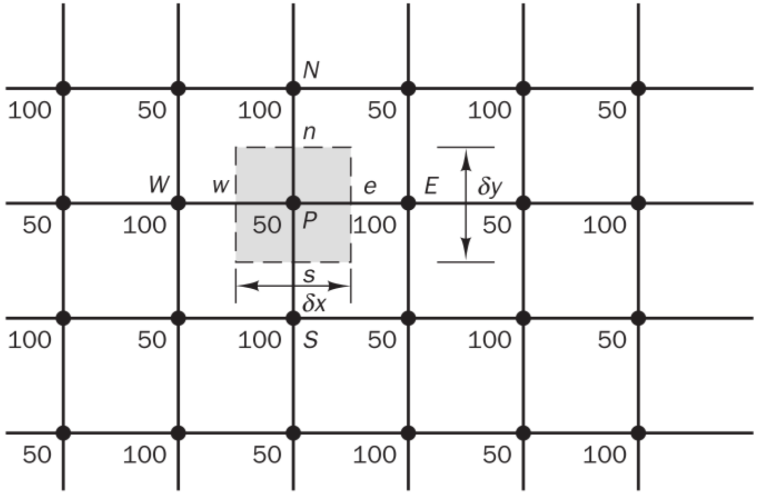


Discritization Schemes Assessment

Scheme	Conser-vative	Bounded	Accuracy	Trans-portive	Remarks
Upwind	Yes	Unconditionally bounded	First order	Yes	Include false diffusion if the velocity vector is not parallel to one of the coordinate directions
Central Differencing	Yes	Conditionally bounded*	Second order	No	Unrealistic solutions at large Pe number
QUICK	Yes	Unconditionally bounded	Third order	Yes	Less computationally stable. Can give small undershoots and overshoots

Variables Storage

- Variables evaluation
  - Checker-board problem
    - The cell pressure is not considered in the pressure gradient
    - Using Staggered grid
      - Scalar
        - Pressure, temperature, density etc.
        - At ordinary nodal points
      - Vectors
        - Velocity
        - At **Staggered** grid, centered around cell faces
  - Generation of velocities at exactly the locations where they are needed



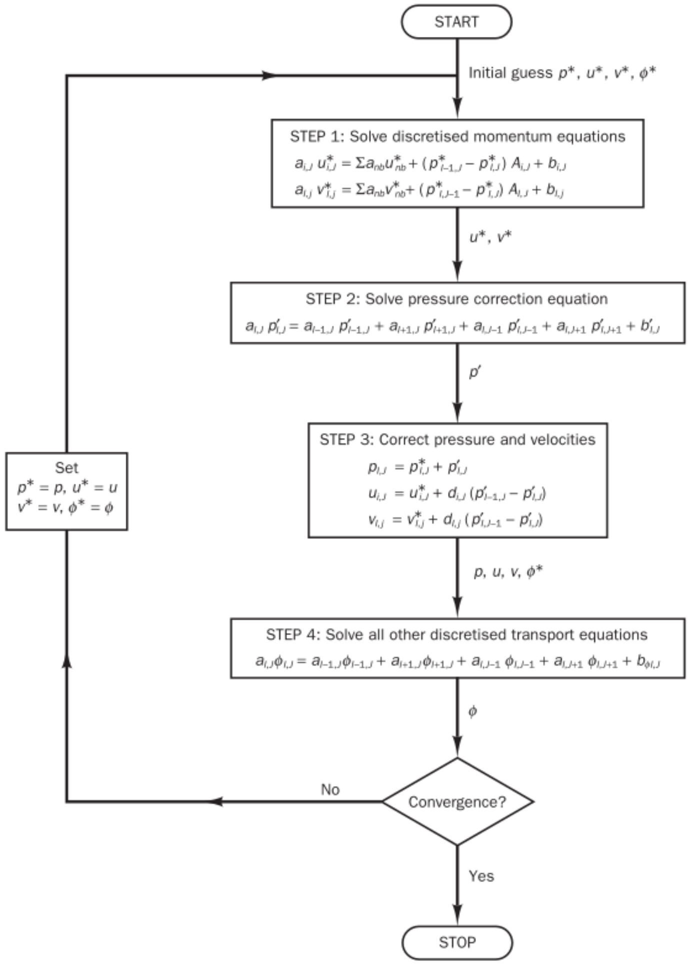
$$\frac{\partial p}{\partial x} = \frac{p_e - p_w}{\delta x} = \frac{\left(\frac{p_E + p_P}{2}\right) - \left(\frac{p_P + p_W}{2}\right)}{\delta x}$$

$$= \frac{p_E - p_W}{2\delta x}$$

An introduction to computational fluid dynamics, Versteeg and Malalasekera, 2007

Pressure – Velocity Coupling Algorithms

- Semi-Implicit Method for Pressure Linked Equations (SIMPLE)
  - Originally for steady state problems
  - A guess-and-correct procedure
  - SIMPLER (SIMPLE Revised) and SIMPLEC (SIMPLE Consistent)
- Pressure Implicit with Splitting of Operators (PISO)
  - Originally for unsteady compressible flows
  - Non-iterative: using one predictor and two corrector steps
- PIMPLE
  - Hybrid SIMPLE/PISO
  - PISO internal loop with SIMPLE External loops
  - Higher stability and reliability for bigger time steps
- Coupled algorithm
  - Discritized pressure and velocity equations are solved in a single matrix
  - Implicit coupling between pressure and velocity
  - Computationally more expensive
  - More stable for low quality meshes or large time steps



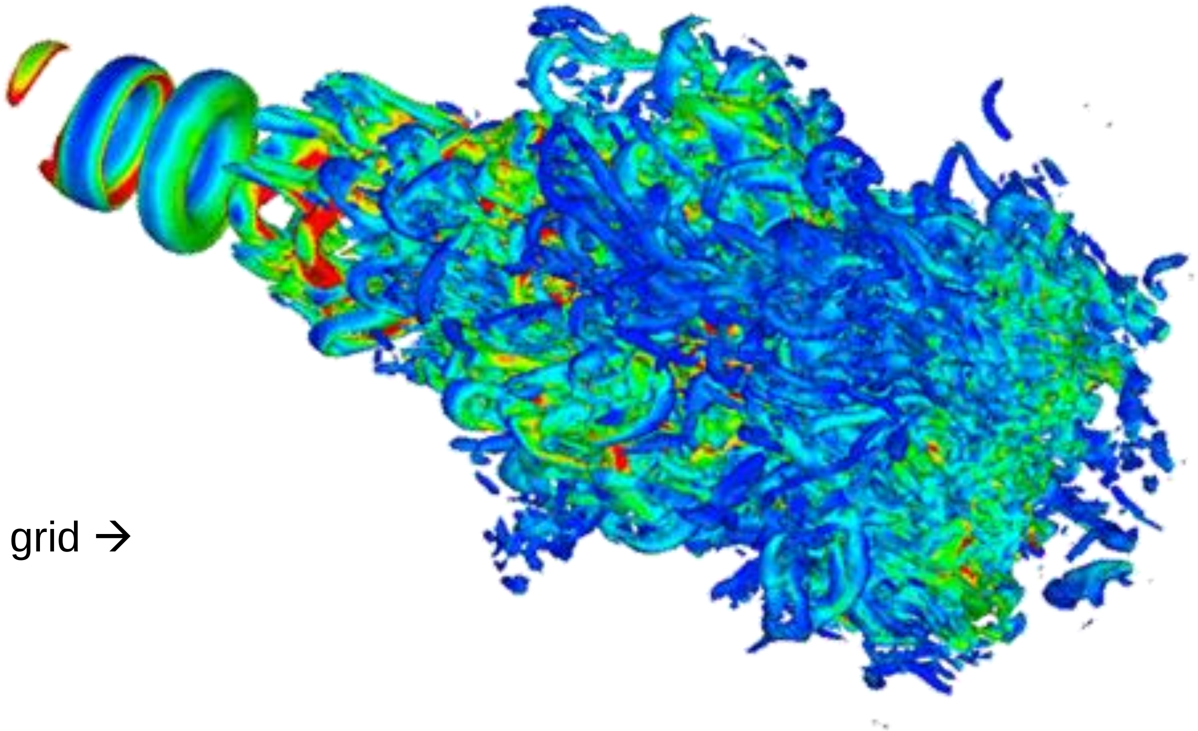
An introduction to computational fluid dynamics, Versteeg and Malalasekera, 2007

Models – Turbulence

- Many engineering applications are turbulent
- Turbulence
  - Reynolds number (Re)
  - Highly transient phenomenon
  - Characterized by a wide range of eddy sizes
  - Fully resolve these eddies numerically
    - Obtain a full profile of the turbulent flow field
    - Computationally very expensive
    - Hence we require a turbulence model

- Turbulence modeling
  - Important feature is averaging
  - Scales of the flow that are not resolved by the grid → models need to be applied

- Turbulence resolving
  - Reynolds Average Navier Stokes (RANS)-based models
  - Large eddy simulations (LES)
  - Direct Numerical Simulation (DNS)



http://home.iitk.ac.in/~aksaha/QIP_Comp_Turb/background.htm

Models – Turbulence – RANS

- Any property can be written as the sum of an average and a fluctuation → **Reynolds decomposition**
  $\tilde{\varphi}$ = Φ + φ
- *The average of the fluctuating component is identically zero*
  - Conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \tilde{\boldsymbol{u}}) = 0 \quad \Rightarrow \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0$$

  - Conservation of momentum (Navier-Stokes equation)

$$\frac{\partial (\rho \tilde{u}_i)}{\partial t} + \nabla \cdot (\rho \tilde{u}_i \tilde{\boldsymbol{u}}) = -\frac{\partial \tilde{p}}{\partial x_i} + \nabla \cdot (\mu \nabla \tilde{u}_i) + \tilde{S}_{Mi} \quad \Rightarrow \quad \frac{\partial (\rho \mathrm{U}_i)}{\partial t} + \nabla \cdot (\rho \mathrm{U}_i \mathbf{U}) = -\frac{\partial P}{\partial x_i} + \nabla \cdot (\mu \mathrm{U}_i) - \left(\frac{\partial (\rho \overline{u u_i})}{\partial x} + \frac{\partial (\rho \overline{v u_i})}{\partial y} + \frac{\partial (\rho \overline{w u_i})}{\partial z} \right) + \mathrm{S}_{Mi}$$

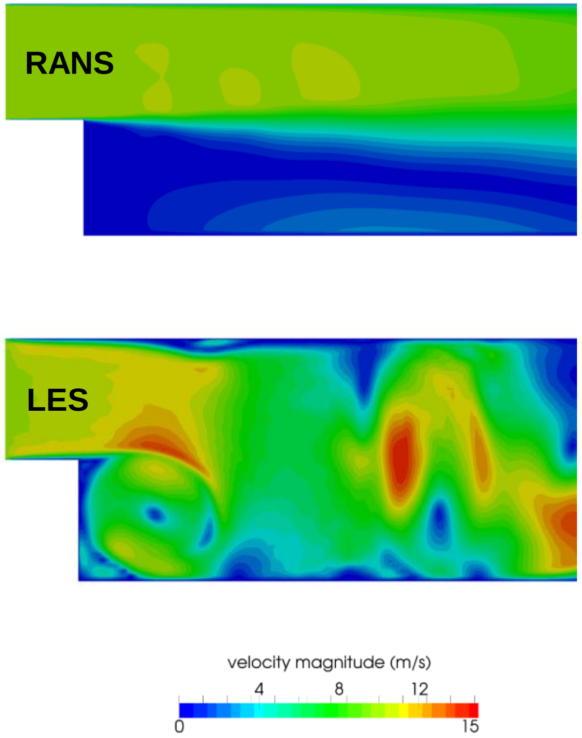
  - Conservation of passive scalars (given a scalar $\tilde{e}$)

$$\frac{\partial (\rho \tilde{e})}{\partial t} + \nabla \cdot (\rho \tilde{e} \tilde{\boldsymbol{u}}) = \nabla \cdot \left(k \nabla \tilde{T} \right) + \tilde{S}_e \quad \Rightarrow \quad \frac{\partial (\rho \mathrm{E})}{\partial t} + \nabla \cdot (\rho \mathrm{EU}) = \nabla \cdot (k \nabla \mathrm{T}) - \left(\frac{\partial (\rho \overline{u e})}{\partial x} + \frac{\partial (\rho \overline{v e})}{\partial y} + \frac{\partial (\rho \overline{w e})}{\partial z} \right) + \mathrm{S}_e$$

- New unknowns
  - 6 turbulent stresses ($\rho\overline{uu}$, $\rho\overline{vu}$, $\rho\overline{wu}$, $\rho\overline{uv}$, $\rho\overline{vv}$, $\rho\overline{wv}$, $\rho\overline{uw}$, $\rho\overline{vw}$, $\rho\overline{ww}$)
  - 3 turbulent fluxes ($\rho\overline{ue}$, $\rho\overline{ve}$, $\rho\overline{we}$)
  - Additional equations based on empirical observations →**Turbulence Modeling** (e.g. k-ε, k-ω, ...)
  - Using PDE’s for the turbulent stresses and fluxes
- Suitable for steady-state problems

Models – Turbulence – LES

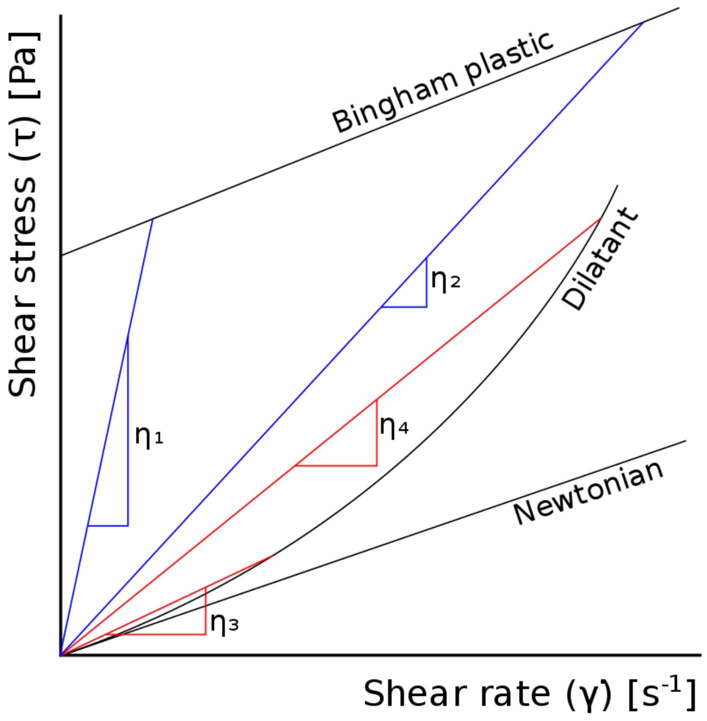
- Physical representation
  - large eddies of the flow are dependent on the geometry
  - The smaller eddies are more universal
- Mathematical approach
  - Velocity field separated into a resolved and sub-grid part using a filter function
    - Convolution of a function with a filtering kernel
  - Large eddies explicitly resolved by the grid
  - Small eddies handled implicitly by sub grid-scale model (SGS)
- most practical (and commercial) implementations of LES use the grid itself
  - No explicit filtering is needed
- Sub grid-scale turbulence models usually employ the Boussinesq hypothesis
  - Turbulent stresses are related to the mean velocity gradients



@Bahram Haddadi

Models – Viscosity

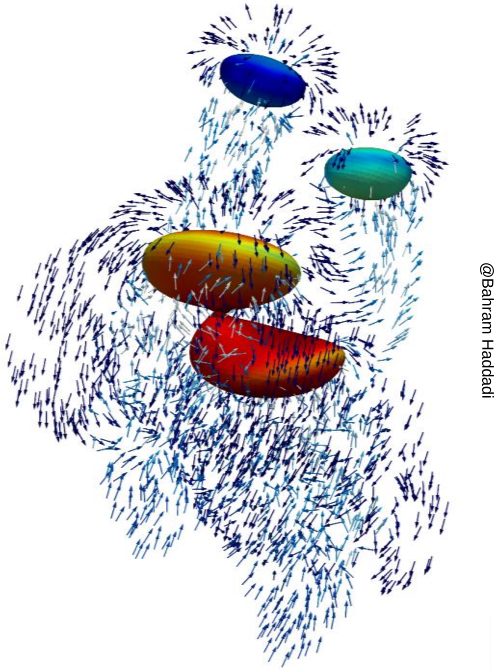
- Viscosity
  - Intensive property of a fluid
  - Fluid internal resistance to motion or deformation
  - Viscous fluids are less willing to flow than the less viscous fluids
- Newton’s law of viscosity
  - Relationship between a fluid’s shear stress and shear rate when subjected to mechanical stress
- Types of viscosity
  - Dynamic (absolute) viscosity: fluid’s internal resistance to flow
  - Kinematic viscosity: ratio of dynamic viscosity to density
  - Apparent (steady shear) viscosity: shear stress ratio to shear rate
- Fluids
  - Newtonian
  - Non-Newtonian
    - Viscosity not constant (e.g. shear-rate dependent)

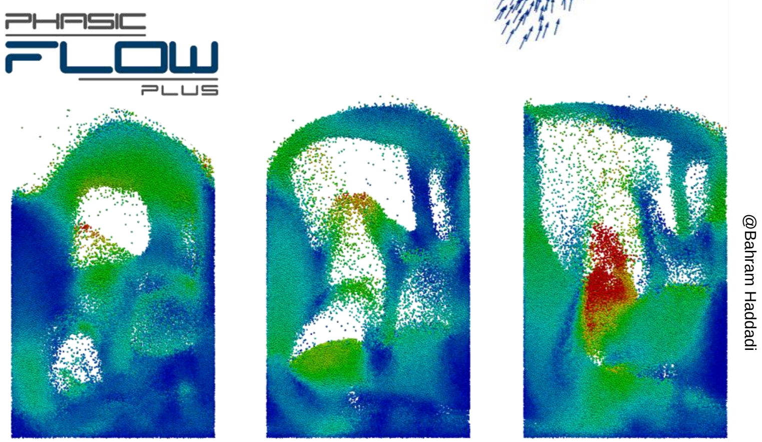


https://en.wikipedia.org/wiki/Apparent_viscosity

Models – Multiphase

- Simultaneous flow of materials in different phases
  - Multiple components can be present per phase.
  - gas-liquid, gas-solid, liquid-solid, liquid-liquid and three-phase flows
- Phases (visual appearance)
  - Separated: the boundary between phases is described in detail
  - Mixed: dispersed particles as well as semi-continuous interfaces exist together
  - Dispersed: one phase is dispersed in a continuous phase
- Modeling approaches
  - Lagrangian
    - Tracking individual point particles during their movement
    - More suitable for dispersed configuration
    - Studying particle flows, e.g. in Discrete Element Method (DEM)
  - Eulerian
    - Observing fluid behavior in a given control volume
    - More suitable for fluid-fluid multiphase flows
    - CFD approaches such as Euler-Euler and Volume of Fluid

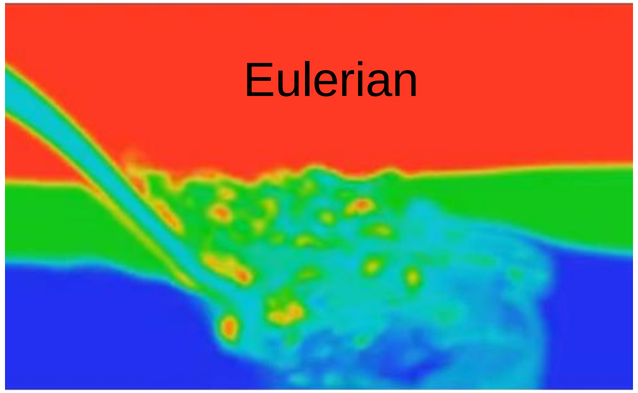


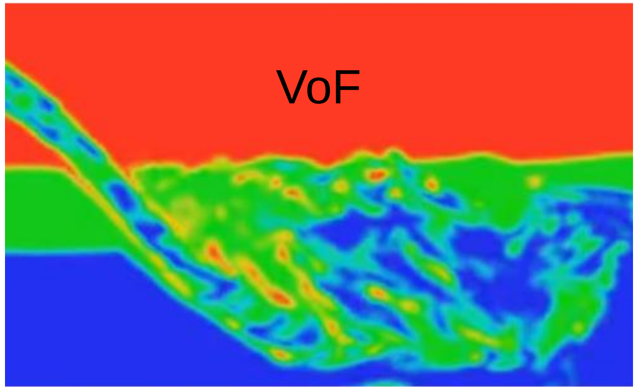


Models – Multiphase – Eulerian

- Euler-Euler approach (Multi-fluid model)
  - All phases are treated as continuous
  - The phases interact through the drag and lift forces
  - Concept of phasic volume fractions
    - Continuous functions of space and time
    - Their sum equal to one
    - A transport equation for the volume fraction is solved
  - Individual conservation equations solved per phase

- Volume of Fluid (VoF) method
  - Fraction function (C)
    - C=1, the control volume is completely filled with the chosen phase
    - C=0, the control volume is filled with a different phase
    - 0<C<1, the interface between phases is present inside the control volume
  - The flow domain is modeled on a fine grid
    - The interface to be resolved
  - To track the interface between phases:
    - All field variables are shared between the phases
    - The transport equations are solved for mixture properties
    - An advection equation for the fraction function C is solved



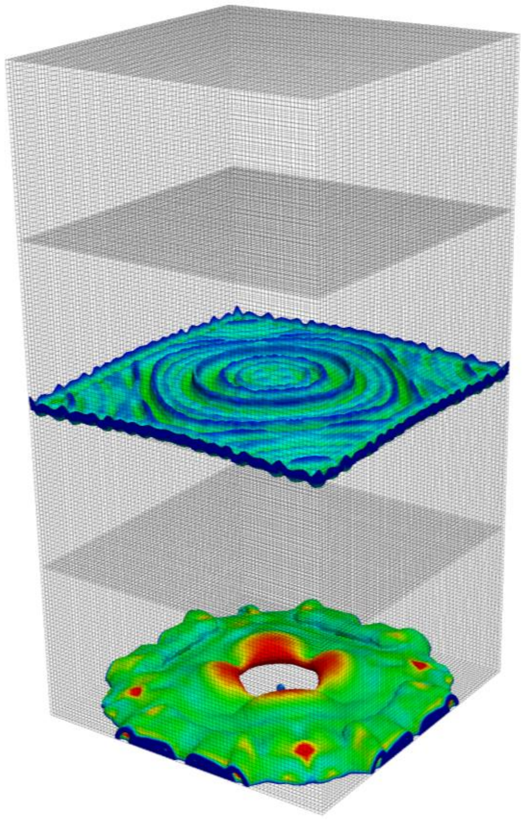


Clemens Lischka, www.youtube.com

Parallel Processing

- Simultaneous use of more than one processor
  - CPU
  - GPU
- Save time and cost
- Computational load will be distributed among processors
  - domain
  - calculations
- Load distribution
  - Shared memory: the whole system seen as a single computer
  - Distributed memory: individual computers connected through network each seen as a computational node

	Shared Memory Multiprocessor	Distributed Memory Multicomputer
Memory	Data is saved in a global memory that can be accessed by all processors	Each computer has a local memory and a processor can only access its local memory
Data transfer between processors	The sender processor simply needs to write the data in a global variable and the receiver can read it	Message is sent explicitly from one computer to another using a message passing library, e.g. Message Passing Interface (MPI)



@Bahram Haddadi