CFD for Aerospace Engineers

CFD

Computational Fluid Dynamics

By: Miguel Angel Saez Ortuño

Check for these and practice if possible

Programs Used:

- ANSYS - CFX and ICEM

History:

1st Man Richardson in 1922 → Initial conditions are most important.

Prandtl in 1925 → Turbulence model

Why using CFD?

Experiments

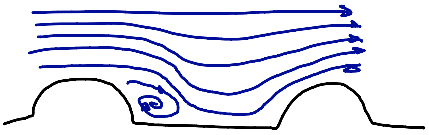
Experiments can be achieved with similar conditions but sometimes is too expensive or too dangerous.

Same conditions only mater if Re and Ma are the same

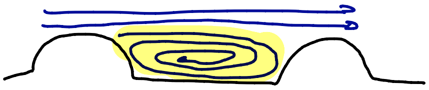
CFD Colors For Directors

- All qualities can be extracted without error, but there is uncertainty
- No measurement errors, cheap, hazard free
- Sometimes XFOIL is better or analytical. Choose wisely.
- RANS can not predict all the flows

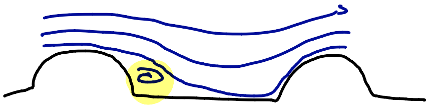
LES GOOD ONE



RANS

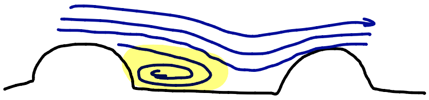


$k-\omega$ MODEL



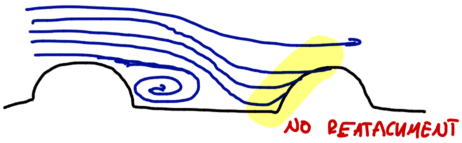
too small separation

$k-\varepsilon$ MODEL



too big separation

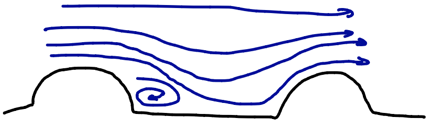
SPALAR-ALLMARAS



NO REATACHMENT

combining both

MENTER



Basic Equations and Numbers

Properties of Fluids

- Fluids deform under the influence of shear forces

The deformation with respect to the initial state is unbounded and the deformation velocity is a functional of the shear forces

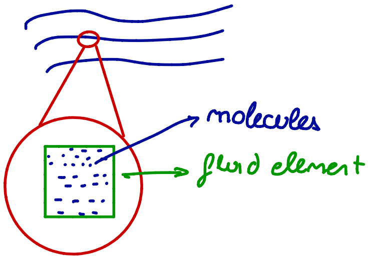
⟶ function of the derivative!

- Solids: deformation itself is a function of the shear force:

Solid	Newtonian Fluid
$\tau \propto \alpha$	$\tau \propto \frac{\partial \alpha}{\partial t}$

In the fluids, there is a very large number of molecules.

- too costly to use direct integration of equation of motion
- the fluid instead is modelled as continuum, with fluid elements
- Macroscopic properties: obtained averaging over many fluid molecules



T, P, ρ, V, μ

Knudsen Number:

$$Kn = \frac{\lambda}{L}$$

λ = mean free path

L = length of transport

- $Kn << 1$ Continuum mechanics
- $Kn > 10$ Molecular dynamics

} Ratio between a molecule hitting another and the length of the transport.

Two options to evaluate the flow

Lagrangian approach:

- Reference frame moves with fluid element
- Material description as function of fluid element and time.

Eulerian approach

- Observer remains at a fixed position
- Field description as function of space and time

Reynolds Transport Theorem: Eulerian

Used to express temporal change of a material volume integral as the temporal change of the integral of a quantity over a fixed domain V and the flux over the boundary Sv of this domain

$$\underbrace{\frac{d}{dt}\iiint_{\tilde{V}(t)} \phi \, dV}_{\text{temporal change of material volume integral}} = \underbrace{\iiint_{\tilde{V}} \frac{\partial \phi}{\partial t} \, dV}_{\text{temporal change of the integral of a quantity over a fixed domain } V.} + \underbrace{\iint_{S_V} \phi u_i n_i \, dS}_{\text{flux over the boundary } S_V \text{ of the domain.}}$$

Conservation Laws Lagrangian

Usually formulated as the temporal derivative of material volume integrals with a dynamic boundary

$$\frac{dm}{dt} = 0 \qquad \frac{\partial}{\partial t}\int_{\tilde{V}} \rho(x,t) \, dV = 0$$

Conservation Laws

Simplified

Inviscid flows: $\mu = 0 \rightarrow$ Euler equations

Viscous effects can be often neglected at high Mach and Re.

Incompressible fluid: pressure independent of density.

Mass conservation implies volume conservation $\nabla \cdot u = 0$

Energy conservation follows directly from momentum conservation

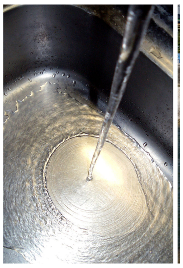
Barotropic fluids: pressure depends only on density

Energy equation is not needed.

Potential flows: inviscid and rotation free

Velocity field can be expressed as the gradient of scalar potential field. $u = \nabla \Phi$

Incompressible potential flows $\Delta \Phi = 0$



Dimensional Analysis

velocity	U_0
length	L
density	ρ_0
temperature	T_0
viscosity	μ_0
pressure	$p_0 = \rho_0 U_0^2$
time	$\tau_0 = L/U_0$
conductivity	κ_0

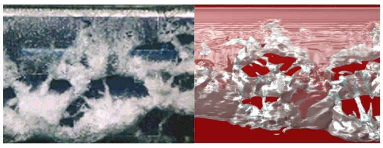
Dimensionless numbers

Reynolds number	$Re = \rho_0 U_0 L/\mu_0$	$\frac{\text{Inertial forces}}{\text{Viscous forces}}$
Mach number	$Ma = U_0/c_0 = U_0/\sqrt{\gamma R T_0}$	$\frac{\text{Advection velocity}}{\text{Speed of sound}}$
Strouhal number	non-dimensional frequency // 0.2 cylinder at high Re $St = f\,t_0 = f\;\underset{\hookrightarrow \text{diameter}}{L}/U_{0 \rightarrow \text{speed}}$	$\frac{\text{Unsteady forces}}{\text{Steady forces}}$
Froude number	$Fr = U_0/\sqrt{Lg}$ Mach number of	$\frac{\text{Inertial forces}}{\text{Gravity}}$
Weber number	$We = \rho_0 L U_0^2/\sigma_0$ Surface tension non-dimensional If high, not required	$\frac{\text{Inertial forces}}{\text{Surface forces}}$
Prandtl number	$Pr = c_p \mu_0/\kappa_0$ Viscosity and thermal conductivity 0.7	$\frac{\text{Viscosity}}{\text{Conductivity}}$ water 3-7

When the flow moves faster than waves.

- Small, single drop of water on calmed medium

- Turbulent water bubbles.



Reynolds Number:

Navier-Stokes in dimensionless form

$$\frac{\partial \underline{u}}{\partial t} + \nabla \cdot (\underline{u}\,\underline{u}) + \frac{1}{\rho} \nabla p - \frac{1}{Re} \nabla \cdot \nabla \underline{u} = 0$$

$$\nabla \cdot \underline{u} = 0$$

} incompressible fluid with constant density and constant viscosity.

Re << 1: creeping flow

- Viscous forces are dominant
- The linear Stokes equation is valid for flows in porous media.

Re >> 1: turbulent flow

Inertial forces dominant

Nonlinearity leads to production of small scales.

Re = O(1): laminar flow

Conservation laws

Derivations fully in 02_BASIC-EQUATIONS...

Continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \underline{u} = 0.$$

Conservation of mass

$$\frac{\partial}{\partial t} m_V = -\int_{S_V} \rho \underline{u} \cdot \underline{n} \, \mathrm{dS}$$

Conservation of Momentum

$$\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} = \sum_n F_{n,j}$$

In differential form

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_i} + f_j$$

In integral form

$$\frac{\partial}{\partial t}\int_{\bar{V}} \rho \underline{u} \, \mathrm{d}V + \int_{S_V} \rho \underline{u}\underline{u} \cdot \underline{n} \, \mathrm{d}S = -\int_{S_V} p \, \mathrm{d}S + \int_{S_V} \underline{\underline{\tau}} \cdot \underline{\mathrm{n}} \, \mathrm{d}S + \int_{\bar{V}} \rho \underline{f} \, \mathrm{d}V$$

Conservation of energy

$$\int_{\tilde{V}(t)} \rho E \, dV = \overset{\text{work flux}}{\int_{\tilde{V}(t)} \rho e \, dV} + \overset{\text{heat flux}}{\int_{\tilde{V}(t)} \frac{1}{2} \rho \underline{u}^2 \, dV}$$

In divergence form

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial(u_i \rho E)}{\partial x_i} = -\frac{\partial u_i p}{\partial x_i} + \frac{\partial u_i \tau_{ij}}{\partial x_i} - \frac{\partial \boxed{q_i}}{\partial x_i} + u_i \rho f_i$$

heat conduction

$$q_i = -k \frac{\partial T}{\partial x_i}$$

Grid Generation

Discretization:

Numerical methods are based on a discrete representation of solution and operators. Discrete mesh of cells.

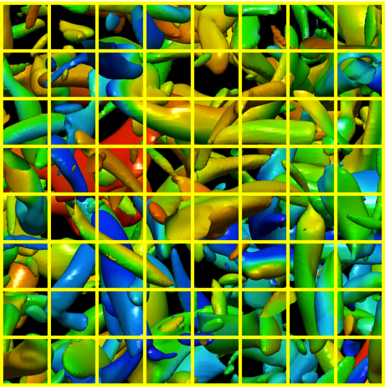
Various CFD methods to represent the solution:

Cell averages: finite volume discretization robust and fast CFX

Point values: finite difference discretization at center of vertex difficult for complex geometry

Coefficients: finite element discretization of basis functions high order, but slow.

Finite Volume



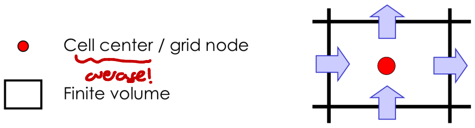
finite volume, they do not overlap.

get average on each finite volume

conservation laws at each FV → evaluated with fluxes. determine the evolution of the cell average.

volumen integrals → surface integrals

Gauss theorem



↓

Conservation law

$$\frac{\partial}{\partial t}\varphi = -\nabla \bullet \Psi \quad \rightarrow \quad \frac{\partial}{\partial t}\int_{\bar{V}} \varphi \, dV = -\int_{\bar{V}} \nabla \bullet \Psi \, dV$$

Gauss theorem applied

$$\int_{\bar{V}} \nabla \bullet \Psi \, dV = \int_{S_V} \mathbf{n} \bullet \Psi dS = \int_{S_{right}} \Psi dS - \int_{S_{left}} \Psi dS + - \cdots$$

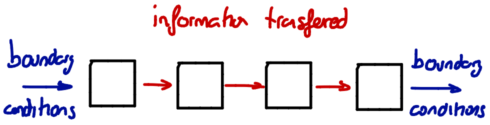
Momentum conservation example

$$\frac{\partial}{\partial t}\int_{\bar{V}} \rho \underline{u} \, \mathrm{d}V + \int_{S_V} \rho \underline{u}\underline{u} \cdot \underline{n} \, \mathrm{d}S = -\int_{S_V} p \, \mathrm{d}S + \int_{S_V} \underline{\underline{\tau}} \cdot \underline{\mathrm{n}} \, \mathrm{d}S + \int_{\bar{V}} \rho \underline{F}_V \, \mathrm{d}V$$

Finite Volume

Several approximation methods to calculate numerically.

- Evaluate flux at cell faces: reconstruction or interpolation
- Surface integral of fluxes: quadrature rules.



After substituting suitable boundary conditions, we obtain a system of equations.

Define Finite Volume Cells

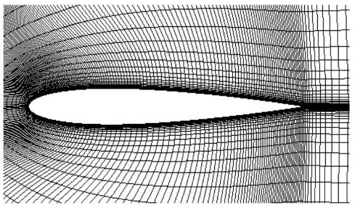
Topology: Relations between neighboring elements /cells

Geometry: Shape and size of cells and entire domain.

Topology:

Structured grids

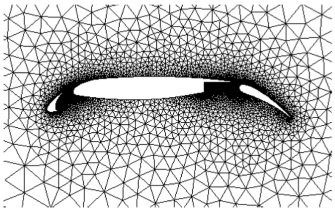
- internal cells are topologically similar
- all internal elements have same number and type of connections



- Sub domains can be projected onto Cartesian domains
- Mainly rectangles (2-D) and hexahedrons (3-D)
- Simple data structure and direct access
- Allows for "by-hand" grid generation, good quality control.
- Simple topology and implementation
- Efficient solution algorithms
- Local grid refinement requires multiblock grids

Unstructured grids

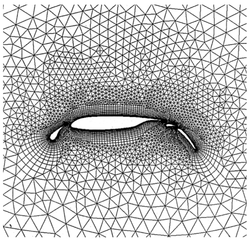
- elements can have neighbors with arbitrary topology
- No general rule for connectivity with neighbors



- Generally not possible to project
- Mainly triangles and tetrahedrons
- Complex data structure and access through connectivity matrix
- Straightforward application to complex geometries.
- Local grid refinement by dividing elements.
- Higher computing time per element

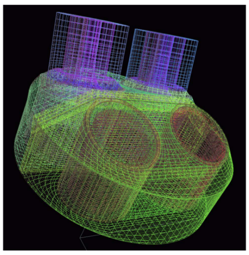
Hybrid grids

- Combination of structured and unstructured grids
- Implementation usually as unstructured grid, or as multi-block grid

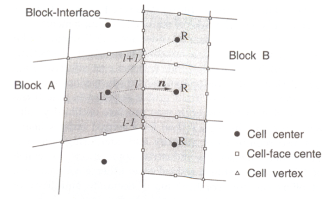


Multi-block grids

- The domain is decomposed in two levels. The sub domains on the first, coarse level are called blocks.
- In a second step, these blocks are the discretized with a structured or unstructured grid of cells.
- This way also complex geometries can be discretized with structured grids.

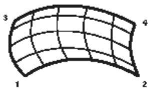


Definition of cell connectivity and flux over block boundaries can be challenging:

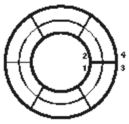


Types of Structured Grids

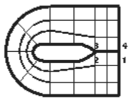
H – grid



O – grid



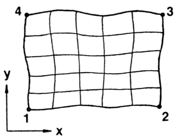
C – grid



Structured Grids

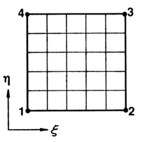
The mapping from physical space to computational space is done by coordinate transformation

Physical space



$$x = x(\xi,\eta)$$
$$y = y(\xi,\eta)$$

Computational space



Grid Requirements

Grid lines must not overlap or intersect between nodes. All cells must have positive volume.

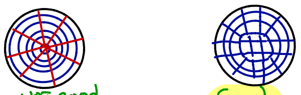
Should be easy to refine/coarsening the grid in selected regions.

The distribution and spacing of the boundary elements should be fine enough to represent geometry.

The grid should be smooth no more than 1.5 than before.

The grid should be as orthogonal as possible.

mesh a pipe flow



Not good in the center

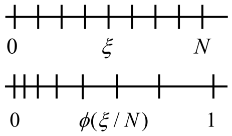
good avoid singularities

Algebraic distribution function:

Defined on block boundaries

Popular used:

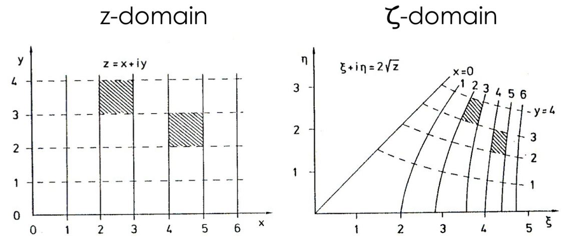
- geometric stretching
- exponential stretching
- hyperbolic stretching



Conformal Mapping

Defined for an entire block, or a 2D plane

The complex transformation $\zeta = \zeta(z)$ maps the complex z-domain $(z = x + iy)$ onto the complex ζ-domain $(\zeta = \xi + i\eta)$



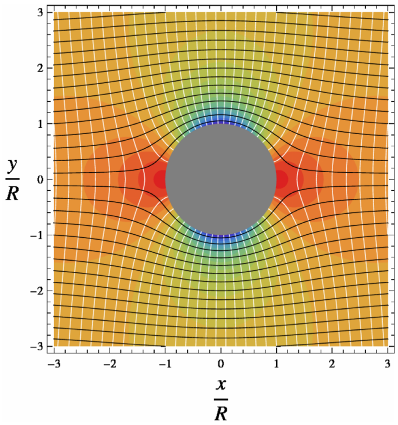
Real space

Grid

Elliptic Grid Generation

Formulate grid generation as a boundary value problem.

Lines of equal potential $(\phi = \text{const})$ and streamlines $(\Psi = \text{const})$ form an orthogonal grid



Either solve Laplace or Poisson equations.

$$\Delta\Phi = 0 \qquad \Delta\Psi = 0$$

$$\downarrow$$

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 \qquad \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0$$

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} = 0$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} = 0$$

$$\alpha = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2$$

$$\beta = \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}$$

$$\alpha = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2$$

This method searches for each pair (ξ, η) a corresponding (x, y), inverse Laplace.

Used to improve manually generated grids. Refinement along point or line: using control functions P and Q.

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = P(\xi, \eta) \qquad \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = Q(\xi, \eta)$$

Automatic Block Generator

Does not work for complex cases.

Works well for simple geometry

ICEM CFD

Define blocking topology. Specify resolution requirements and block boundaries.

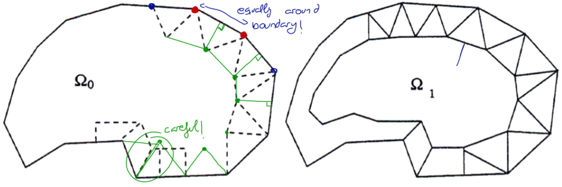
Icem generates automatically the mesh.

Unstructured Grids

- They offer much better flexibility. Tetrahedrons can be easier fitted to a geometry than hexahedrons.

Advancing Front Method

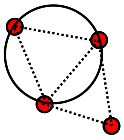
- Prescribe point distribution at domain boundary (2-D surface mesh/ surface triangulation)
- Method generates one cell layer along boundary
- This cell layer defines a new boundary, the „*advancing front*", which enclosed the remaining (unmeshed) domain



Sometimes requires final smoothing step: elliptic grid

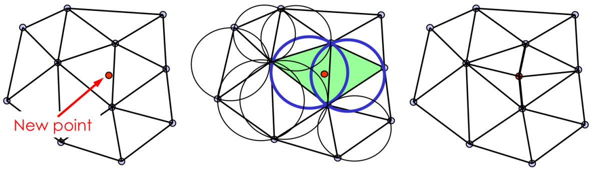
Also, the cell size can be controlled with background mesh.

Delaunay Triangulation



- Find the optimum connectivity matrix for a given cloud of points or distribution of points
- The circumscribed circle of every triangle must not contain any other point.
- Then the smallest angle interior angle is maximum. (Delaunay property)

Triangulation: Determine a network of triangles from a given cloud of points.



Advancing front

- Points and elements are generated simultaneously
- Several possible solutions
- Can control properties of elements individually
- Not robust, prone to errors (e.g. islands)
- Usually requires smoothing

many solutions, not optimal

fast, but not so good

Delaunay triangulation

- Point distribution is given (or defined through rules)
- Elements are created
- There is exactly one optimal solution

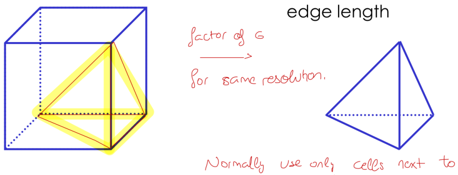
one unique solution

Structured Grids are Better

Memory and computing time in 3-D

better for memory and computational time.

1 hexahedron = 6 tetrahedrons with same edge length



factor of 6 ⟶ for same resolution.

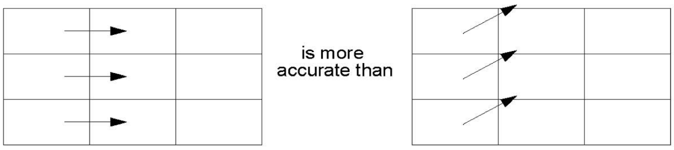
Normally use only cells next to!
Structure meshes are better for

Accuracy of discrete approximations

Numerical diffusion	the truncation error of discrete approximation of the continuous operations affects the numerical solution.

Numerical Diffusion:

- Reduce the effect by:
  - Making cell faces orthogonal to flow direction.
  - Align grid lines with flow direction.



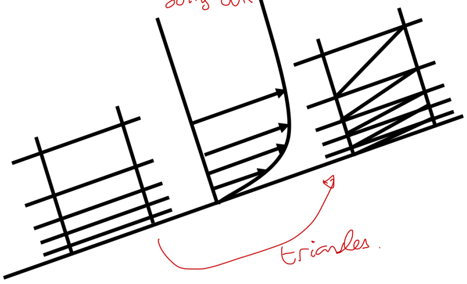
Again, unstructured grids won't align with flow thus they will be less precise.

More numerical error.

Prism Layers:

Combine the advantages of unstructured and structured grids. Improves accuracy at boundaries.

conservative because flux enters equals flux going out



Resolution of Boundary Layers – Recommended resolutions.

There are large velocity gradients at walls

Turbulence production happens mostly at $y^+ < 20$.

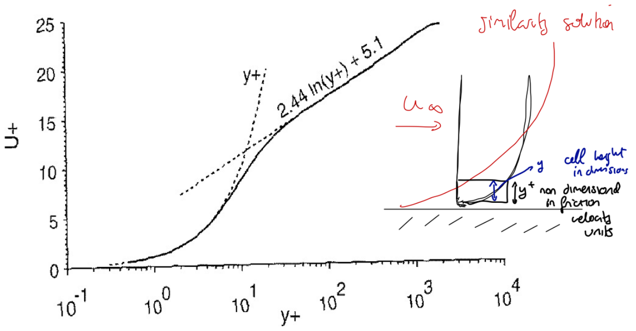
The wall shear can be computed accurately and reliably only if the first cell is within $y^+ \approx 1$

Resolution

$$y^+ = \frac{u_\tau}{\nu} y$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

$$u^+ = \frac{u}{u_\tau}$$



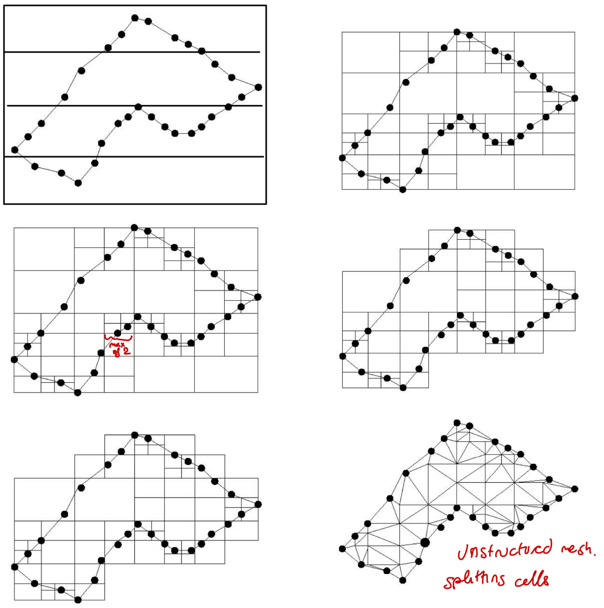
Wall Models

- Wall functions improve boundary conditions for simple equilibrium boundary layers. Allows for much coarser y+ resolution

- CFX includes one, ω based RANS

- Not working for strongly curved walls at large pressure gradients and for non-equilibrium turbulence

The octree method



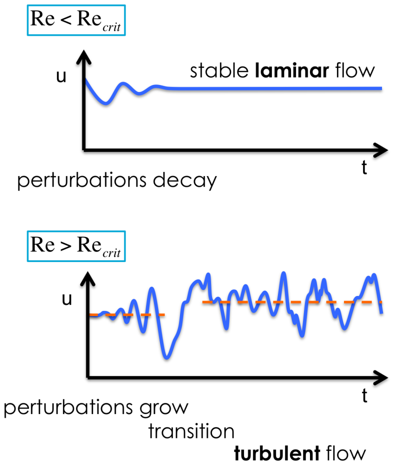
Data can be stored efficiently in a tree data structure

Every leave of the tree corresponds to a structured grid block.

At block boundaries, unstructured-grid interpolation rules are necessary.

TURBULENCE

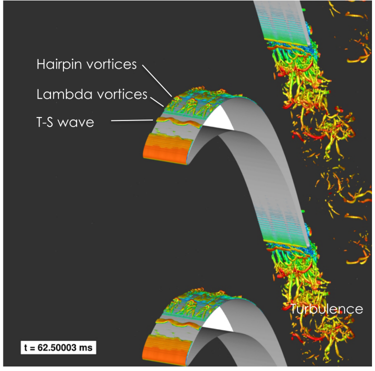
Osborne Reynolds first investigated it. Laminar flow recovers from disturbances at low velocities and becomes unstable at high speeds. $Re = \frac{\rho L U}{\mu}$



Ludwig Prandtl: discovered streamwise instability waves in a boundary layer.

Tollmien: found that Re_{crit} depends on wavelength

T-S waves: unstable and evolve to λ and hairpin vortices and eventually to turbulences.



Origin of Turbulence

- Very small perturbation grow in shear layers if the Re exceeds a critical value
- The Re_{crit} depends on wavelength of the perturbation
- The Kelvin-Helmholtz instability appears if there is a velocity difference across the boundary between two fluid layers

Effects of Turbulence

Enhances mixing

Increases wall friction

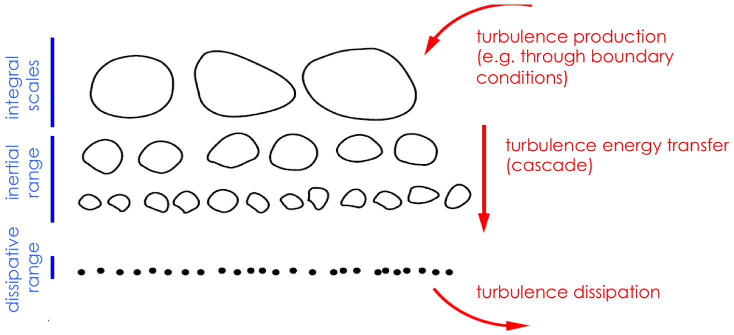
Delays flow separation

increases lift at low speeds

reduce pressure drag of bluff bodies golf ball

Turbulence Energy Cascade

Turbulence energy is generated on the largest (integral) scales, transfered to medium-small scale vortices (inertial subrange) and dissipated in (sub-) micro scale vortices (dissipative scales)


integral scales
inertial range
dissipative range
turbulence production (e.g. through boundary conditions)
turbulence energy transfer (cascade)
turbulence dissipation

More pronounced differences at high Re.

Large scales: Vortices are induced by and strongly depend on geometry and boundary conditions of the flow.

Small scales: Obtain their energy from the large scales through energy cascade.

Indirectly affected by geometry and b.c.

Easier to model than larger scales.

Characteristics of Turbulent Flow:

Unsteady, Rotational, Viscous, Breaking of symmetries, Chaotic, Wide range of length and time scales. "Coherent structures"

Potential flows are no rotational, therefore potential will be laminar flows.

Smallest Vortices

The smallest vortex structures characteristics can be obtained with:

Kolmogorov length $\eta_K = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4}$

Kolmogorov velocity $u_K = (\nu\varepsilon)^{1/4}$

Kolmogorov time $\tau_K = \sqrt{\frac{\nu}{\varepsilon}}$

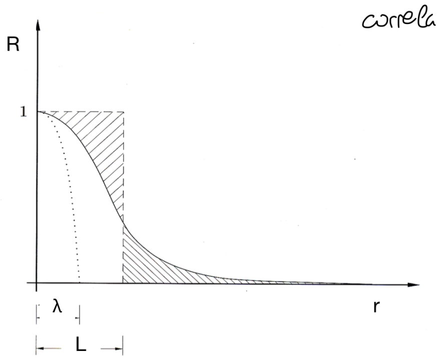
Largest Vortices:

Integral length scale L of the largest vortex structures can be estimated using flow geometry.

$$R(x,r)=\frac{\langle u'(x,t)u'(x+r,t)\rangle}{\sqrt{\langle u'^2(x,t)\rangle}\sqrt{\langle u'^2(x+r,t)\rangle}}$$ two point correlation function

The turbulence of two neighbouring points are coorrelated this function measures the time average coorrelation between fluctuating quantities, at a distance "r".

If this coorelation is computed for different r, the integral of R over r will be the Integral length scale.



correlation length

$$\downarrow$$

$$L=\int_0^\infty R(x,r)\,dr$$

Direct Numerical Simulation (DNS)

Captures all length and time scales of a turbulent flow. η_k and τ_k directly without assumptions. Using Navier stock equations.

These hold for laminar and turbulent flows.

$$\left.\begin{aligned}&\frac{\partial \underline{u}}{\partial t}+\nabla\cdot(\underline{u}\underline{u})+\frac{1}{\rho}\nabla p-\frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{u}=0\\&\nabla\cdot\underline{u}=0\end{aligned}\right\}$$ incompressible fluid with constant density and constant viscosity

Computational Cost

Number of grid points $N_L \sim \frac{L}{\eta_k} \sim \mathrm{Re}^{3/4}$ ⟶ 3 Dimensions $N_L^3 \sim \mathrm{Re}^{9/4}$

Number of time steps $N_T \sim N_L$

We can compute small things at low Reynolds number.

Total cost ⟶ $N_T \cdot N_L = \mathrm{Re}^3$

Uses:

Useful for fundamental turbulence research but not for every-day engineering flow simulations.

CFD for Aerospace Engineers

Requirements: High Re, moderate M, Complex geometry, Transition, turbulence and separation.

Derive simpler equations from conservation laws that capture the most important flow qualities

METHODS: RANS steady mean solution
URANS include unsteady effect
LES large and energetic scales
DES hybrid methods.

DNS – direct numerical simulation

- All turbulent structures accurately resolved in space and time
- Very fine 3-D grid and time steps, effort scales with Re.

LES – large-eddy simulation

- All gradients of the mean flow and the most energetic (that is, the largest) turbulent structures are resolved in space and time
- 3-D grid with moderate number of cells and time steps; cost independent of Re (except in boundary layers)

RANS – Reynolds-averaged Navier-Stokes simulation

- All gradients and structures of the mean flow are resolved
- Coarse, sometimes only 2-D grid; steady state or a few time steps

Reynolds Averaged Navier Stokes

Reynolds averaging

Ensemble averaged solution

$$\langle u_i \rangle = \frac{1}{N}\sum_{\mu=1}^{N} u_i\big|_\mu$$

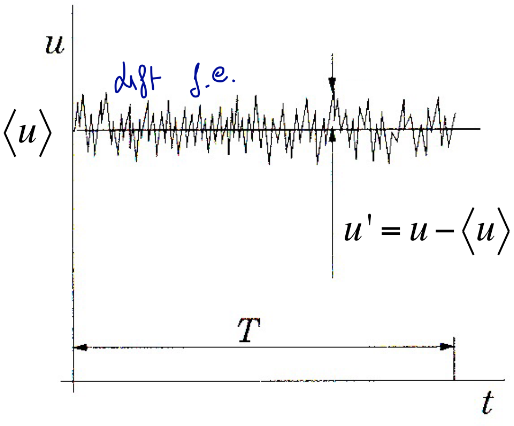
mean flow if you run simulation many times

average

↳ integral average.

For statically stationary processes:

$$\langle u_i \rangle = \lim_{t\to\infty}\frac{1}{t}\int_0^t u_i(t')dt'$$



The solution is decomposed in the mean value and the fluctuation.

$$\underbrace{u_i}_{\text{solution}} = \underbrace{\langle u_i \rangle}_{\text{mean}} + u_i' \longrightarrow \text{fluctuation}$$

Reynolds averaging is an orthogonal projection:

$$\langle\langle u_i \rangle\rangle = \langle u_i \rangle,\ \langle u_i' \rangle = 0$$

Average the average, same value.

Remind Navier Stokes dimensionless

$$\left.\begin{aligned} &\frac{\partial \underline{u}}{\partial t} + \nabla\cdot(\underline{u}\underline{u}) + \frac{1}{\rho}\nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{u} = 0 \\ &\nabla\cdot\underline{u} = 0 \end{aligned}\right\}$$

incompressible fluid
with constant density
and constant viscosity

Rules on Reynolds Averaging

Rules

$$\langle u+v\rangle = \langle u\rangle + \langle v\rangle$$

$$\langle au\rangle = a\langle u\rangle \quad ,a = const$$

$$\langle\langle u\rangle v\rangle = \langle u\rangle\langle v\rangle$$

$$\left\langle \frac{\partial u}{\partial x}\right\rangle = \frac{\partial\langle u\rangle}{\partial x}$$

$$\langle u'\rangle = 0$$

$$\langle\langle u\rangle v'\rangle = 0$$

$$\langle uv\rangle = \langle u\rangle\langle v\rangle + \langle u'v'\rangle$$

pressure and density

$$\langle \frac{1}{\rho}\nabla p\rangle = \frac{1}{\rho}\cdot\nabla\langle\langle p\rangle\rangle + \frac{1}{\rho}\nabla\langle p'\rangle = \frac{1}{\rho}\nabla\langle p\rangle$$

Reynolds

$$\langle \frac{1}{Re}\nabla\nabla u\rangle \rightsquigarrow \frac{1}{Re}\nabla\nabla\langle u\rangle$$

RANS Derivation

Substitude the sum of mean value and fluctuation into the N-S equation.

Apply averaging operator, simplify as much as possible.

$$\frac{\partial u}{\partial t} + \nabla\cdot(\underline{u}\underline{u}) + \frac{1}{\rho}\nabla p - \frac{1}{Re}\nabla\cdot\nabla\underline{u} = 0$$

$$\nabla\cdot\underline{u} = 0$$

mass incompressible

$$\nabla\cdot u = 0$$

$$\langle\nabla(\langle u\rangle + u')\rangle = \langle 0\rangle$$

$$\nabla\cdot(\underbrace{\langle\langle u\rangle\rangle}_{\langle u\rangle} + \underbrace{\langle u'\rangle}_{0}) = 0$$

$$\nabla\langle u\rangle = 0$$

momentum conservation

$$\langle\frac{\partial u}{\partial t}\rangle = \frac{\partial\langle\langle u\rangle\rangle}{\partial t} + \cancelto{0}{\frac{\partial\langle u'\rangle}{\partial t}}$$

$$\sim \frac{\partial\langle u\rangle}{\partial t}$$

Rest

$$\langle\nabla(uu)\rangle = \langle\nabla\cdot[(\langle u\rangle + u')(\langle u\rangle + u')]\rangle \qquad u = \langle u\rangle + u'$$

$$= \nabla\cdot\langle\langle u\rangle\langle u\rangle\rangle + \nabla\langle u'\langle u\rangle\rangle + \nabla\langle\langle u\rangle u'\rangle + \nabla\cdot\langle u'\cdot u'\rangle$$

$$= \nabla\cdot\langle u\rangle\langle u\rangle + \underbrace{\nabla\cdot\langle u\rangle\langle u'\rangle}_{0} + 0 + \underline{\nabla\cdot\langle u'\cdot u'\rangle}$$

→ aditional, used as independent

Combination of all RANS

$$\frac{\partial\langle u\rangle}{\partial t} + \nabla\langle\underline{u}\rangle\langle\underline{u}\rangle + \nabla\langle\underline{u}'\cdot\underline{u}'\rangle + \frac{1}{\rho}\nabla\langle p\rangle - \frac{1}{Re}\nabla\nabla\langle\underline{u}\rangle = 0$$

$$\nabla\cdot\langle\underline{u}\rangle = 0$$

RANS

$$\frac{\partial\langle\underline{u}\rangle}{\partial t} + \nabla\cdot(\langle\underline{u}\rangle\langle\underline{u}\rangle) + \frac{1}{\rho}\nabla\langle p\rangle - \frac{1}{Re}\nabla\cdot\nabla\langle\underline{u}\rangle = -\nabla\cdot\langle\underline{u}'\underline{u}'\rangle$$

$$\nabla\cdot\langle\underline{u}\rangle = 0$$

Reynolds stress tensor $\tau_{ij} = -\langle u'_i u'_j\rangle$

More unknowns than equations

Turbulence models provide approx of stress tensor

Transportation equation for Re stress

RST Reynolds Stress Transport.

momentum equation multiply:

i-component with U_j'

j-component with U_i'

then apply the averaging operator

The arithmetic mean of both equations is the RST with the general form

$$\frac{\partial\langle u'_i u'_j\rangle}{\partial t} + \underbrace{K_{ij}}_{\text{Advection}} = \underbrace{P_{ij}}_{\text{Production}} + \underbrace{T_{ij} + D^{v}_{ij} + D^{p}_{ij}}_{\text{Diffusion}} + \underbrace{\Phi_{ij}}_{\text{Pressure strain correlation}} - \underbrace{\varepsilon_{ij}}_{\text{Dissipation}}$$

Transport equation

Advection
- Transport by mean flow

$$K_{ij} = \langle u_k \rangle \frac{\partial \langle u'_i u'_j \rangle}{\partial x_k}$$

Turbulent diffusion
- Transport by turbulence

$$T_{ij} = \frac{\partial \langle u'_i u'_j u'_k \rangle}{\partial x_k}$$

non linear
triple

Production
- Source term
- Increase of turbulence intensity by energy transfer from the mean flow

$$P_{ij} = -\left(\langle u'_i u'_k \rangle \frac{\partial \langle u_j \rangle}{\partial x_k} + \langle u'_j u'_k \rangle \frac{\partial \langle u_i \rangle}{\partial x_k} \right)$$

Viscous diffusion
- Transport by viscous stress

$$D^{v}_{ij} = \frac{1}{\mathrm{Re}} \frac{\partial^2 \langle u'_i u'_j \rangle}{\partial x_k \partial x_k}$$

Pressure diffusion
- Transport by pressure fluctuations

$$D^{p}_{ij} = -\frac{1}{\rho}\left(\frac{\partial \langle u'_j p' \rangle}{\partial x_i} + \frac{\partial \langle u'_i p' \rangle}{\partial x_j} \right)$$

Pressure strain correlation
- Redistribution of energy among components

$$\Phi_{ij} = \left\langle \frac{p'}{\rho}\left(\frac{\partial u'_j}{\partial x_i} + \frac{\partial u'_i}{\partial x_j} \right) \right\rangle$$

effect of pressure fluctuations

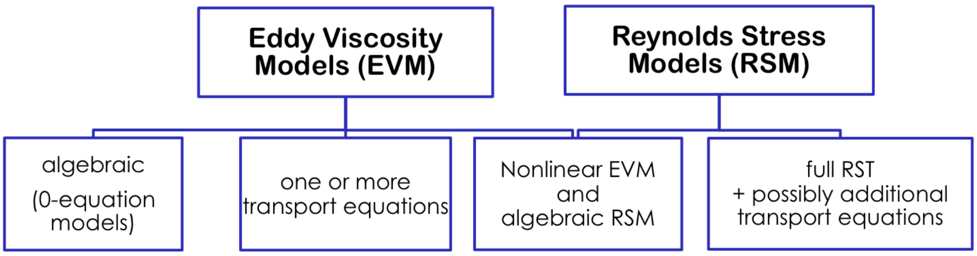
Dissipation
- Sink term
- Dissipation into thermal energy

$$\varepsilon_{ij} = \frac{2}{\mathrm{Re}} \left\langle \frac{\partial u'_j}{\partial x_i} \frac{\partial u'_i}{\partial x_j} \right\rangle$$

The order of new unknowns increases during every step. Thus we can not solve it this way. We have to use empirical approximations TURBULENCE MODELS

TURBULENCE MODELS

viscosity coefficient ⟶ Reynolds Number


Eddy Viscosity Models (EVM)

Reynolds Stress Models (RSM)

algebraic (0-equation models)

one or more transport equations

Nonlinear EVM and algebraic RSM

full RST + possibly additional transport equations

EDDY VISCOSITY HIPOTHESIS – EDDY VISCOSITY MODEL (EVM)

Turbulence leads to momentum exchange between fluid elements

turbulence kinetic energy

Kronecker delta

$$-\langle u'_i u'_j \rangle \cong 2\nu_T S_{ij} - \frac{2}{3}\delta_{ij} k$$

shear rate

eddy viscosity

The deviator of RS, $-\langle u'_i u'_j \rangle$ is proportional to the mean shear rate. Proportionality factor is eddy viscosity Vt.

The turbulence kinetic energy adds turbulence to flow. Turbulence viscosity to add mixing.

$$\delta_{ij} = \begin{cases} 1 & , if\ i = j \\ 0 & , if\ i \neq j \end{cases}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) - \frac{1}{3}\delta_{ij} \frac{\partial \langle u_k \rangle}{\partial x_k}$$

turbulence intensity ←

$$k = \frac{1}{2}\langle u'_i u'_i \rangle = \frac{1}{2}\sum_{i=1}^{3} \langle u'_i u'_i \rangle$$

Substituting EVM on RANS yields: NEW RANS

$$\frac{\partial \langle u_i \rangle}{\partial t} + \langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} = -\frac{\partial}{\partial x_i}\left(\frac{\langle p \rangle}{\rho} + \frac{2}{3}k \right) + \frac{\partial}{\partial x_j}\left(2(\nu + \nu_t) S_{ij} \right)$$

Problem reduced to one scalar field: eddy viscosity

Strong assumption but confirmed by DNS for flows with boundary layers.
distinct flow direction

Furthermore Vt can be expressed as mean velocity.
next page

EDDY VISCOSITY MODELS (CONTINUED)

Dimensional arguments lead to a variety of expressions of eddy viscosity.

- Zero-equation models: e.g.: $\nu_t \propto l_m^2 \left\| \frac{\partial\langle u_i\rangle}{\partial x_j} \right\|$
- One-equation models: e.g.: $\nu_t \propto l_m \sqrt{k}$
- Two-equation models: e.g.: $\nu_t \propto \frac{k^2}{\varepsilon}$ or $\nu_t \propto \frac{k}{\omega}$

kinetic energy, dissipation

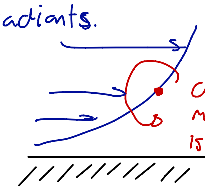
$\boxed{\text{dissipation}/k}$

Models are characterized by the number of required transport equations.

ZERO-EQUATION MODELS - BOUNDARY LAYER METHOD

Mixing length - l_m : typical length scale of turbulence chose properly.

Turbulent mixing is only important if you have velocity gradients.



characteristic mixing length is the distance from wall

Not use in complex flows around the walls.

Eddy viscosity is approx constant in the outer part of the b-l.

Turbulence length scales smaller towards the walls.

$$l_m = 0.2\kappa u_\tau \delta_{99}, \quad y \geq 0.2\delta_{99}$$

boundary layer thickness

$$l_m = \kappa y \underbrace{\left(1 - e^{-y^+/A^+}\right)}, \quad y < 0.2\delta_{99}$$

0 pressure gradient boundary layer

von Kármán constant
$\kappa = 0.41$

van Driest damping
$A^+ \approx 26$

ONE EQUATION MODEL

Includes turbulence intensity in eddy viscosity

replacing local velocity gradient by local turbulent intensity gradient

$$-\langle u_i' u_j' \rangle = 2\nu_T S_{ij} - \frac{2}{3}\delta_{ij} k$$

Transport equation for the turbulence kinetic energy "k."

$$k = \frac{1}{2}\langle u_i' u_i' \rangle = \frac{1}{2}\sum_{i=1}^{3}\langle u_i' u_i' \rangle$$

one half of the trac of the Re stress tensor

$$\frac{\partial k}{\partial t} + \underbrace{\langle u_j \rangle \frac{\partial k}{\partial x_j}}_{\text{Advection}} = \underbrace{-\langle u_i' u_j' \rangle \frac{\partial \langle u_i \rangle}{\partial x_j}}_{\text{Production:}} + \underbrace{\frac{\partial}{\partial x_j}\left(-\frac{1}{2}\langle u_i' u_i' u_j' \rangle + \frac{1}{\text{Re}}\frac{\partial k}{\partial x_j} - \frac{1}{\rho}\langle p' u_j' \rangle \right)}_{\text{Diffusion}} \underbrace{- \frac{1}{\text{Re}}\left\langle \frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k} \right\rangle}_{\text{Dissipation}}$$

$$\left(2\nu_T S_{ij} - \frac{2}{3}\delta_{ij} k \right)\frac{\partial \langle u_i \rangle}{\partial x_j} \qquad \frac{\nu_t}{\text{Pr}_k}\frac{\partial k}{\partial x_j} \qquad C_D \frac{k^{3/2}}{l_m}$$

ONE EQUATION: MODEL TRAUSPORTATION EQUATION

$$\underbrace{\frac{\partial k}{\partial t}+\langle u_j\rangle\frac{\partial k}{\partial x_j}}_{\text{Advection}}=\underbrace{\tau_{ij}\frac{\partial\langle u_i\rangle}{\partial x_j}}_{\text{Production}}+\underbrace{\frac{\partial}{\partial x_j}\left(\left[\frac{1}{\mathrm{Re}}+\frac{\nu_t}{\mathrm{Pr}_k}\right]\frac{\partial k}{\partial x_j}\right)}_{\text{Diffusion}}-\underbrace{C_D\frac{k^{3/2}}{l_m}}_{\text{Dissipation}}$$

$$\tau_{ij}=2\nu_T S_{ij}-\frac{2}{3}\delta_{ij}k$$

$$\nu_T=l_m\sqrt{k}$$

$$C_D=0.07...0.09$$

$$\mathrm{Pr}_k=1$$

REVIEW:

- Includes turbulence production, transport and dissipation, for this reason much better than all algebraic 0-equation models.
- Computationally very efficient because only one transport equation.

- The assumption of a constant mixing length for all flows outside boundary layers is a strong limitation and often not justified.

- For this reason not suitable for internal flows and flow separation, but often satisfying results for external flows and attached boundary layers.

JONES & LAUNDER TWO EQUATION MODEL $k-\varepsilon$ model.

Equilibrium between turbulence production and disipation. Isotropic turbulence.

$$\ell_m = C_D\frac{k^{3/2}}{\varepsilon} \qquad \nu_t = C_D\frac{\boxed{k^2}}{\boxed{\varepsilon}}$$

→ Partial differential equation

→ aditional transport equation
because of complexity of ε. } Postulate ε

Same unknowns as in one equation model: But disipation is solved on ε.

$$P = \tau_{ij}\frac{\partial\langle u_i\rangle}{\partial x_j} \qquad \tau_{ij}=2\nu_T S_{ij}-\frac{2}{3}\delta_{ij}k$$

$$T+D^p=\frac{\nu_t}{\mathrm{Pr}_k}\frac{\partial k}{\partial x_j} \qquad \nu_T=C_D\frac{k^2}{\varepsilon}$$

$$\underbrace{\frac{\partial \varepsilon}{\partial t}+\langle u_j\rangle\frac{\partial \varepsilon}{\partial x_j}}_{\text{Advection}}=\underbrace{C_{\varepsilon 1}\frac{\varepsilon}{k}\tau_{ij}\frac{\partial\langle u_i\rangle}{\partial x_j}}_{\text{Production}}+\underbrace{\frac{\partial}{\partial x_j}\left(\left[\frac{1}{\mathrm{Re}}+\frac{\nu_t}{\mathrm{Pr}_\varepsilon}\right]\frac{\partial \varepsilon}{\partial x_j}\right)}_{\text{Diffusion}}-\underbrace{C_{\varepsilon 2}\frac{\varepsilon}{k}\varepsilon}_{\text{Dissipation}}$$

$C_D=0.09$, $\mathrm{Pr}_k=1$, $\mathrm{Pr}_\varepsilon=1,3$, $C_{\varepsilon 1}=1,44$, $C_{\varepsilon 2}=1,92$

Obtained in flows for with k and ε cancel out.

REVIEW

- Low computational overhead because only two additional transport equations.
  Gives good results for external flows (aerodynamics)

- The k-ε model should applied only to flows without strong pressure gradients, stream line curvature or separation.
  Formulation of (numerical) boundary conditions for ε is difficult.

- Eddy viscosity models assume that the Reynolds stress is proportional to the mean shear rate. They cannot distinguish effects of the individual components of the Reynolds stress tensor. The model fails to predict anisotropic influences, such as streamline curvature and directional volume forces (gravity).

Wilcox Two-Equation Model k-ω model, one of most used

Solve the transport equation of turbolence kinetic energy k and a postulated transport equation for the dissipation ω.

$$\tau_{ij} = 2\nu_T S_{ij} - \frac{2}{3}\delta_{ij} k \quad , \quad \nu_T = \frac{k}{\omega} \quad , \quad \omega = \frac{1}{C_D}\frac{\varepsilon}{k}$$

Similar to k-ε but very different results.

Transport equation for turbolence kinetic energy

$$\underbrace{\frac{\partial k}{\partial t} + \langle u_j \rangle \frac{\partial k}{\partial x_j}}_{\text{Advection}} = \underbrace{\tau_{ij}\frac{\partial \langle u_i \rangle}{\partial x_j}}_{\text{Production}} + \underbrace{\frac{\partial}{\partial x_j}\left(\left[\frac{1}{\mathrm{Re}} + \frac{\nu_t}{\mathrm{Pr}_k}\right]\frac{\partial k}{\partial x_j}\right)}_{\text{Diffusion}} - \underbrace{C_D k \omega}_{\text{Dissipation}}$$

$C_D = 0.09$ $\quad \alpha = 5/9$

$\mathrm{Pr}_k = 2$ $\quad \beta = 3/40$

$\mathrm{Pr}_\omega = 2$

Transport equation for specific dissipation rate.

$$\underbrace{\frac{\partial \omega}{\partial t} + \langle u_j \rangle \frac{\partial \omega}{\partial x_j}}_{\text{Advection}} = \underbrace{\alpha \frac{\omega}{k}\tau_{ij}\frac{\partial \langle u_i \rangle}{\partial x_j}}_{\text{Production}} + \underbrace{\frac{\partial}{\partial x_j}\left(\left[\frac{1}{\mathrm{Re}} + \frac{\nu_t}{\mathrm{Pr}_\omega}\right]\frac{\partial \omega}{\partial x_j}\right)}_{\text{Diffusion}} - \underbrace{\beta \omega^2}_{\text{Dissipation}}$$

Review

Low computational overhead because only two additional transport equations.

Gives very good results for boundary layer flows and for flows with pressure gradients and separation (surprising)

Simple wall boundary conditions for ω.

Very sensitive on inflow and freestream boundary conditions, for this reason is the k-ε better for external aerodynamics.

Over estimation of turbulence production at stagnation points.

Blending with k-ε leads to **SST model** (Florian Menter, 1993)

SST gives similar results as Wilcox (2004) „improved“ k-ω model.

S-A Eddy Viscosity Transport Models Spalart-Allmaras model

Based on postulated transport equation for a functional of the eddy viscosity.

$$\nu_t = \tilde{\nu} f_{v1}$$

$$\frac{\partial \tilde{\nu}}{\partial t} + \langle \underline{u} \rangle \cdot \nabla \tilde{\nu} = C_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + f_{t1}\Delta \underline{u}^2 + \frac{1}{\sigma}\nabla \cdot [(\nu + \tilde{\nu})\nabla \tilde{\nu}]$$

$$+ \frac{C_{b2}}{\sigma}|\nabla \tilde{\nu}|^2 - \left[C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2$$

With NOT Important probably but anyways

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad \chi = \frac{\tilde{\nu}}{\nu}$$

$$\tilde{S} = \sqrt{2\Omega_{ij}\Omega_{ij}} + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} - \frac{\partial u_i}{\partial x_j}\right)$$

$$f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{1/6} \qquad g = r + C_{w2}(r^6 - r) \qquad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

$$f_{t1} = C_{t1} g_t \exp\left(-C_{t2}\frac{\omega_t^2}{\Delta \underline{u}^2}\left[d^2 + g_t^2 d_t^2\right]\right)$$

$$f_{t2} = C_{t2}\exp\left(-C_{t4}\chi^2\right)$$

and with calibrated parameter values for all C_{XX} and σ
and with the von Kármán constant $\kappa = 0.41$
and with the wall distance d

Review

Very robust.

Computationally very efficient because it requires just one transport equation.

Often very good results for simple attached flows and also for flow-separation location.

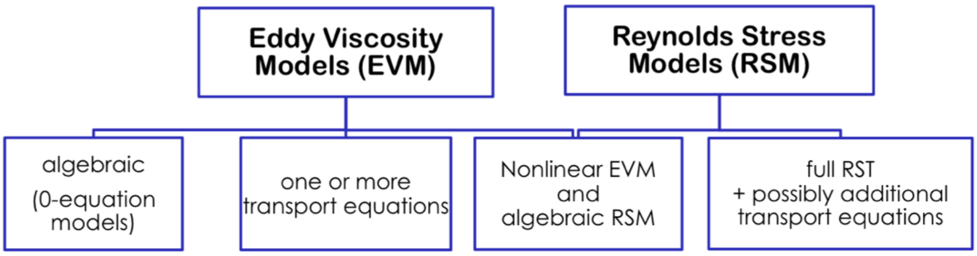
Less suitable for the prediction of flow reattachment and free shear layers.

No underlying physical theory at all, if you still care...

Basis of original detached eddy simulation (DES) method.

Turbulence Models

viscosity coefficient ⟶ Reynolds Number



Eddy Viscosity Models (EVM)

Reynolds Stress Models (RSM)

algebraic (0-equation models)

one or more transport equations

Nonlinear EVM and algebraic RSM

full RST + possibly additional transport equations

Reynolds Stress Models

Directly solve model transport equations for all components of the unknown Re tensor.

how can we model this

$$\frac{\partial\langle u_i'u_j'\rangle}{\partial t}+\underbrace{K_{ij}}_{\text{Advection}}=\underbrace{P_{ij}}_{\text{Production}}+\left(\underbrace{T_{ij}+D_{ij}^{v}+D_{ij}^{p}}_{\text{Diffusion}}+\underbrace{\Phi_{ij}}_{\text{Pressure strain correlation}}-\underbrace{\varepsilon_{ij}}_{\text{Dissipation}}\right)$$

Approximations of higher order correlations and dissipation rate tensor.

Reynolds Stress Models (RSM)

Pressure diffusion is rather small, neglected D_{ij}^{P}

Triple correlations in the turbulent transport term are approximated by: Hanjalic & Launder.

$$T_{ij}=\frac{\partial}{\partial x_k}\langle u_i'u_j'u_k'\rangle$$

assume gaussian distribution ↓

$$\approx\frac{\partial}{\partial x_k}\left(-C_s\frac{k}{\varepsilon}\left(\langle u_i'u_l'\rangle\frac{\partial\langle u_j'u_k'\rangle}{\partial x_l}+\langle u_j'u_l'\rangle\frac{\partial\langle u_i'u_k'\rangle}{\partial x_l}+\langle u_k'u_l'\rangle\frac{\partial\langle u_i'u_j'\rangle}{\partial x_l}\right)\right)$$

Triple correlation is a combination of double correlation.

Turbulence dissipation: acts on small scales. We assume scale separation and decoupling through the turbulence energy cascade.

The dissipation rate tensor ε_{ij} is modelled as an isotropic tensor, represented by a scalar quantity.

$$\varepsilon_{ij}\simeq\frac{2}{3}\delta_{ij}\,\varepsilon$$

Pressure-strain correlation cannot produce or disipate turbulence energy.

$$\Phi_{ij} = \left\langle \frac{p'}{\rho}\left(\frac{\partial u'_j}{\partial x_i} + \frac{\partial u'_i}{\partial x_j}\right)\right\rangle$$

takes energy from u fluctuation into v.
Total reinolds stress tensor remains constant

- Φ_{ij}^{S} slow: relaxation to isotropic equilibrium slote
- Φ_{ij}^{R} rapid: immediate effects of mean flow gradients and extend forces.

This term redistributes turbulence between the components of the Re stress tensor.

The pressure-strain correlation only affects the ansotropy aij of the Re stress tensor.

$$a_{ij} = \frac{\langle u'_i u'_j \rangle}{\langle u'_k u'_k \rangle} - \frac{1}{3}\delta_{ij}$$

$$\Phi_{ij}^{S} = -\varepsilon\left(C_1 a_{ij} + C_2\left(\underbrace{a_{ik}a_{kj} - \frac{1}{2}P_{ij}\delta_{ij}}_{\text{second non lineants, more complex, calibration}}\right)\right)$$

although not required!

LRR and SGM differences in pg. 41 & 42 OS_RANS

$$\Phi_{ij}^{R} = k\frac{\partial\langle u_k\rangle}{\partial x_l}\left(X_{kjli}(a_{nm}) + X_{kilj}(a_{nm})\right)$$

Recommendations & Experience

- The ***k-ε* model** performs well for exterior flows and thin, two-dimensional shear layers. Predictions are poor for boundary layers and in particular in cases with strong pressure gradients.
  The ***k-ω* model** yields more accurate results for boundary layers than the *k-ε* model.
- The **SST model** (Menter, 1993) combines the *k-ω* and *k-ε* model. However, it does not always give more accurate results that the separate models.
  Wilcox (2004) proposed a new version of his *k-ω* model, which gives very similar results as the SST model (can be good or bad)
- Linear eddy viscosity models overestimate turbulence production at stagnation points.
  There are many ad-hoc modifications and workarounds, e.g., for stream line curvature.

- **Reynolds stress models (RSM)** give (by design) superior results for flows with strong streamline curvature or swirl, and for (anisotropic) turbulence induced secondary flow structures.

Problems of RSM:

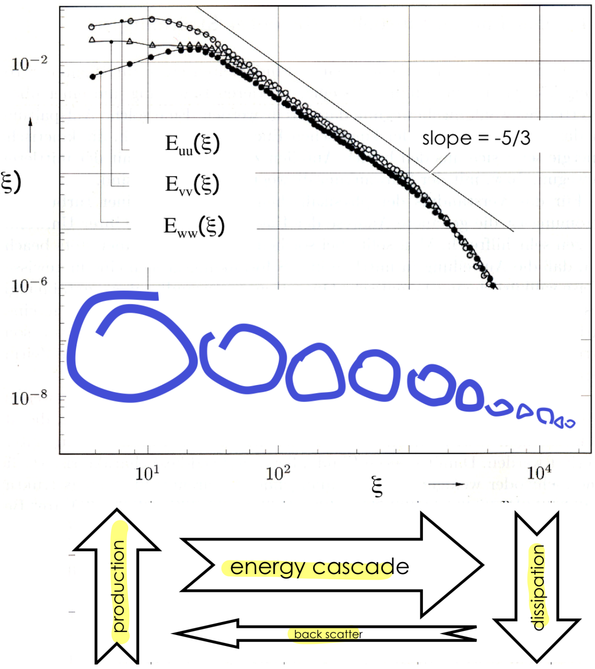
- RSM are computationally expensive
  (at least 6, usually 7, transport equations)
- Convergence can be slow or even unstable.

- **Explicit algebraic RSM** (EARSM) reconstruct the Reynolds stress tensor or an anisotropic eddy viscosity tensor from a reduced number of transport equations.
- **Recommendation:** The in CFX implemented version of the *ω*–based EARSM of Wallin and Johansson (2000) is robust and gives excellent results at low computational cost.

LES: Large Eddy Simulation

Spectrum of turbulence energy

Energy flux from small to large scales



Large turbulence scales	**Small turbulence scales**
Produced by external energy input	Receive energy through cascade from larger scales
Depend on geometry and case specific boundary conditions	Similar in all turbulent flows
Inhomogeneous and anisotropic	Usually relatively homogeneous and isotropic
Large and long living	Small and short live time
High energy content, strong effect on loads and performance	Low energy content
Modeling very difficult	Easy to model
Models work only for special cases	**Universal models possible**

The main question is, where do we make the separation between large and small?

Using a grid or a filter. Big solved by simulation and small by modelling

Small Scales

Unresolved scales *(sub-grid scales / sub-filter scales)*

- Cannot be resolved unless the grid is very fine (c.f. Kolmogorov micro scales)
- Must be modeled in an LES

Dissipation!

Large Scales

Resolved scales *(sometimes grid scales)*

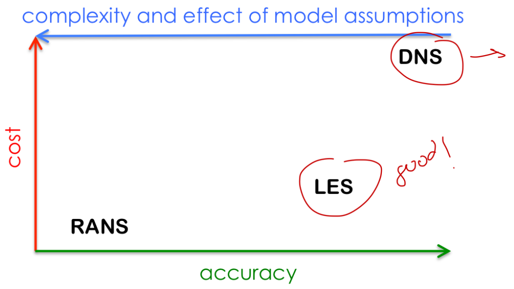
- Spatial and temporal evolution can be *represented* and *resolved* with moderate effort
- Are computed directly and represent the solution of an LES

Simulation Classification

DNS: exact simulation of all scales

RANS: only the average value is computed,
the complete turbulence spectrum is modeled

LES: the most energetic large scales are simulated,
the effect of small scale turbulence structures is modeled

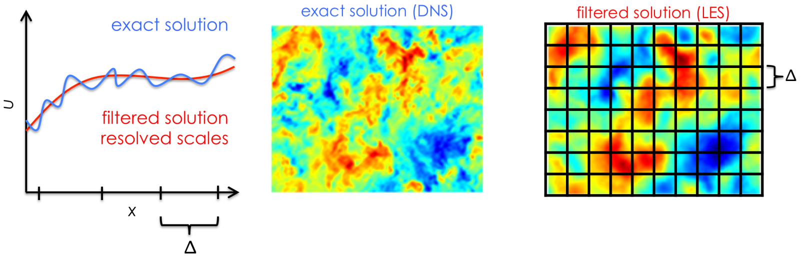


Filters or Grids to separate scales:

Spatial filter:

- captures large energetic scales
- removes small scales.

Δ: filter width ≅ grid width



Filtes : Scale Separation

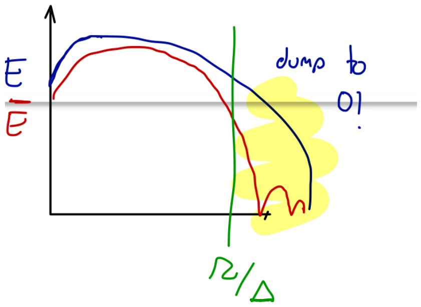
Convolution integral "kernel filter" average basically.

$$\bar{u}_i = G * u_i = \int_{-\infty}^{+\infty} G(x-x')\, u_i(x')\, dx' \quad \text{with} \quad \int_{-\infty}^{+\infty} G(x-x')\, dx' = 1$$

Convolution theorem, convolution in real space (physical) corresponds to multiplication in fourier (spectral) space

$$F\left(\bar{u}_i\right) = F\left(G * u_i\right)$$
$$\hat{\bar{u}}_i(\xi) = \hat{G}(\xi)\ \hat{u}_i(\xi)$$

Spectral Outfall



Common Filter Kernels

volume method

top-hat filter (box filter)

mathematics

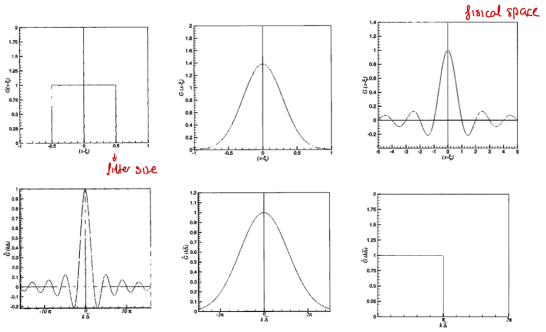
Gauss filter

polynomial, you don care about fisical space

spectral-cutoff filter

Real space $G(x)$

Spectral space $\hat{G}(\xi)$



Filtering produces the large-scale part (the resolved)

$\bar{u}_i = G \cdot u_i$

From the identity $u_i = \bar{u}_i + u_i''$ trivially follow the subgrid scales

$u_i'' = u_i - \bar{u}_i$

Filtering is no orthogonal projection

$\bar{\bar{u}}_i \neq \bar{u}_i \qquad \overline{\overline{u_i''}} \neq 0$

LES EQUATIONS

We assume a homogeneous filter $G \circ \partial_x u = \partial_x G \circ u$

Applying this filter to the momentum equation gives.

$$\frac{\partial \underline{\bar u}}{\partial t} + \overline{\nabla\cdot(\underline{u}\underline{u})} + \frac{1}{\rho}\nabla \bar p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{\bar u} = 0$$

And the N-S eqs for large scales

$$\frac{\partial \underline{\bar u}}{\partial t} + \nabla\cdot(\underline{\bar u}\,\underline{\bar u}) + \frac{1}{\rho}\nabla \bar p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{\bar u} = -\nabla\cdot\underbrace{\left(\overline{\underline{u}\underline{u}} - \underline{\bar u}\,\underline{\bar u}\right)}$$

$$\nabla\cdot\underline{\bar u} = 0$$

subgrid-scale stress tensor $\tau_{ij} = \overline{u_i u_j} - \bar u_i \bar u_j$

Substitude $u_i = \bar u_i + u_i''$ in τ_{ij}

$$\tau_{ij} = \underbrace{\overline{\bar u_i \bar u_j} - \bar u_i \bar u_j}_{\text{Leonard stress tensor}} + \underbrace{\overline{\bar u_i u_j''} + \overline{u_i'' \bar u_j}}_{\text{Cross stress tensor}} + \underbrace{\overline{u_i'' u_j''}}_{R}$$

Leonard stress tensor

(we can compute it exactly!) we know everything

$$L_{ij} = \overline{\bar u_i \bar u_j} - \bar u_i \bar u_j$$

Cross stress tensor

$$C_{ij} = \overline{\underset{\text{known}}{\bar u_i} u_j''} + \overline{u_i'' \bar u_j} \xrightarrow{} \text{not known}$$

goes accross small and large scales.

Reynolds stress tensor

$$R_{ij} = \overline{u_i'' u_j''}$$ → small, only what we dont know

Subgrid Scale Modeling

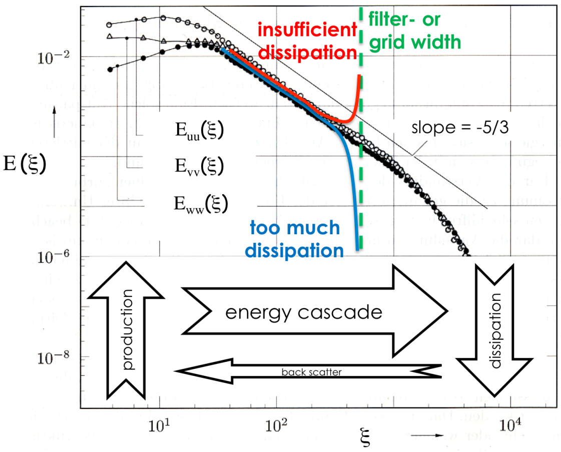
Requirements: exact is impossible.

> we want to model the effect of the subgrid-scale turbulence on the large resolved scales.

> Observations:

- Subgrid scales receive energy through energy cascade
- Backscatter is much smaller than forward arrow size!
- Dissipation happens on unresolved scales.

- Subgrid scale models have to provide the correct energy drain from the resolved scales. If not, this can happen:



SMAGORINSKY MODEL

$$\tau_{ij} = \frac{1}{3}\tau_{kk}\delta_{ij} - 2\nu_{SGS}\bar S_{ij}$$

$$\bar S_{ij} = \frac{1}{2}\left(\frac{\partial \bar u_i}{\partial x_j} + \frac{\partial \bar u_j}{\partial x_i}\right)$$

Turbulence causes momentum exchange analogously to the molecular diffusion (gas kinetics)

$C_S \approx 0.2$ from energy spectrum of homogeneous isotropic turbulence

Using Prandtl's mixing (0-equation) on LES we obtain

$$\nu_{SGS} = (\underset{\uparrow}{C_S}\underset{\uparrow}{\bar\Delta})^2\sqrt{2\bar S_{ij}\bar S_{ij}} = (C_S\bar\Delta)^2|\bar S_{ij}|$$

filter width $\bar\Delta = \sqrt[3]{\text{cell volume}}$

Smagorinsky constant

Close to the walls C_S has to be corrected by van-Driest damping.

$$C_S = C_{S0}\left(1 - e^{-\frac{y^+}{26}}\right)^2$$

isotropic turbulence	$C_S \approx 0.14 \ldots 0.25$
channel flow (high Re)	$C_S \approx 0.1$
channel flow (low Re)	$C_S \approx 0.065$

Smagorinsky Model Review

Review (personal)

- Very simple and efficient model
- Does not require additional PDEs

- No numerical problems
- Drains sufficient amount of energy from the resolved scales, which stabilizes the simulation and is the reason for the huge success of this model until today

- Very unsatisfying correlation with exact subgrid-scale stresses
- Better model are required (and available) for complex flows
- Optimal value of Smagorinsky constant varies a lot

- The Smagorinsky model is implemented CFX and other solvers.

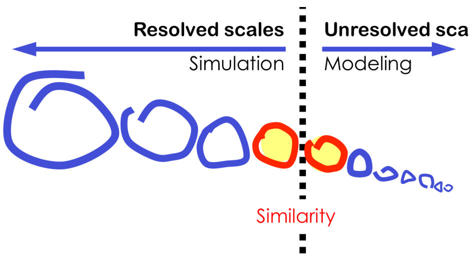
DSM: Dynamic Smagorinsky Model

The problem with this model is with the Cs. The dynamic Smagorinsky (DSM) proposes a Cs = Cs (x,t)

Hypothesis:

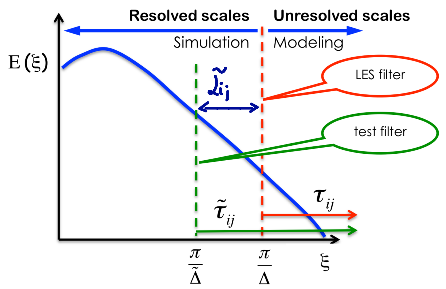
Most of the effect of the SGS stress tensor is due to the interaction btw the small and large scales.

Turbulence on those scales are similar



DSM: Continued

The filtered velocity field is filtered again using a test filter $\widetilde{\Delta} \cong 2\Delta$



τ_{ij} and $\widetilde{\tau}_{ij}$ can be modeled with the same value for the model constant. Smagonnsky Eddy viscosity model

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j = \frac{1}{3}\tau_{kk} - 2(C_s\Delta)^2|\bar{S}_{ij}|\bar{S}_{ij}$$

$$\widetilde{\tau}_{ij} = \widetilde{\overline{u_i u_j}} - \widetilde{\bar{u}}_i \widetilde{\bar{u}}_j = \tfrac{1}{3}\widetilde{\tau}_{kk} - 2(C_s\widetilde{\Delta})^2|\widetilde{\bar{S}}_{ij}|\widetilde{\bar{S}}_{ij}$$

The germano identity: $\widetilde{\tau}_{ij} = \widetilde{L}_{ij} + \tau_{ij}$ connects the models on filter and test filter level. $\widetilde{L}_{ij} = \widetilde{\bar{u}_i \bar{u}_j} - \widetilde{\bar{u}}_i \cdot \widetilde{\bar{u}}_j$ → Leonard stress

If we substitute the values:

$$\widetilde{L}^d_{ij} = \widetilde{L}_{ij} - \tfrac{1}{3}\widetilde{L}_{kk}\delta_{ij} = -2C_D\widetilde{\Delta}^2|\widetilde{\bar{S}}_{ij}|\widetilde{\bar{S}}_{ij} + 2\,\overset{\sim}{\overbrace{C_D\Delta^2|\bar{S}_{ij}|\bar{S}_{ij}}}$$

$C_D = (C_s)^2$ can not be extracted directly. $C_D(x,t)$ is filtered by test-filter.

If we neglect the spatial variation of CD we obtain:

$$\widetilde{L}^d_{ij} = C_D\left(2\,\overset{\sim}{\overbrace{\Delta^2|\bar{S}_{ij}|\bar{S}_{ij}}} - 2\widetilde{\Delta}^2|\widetilde{\bar{S}}_{ij}|\widetilde{\bar{S}}_{ij}\right)$$

$$\widetilde{L}^d_{ij} = C_D M_{ij}$$ solved by least-sqare optimization $$C_D = \frac{\widetilde{L}^d_{ij} M_{ij}}{M_{ij} M_{ij}}$$

This particular form was proposed by Lilly (1991) and should be preffered over the original proposal of Germano

Review

- For the first time a closed model!
- Significant improvement over original Smagorinsky model
- No further empirical assumptions required
- Improves prediction of near-wall turbulence
- Does not require van-Driest damping if the test filter is formulated in such a way that it filters only on wall tangential directions

- **Backscatter:** instants of local energy transfer from small to large scales. C_D can assume positive and negative values, the model is therefor able to model „backscatter" – in theory.
- In practice, this backscatter model often leads to instabilities, so that negative values of ν_{SGS} are usually clipped.
- Still poor correlation of modeled and exact SGS stress tensor.

Scale Similarity Model

Approach without the eddy viscosity approximation

The resolved filtered velocity is used as unfiltered velocity and substituted into the definition of the SGS tensor.

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$$
$$\approx \overline{\bar{u}_i \bar{u}_j} - \bar{\bar{u}}_i \bar{\bar{u}}_j$$

Numerically unstable, not enough disipation.

Approximate Deconvolution Model (ADM)

Generalized high-order scale-similarity model

Series expansion of inverse filter operation:

$$\frac{1}{G} = \sum_{m=0}^{\infty} (1-G)^m$$

Deconvolution of the filtered velocity returns the unfiltered velocity with arbitrary accuracy.

$$u_i = \sum_{m=0}^{\infty} (1-G)^m * \overline{u}_i$$

More information how it works in pg. 33 of OG-LES

- Excellent correlation of modeled and exact SGS stress tensor.
- Numerically unstable without additional dissipation mechanism.

Mixed Models

Zang: combined method with a dynamic eddy viscosity model

$$\tau_{ij} = \overline{\tilde{u}_i \tilde{u}_j} - \overline{\tilde{u}}_i \overline{\tilde{u}}_j - 2(C_s\Delta)^2 |\bar{S}_{ij}| \bar{S}_{ij}$$

one of the best

Current Trends

Combining RANS and LES

- Zonal coupling: some zones use RANS and others LES problems at coupling conditions.
- Detached eddy simulation: based on grid resolution and wall distance.

Implicit large-eddy simulation (ILES)

Full coupling of numerical modeling (discretization) and physical modeling (SGS turbulence model).

Often used as synonym for under-resolved DNS with dissipative finite volume schemes, which are stabilized by (an often excessive amount of) numerical diffusion.

However, numerical schemes can been designed in such a way that numerical diffusion is consistent with turbulence theory.

Different from eddy viscosity models, implicit LES is not limited by assuming isotropy of SGS turbulence.

Particularly suitable for complex flows with anisotropic turbulence and for complex fluids.

Summary

RANS

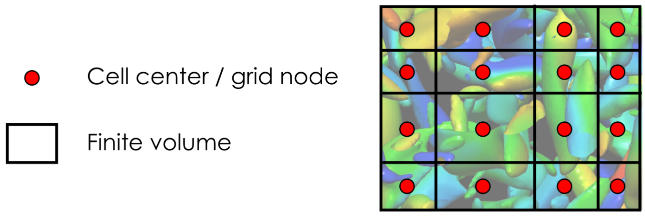
- Decompose flow in (temporal or ensemble)mean and fluctuation
- The whole turbulence spectrum is modeled
- Grid refinement beyond a certain level does not automatically lead to better results, because we still solve approximate model equations.

LES

- Decompose flow in resolved scales and unresolved (SGS) scales
- Model only the effect of subgrid scales on resolved scales
- LES is always unsteady and 3-D
- Grid refinement leads to more accurate results (LES -> DNS for Δ -> 0)
- High potential for complex flows, but more expensive than RANS.

FINITE VOLUME METHOD

- The computational space is decomposed into non-overlapping control volumes := finite volumes (FV)
- Every FV is considered as a control volume (CV) for which we compute the evolution of the mean values.
- The solution represents the cell average value.



The conservation law is integrated over the FV

$$\frac{\partial}{\partial t}\varphi = -\nabla \bullet \Psi \quad \rightarrow \quad \frac{\partial}{\partial t}\int_V \varphi \, dV = -\int_V \nabla \bullet \Psi \, dV$$

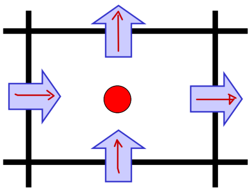
↑ Conserved quantity ↑ Flux $\Psi = \Psi(\varphi)$

Gauss' theorem is applied to transform volume integrals to surface integrals.

$$\int_V \nabla \bullet \Psi \mathrm{d}V = \int_{S_V} \mathbf{n} \bullet \Psi \mathrm{d}S = \int_{S_{right}} \Psi \mathrm{d}S - \int_{S_{left}} \Psi \mathrm{d}S + - \cdots$$
$$= F_{right} - F_{left} + - \cdots$$

The fluxes ⇨ over the FV surface S_V determine the time evolution of the cell average

interpolation to cell average
integrate flux over all cells



The numerical evaluation of the flux balance over the FV requires approximation methods.

The fluxes have to be approximated from the known cell average solutions φ

For each FV!

APPROXIMATIONS:

Quadrature: Surface integral of fluxes is a sum of discrete values at one or several points at the cell surface.

Interpolation: Values of φ at the cell surfaces are reconstructed from the values of φ at the cell centers

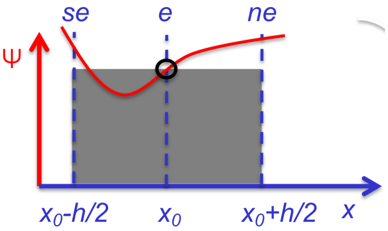
Approximation of Surface Integrals

Mid-point rule: (2nd order)

$$F_e = \int_{S_e} \Psi \, dS = \tilde{\Psi}_e A_e \qquad A_e = \int_{S_e} 1 \, dS \qquad \tilde{\Psi}_e = \frac{1}{A_e} \int_{S_e} \Psi \, dS = \text{mean value.}$$

The integral is approximated as the product of the integrant Ψ_e with the area of the cell face.

Error:

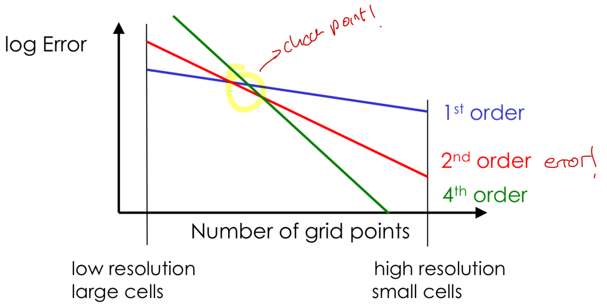


$$\Psi(x) = \overset{\Psi(x_0)}{\Psi_0} + \tilde{x} \left.\frac{\partial \Psi}{\partial x}\right|_0 + \frac{\tilde{x}^2}{2} \cdot \left.\frac{\partial^2 \Psi}{\partial x^2}\right|_0 + \ldots \qquad \tilde{x} = x - x_0$$

$$\int_{-h/2}^{h/2} \Psi(\tilde{x}) \, d\tilde{x} = h \cdot \Psi_0 + 2 \frac{h^3}{8 \cdot 6} \left.\frac{\partial \Psi^2}{\partial x^2}\right|_0$$

$$\underbrace{\frac{1}{h} \int = \Psi_0}_{\text{Midpoint rule}} \quad \underbrace{\frac{h^2}{24} \left.\frac{\partial \Psi^2}{\partial x^2}\right|_0}_{} \quad \underline{\text{error truncation}} \longrightarrow O(h^2)$$

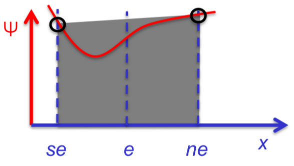
The order of the method determines the "rate of grid convergence for a sufficiently smooth solution".



We can observe that the higher order methods are actually not so good for low resolution and large cells.

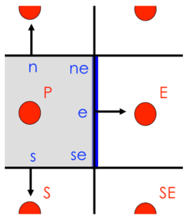
Trapezoidal Rule: (2nd order)

$$F_e = \int_{S_e} \Psi \, dS \approx A_e \frac{\Psi_{ne} + \Psi_{se}}{2}$$



Simpson Rule (4th order)

$$F_e = \int_{S_e} \Psi \, dS \approx A_e \frac{\Psi_{ne} + 4\Psi_e + \Psi_{se}}{6}$$



Approximation of Volume Integrals

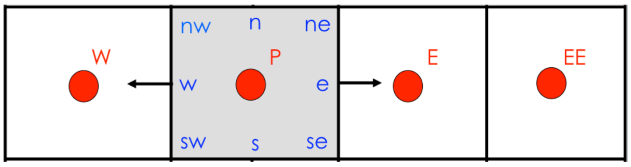
Most common is a 2nd order approximation

$$Q_P = \int_V q \, dV = \bar{q} V_P \approx q_P V_P$$

This rule is **exact**, if *q* is constant or a linear function within *V*.

Interpolation

To evaluate the quadrature rules we need to compute the flux at one or several points at the cell surface.

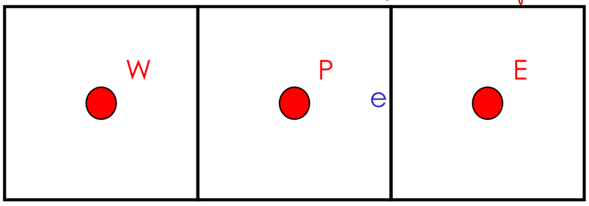


These points are reconstructed from the volume integrals by spatial interpolation. $\varphi_e = f(\varphi_w, \varphi_P, \varphi_E \dots)$

Upwind Interpolation (Upwind differencing scheme - UDS)

$$\Psi_e = \begin{cases} \Psi_P, \text{ if } (\underline{u}\cdot\underline{n})_e > 0 \\ \Psi_E, \text{ if } (\underline{u}\cdot\underline{n})_e < 0 \end{cases}$$

solution at P if left to right, e if right to left.

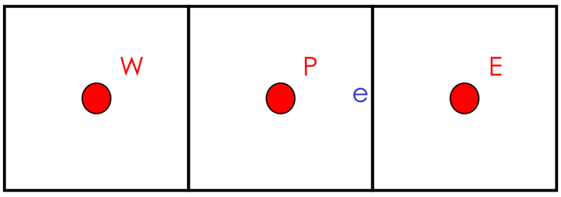


Linear Interpolation (Central differencing scheme - CDS)

$$\Psi_e = \Psi_E\lambda_e + \Psi_P(1-\lambda_e)$$

$$\lambda_e = \frac{x_e - x_P}{x_E - x_P}$$

average of two closest solution at cell center



Truncation Error - Numerical Diffusion

Analyzed with Taylor series expansion:

$$\Psi_e = \Psi_P + (x_e - x_P)\left(\frac{\partial \Psi}{\partial x}\right)_P + \frac{(x_e - x_P)^2}{2}\left(\frac{\partial^2 \Psi}{\partial x^2}\right)_P + \dots$$

Truncation error of UDS

$$E_{\text{UDS}} \propto (x_e - x_P)\left(\frac{\partial \Psi}{\partial x}\right)_P$$

(plus higher-order terms)

Proportional to grid width $(x_e - x_P)$

Example: UDS discretization of linear advection equation:

$$\frac{\partial \varphi}{\partial t} + U\frac{\partial \varphi}{\partial x} = 0$$

Effect of numerical approximations on numerically computed solution?

Upwind approximation:

$$\frac{\partial \varphi}{\partial t} \approx \frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t}$$

Temporal discretization: n – time step

$$\frac{\partial \varphi}{\partial x} \approx \frac{\varphi_i^n - \varphi_{i-1}^n}{\Delta x}$$

Spatial discretization: i – grid node

Taylor series expansion of φ

$$\boxed{\varphi_i^{n+1} = \varphi_i^n + \Delta t}\frac{\partial \varphi}{\partial t} + \boxed{\frac{\Delta t^2}{2}\frac{\partial^2 \varphi}{\partial t^2} + \dots}$$

numerical method

$$\boxed{\varphi_{i-1}^n = \varphi_i^n - \Delta x}\frac{\partial \varphi}{\partial x} + \boxed{\frac{\Delta x^2}{2}\frac{\partial^2 \varphi}{\partial x^2} + \dots}$$

let alone

error

what we want to compute

Example (continued)

Substitution

$$\frac{\partial\varphi}{\partial t}+U\frac{\partial\varphi}{\partial x}=0 \quad\Rightarrow$$

$$\frac{\varphi_i^{n+1}-\varphi_i^n}{\Delta t}+U\frac{\varphi_i^n-\varphi_{i-1}^n}{\Delta x}\underbrace{-\frac{\Delta t}{2}\frac{\partial^2\varphi}{\partial t^2}+U\frac{\Delta x}{2}\frac{\partial^2\varphi}{\partial x^2}}_{\text{truncation error}}=0$$

We neglect the truncation error

The exact solution will be including the truncation error.

$$\frac{\partial\varphi}{\partial t}+U\frac{\partial\varphi}{\partial x}+\left(\underbrace{\frac{U^2\Delta t}{2}-\frac{U\Delta x}{2}}_{}\right)\frac{\partial^2\varphi}{\partial x^2}=0$$

This error is an aditional difusion term. (numerical diffusion)

$\boxed{V_N}$ expressed as constant. If the original equation has a diffusion term already then both diffussions are summed.

$$Re=\frac{uL}{V} \qquad Re_{effective}=\frac{uL}{V+V_N}$$

A numerical method with negative (ND) is unstable.

$$\Delta t\le \Delta x/U$$

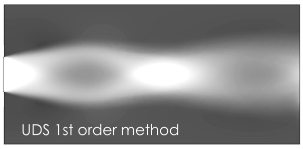
Max accuracy at $\Delta t=\Delta x/U$

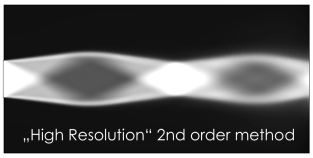
$$CFL=\Delta t\frac{U}{\Delta x}$$

Truncation error on (CDS) central differencing scheme

$$E_{CDS}\propto\frac{(x_e-x_P)(x_E-x_e)}{2}\left(\frac{\partial^2\Psi}{\partial x^2}\right)_P$$

approx proportional to the square of the cell size $(x_e-x_P)(x_E-x_e)$ 2nd order method

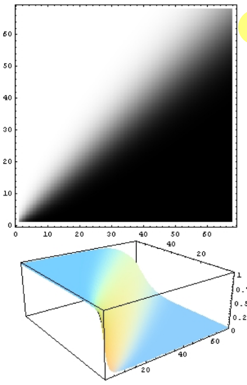

UDS 1st order method


„High Resolution“ 2nd order method

Numerical diffusion and dispersion

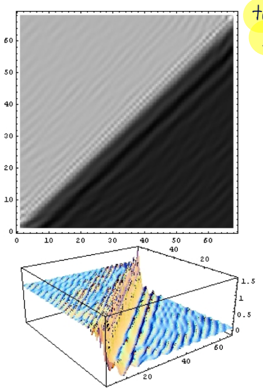
UDS
diffusive error

dump function artificially



shock better!

CDS
dispersive error



turbulent better!

Discretization of Gradients

To evaluate diffusion terms we have to approximate gradients of φ at the FV surface.

$$\frac{\partial}{\partial t}\int_{\bar V}\rho\underline{u}\,\mathrm{d}V+\int_{S_V}\rho\underline{u}\underline{u}\cdot\underline{n}\,\mathrm{d}S=-\int_{S_V}p\,\mathrm{d}S+\boxed{\int_{S_V}\underline{\underline{\tau}}\cdot\underline{\mathrm{n}}\,\mathrm{d}S}+\int_{\bar V}\rho\underline{F}_V\,\mathrm{d}V$$

Viscous stress is approximated with a CDS method because this closely corresponds to the isotropic character of a diffusion:

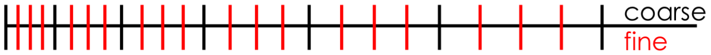
$$\left(\frac{\partial\varphi}{\partial x}\right)_e\approx\frac{\varphi_E-\varphi_P}{x_E-x_P}$$

Discretization of Gradients (continued)

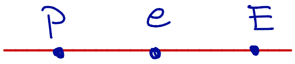
From the Taylor series expansion follows:

$$\left(\frac{\partial\psi}{\partial x}\right)_e = \frac{\psi_E - \psi_P}{x_E - x_P} + E \longrightarrow E \approx -\underbrace{\frac{(x_E-x_e)^2 \ominus (x_e-x_P)^2}{2(x_E-x_P)}}_{\sim\Delta x}\left(\frac{\partial^2\psi}{\partial x^2}\right)_e + \underbrace{\frac{(x_E-x_e)^3 \oplus (x_e-x_P)^3}{6(x_E-x_P)}}_{\sim\Delta x^2}\left(\frac{\partial^3\psi}{\partial x^3}\right)_e + \dots$$

Non uniform grids, 1st order method since the leading order error scales with Δx

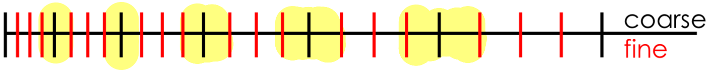


For uniform grids, 2nd order method. The order increases by one.



Non Uniform Grid

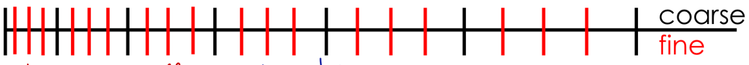
1. Refine cells by splitting them into equal intervals:



The first order terms appear where the distance to both sides is not the same.

The global truncation error will converge only slightly slower than a 2nd order method.

2. Refine cells by splitting them into sub intervals with equal stretching.



dx is very small, sometimes better

The first error term converges faster than the second error term. 2nd order convergence.

Conclusions

- Non uniform grids can give the same rate of grid convergence as refinement of uniform grids.
- Grids should be smooth

Higher Order Methods

- Can give more accurate solution on same grid.
- Can be obtained by including information from more neighbor cells.
- Simple for structured grids
- Difficult / computationally expensive for unstructured grids.

Ansys - CFX Interpolation

1st order Upwind Differencing Scheme UDS_1

2nd order Upwind Differencing Scheme UDS_2

2nd order Central Differencing Scheme CDS

1st-2nd order blend factor (UDS_1 <-> UDS_2, $0 \leq \beta \leq 1$)

UNSTEADY PROBLEMS

- For unsteady problems time has to be discretized aswell, making it a 4th dimension.
- Unsteady problems, the future has no influence on the past, thus they are parabolic in time.
- Unsteady problems = Initial boundary value problems (solution depends on initial conditions and boundary conditions)
- Same method as spatial discretization

EXAMPLE conservation law : What is the solution φ at t_{n+1} $t_n = t_0 + \Delta t$

$$\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t} = f(t,\varphi(t)) \quad ; \quad \varphi(t_0) = \varphi^0$$

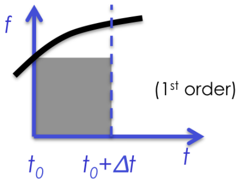
$\longrightarrow$ Integration $\displaystyle\int_{t_n}^{t_{n+1}} \frac{\mathrm{d}\varphi}{\mathrm{d}t}\mathrm{d}t = \varphi^{n+1} - \varphi^n = \int_{t_n}^{t_{n+1}} f(t,\varphi(t))\mathrm{d}t$ $\longrightarrow$

$$\boxed{\varphi^{n+1} = \varphi^n + \int_{t_n}^{t_{n+1}} f(t,\varphi(t))\mathrm{d}t}$$

TIME MARCHING METHODS

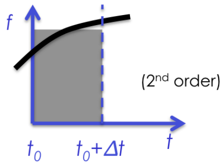
Euler forward (explicit)

$$\varphi^{n+1} = \varphi^n + f(t_n,\varphi^n)\Delta t$$



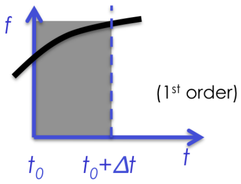
Mid-point rule (implicit)

$$\varphi^{n+1} = \varphi^n + f(t_{n+1/2},\varphi^{n+1/2})\Delta t$$



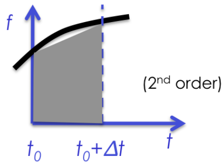
Euler backward (implicit)

$$\varphi^{n+1} = \varphi^n + f(t_{n+1},\varphi^{n+1})\Delta t$$



Trapezoidal rule (implicit)

$$\varphi^{n+1} = \varphi^n + \frac{f(t_n,\varphi^n) + f(t_{n+1},\varphi^{n+1})}{2}\Delta t$$



For a small Δt all methods converge to the same solution

The order of a method tells how fast it converges to true if Δt is sufficiently small.

$$CFL = \Delta t \frac{U}{\Delta x} < 0.5$$ is good enough for time integration errors.

IMPLICIT values of φ at $t > t_n$

ADVANTAGES

Stable for much larger time steps

DISADVANTAGES

large memory requirements

iteratively, complex implementation

Careful with implicit time marching and LES

EXPLICIT φ at t_n only

ADVANTAGES

No iteration necessary, efficient

Straight forward implementation

low memory requirements

DISADVANTAGES

Unstable for large time steps.

CFL gives max step

Local vs Physical Time Step

The maximum reasonable time step size Δt is limited by flow physics and or numerical stability.

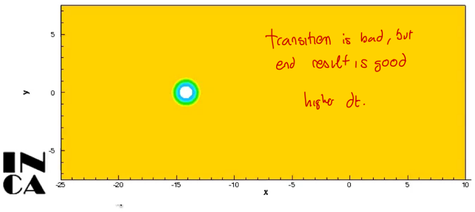
The solution must not physically propagate over a large distance than what is covered by the influence domain of the computational stencil.

Influence domain: h ($\simeq$ cell width) — Improvements by $\Delta t = \Delta t(x)$

Signal speed: $S = |U| + c$ compressible

$S = |U|$ incompressible

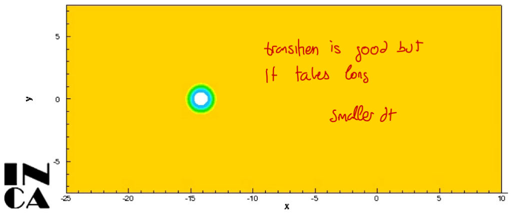
Max time step: $\Delta t \approx \min h/S$ → Appropiate time step is often limited by smallest cell in domain.



Local time step:

Transient behavior is not physical!

Insufficiently converged results can be completely wrong!



Physical time step:

Transient is physical flow evolution. Converged steady state result is reached very slowly (or never).

Converged steady state results in identical but are obtained faster with local time step. But the transient is not good. Can be used as an accelerator in dual time stepping methods.

Dual Time Stepping Method

Split discrete time derivative into:

- Large physical time step Δt, using an implicit method.
- Pseudo-time part $\Delta \tau$

since you shoul get 0 at steady state

discretize implicit

$$\frac{\partial \varphi}{\partial t} = f(\varphi(t)) \Leftrightarrow \lim_{\tau \to \infty}\left(\frac{\partial \varphi}{\partial \tau} = f(\varphi(\tau)) - \frac{\partial \varphi}{\partial t}\right)$$

Discretization with Euler implicit for physical time and Euler explicit with local time step size for pseudo time.

$$\lim_{\tau \to \infty}\left(\underbrace{\frac{\varphi(\tau+\Delta\tau)-\varphi(\tau)}{\Delta\tau}}_{\text{pseudo time}} = -\underbrace{\frac{\varphi(t+\Delta t)-\varphi(t)}{\Delta t}}_{\text{physical time}} + f(\varphi(\tau))\right)$$

$$\varphi(t+\Delta t) \rightarrow \varphi(\tau+\Delta\tau) \approx \varphi(\tau)$$
$$f(\varphi(\tau)) \rightarrow f(\varphi(t+\Delta t))$$

Ansys - CFX

Dual time stepping method — Local or physical time step size

1st order Euler (implicit) — Impose CFL if needed.

2nd order Euler (implicit)

BOUNDARY CONDITIONS

EXAMPLE: 1D Diffusion-advection equation

$$\frac{\partial\varphi}{\partial t} = f(\varphi) \quad , \quad f(\varphi(t)) = -U\frac{\partial\varphi}{\partial x} + \Gamma\frac{\partial^2\varphi}{\partial x^2}$$

Discrete equation:

$$\varphi_i^{n+1} = \varphi_i^n + \Delta t\left[-U\frac{\varphi_{i+1}^{n+1} - \varphi_{i-1}^{n+1}}{2\Delta x} + \Gamma\frac{\varphi_{i+1}^{n+1} - 2\varphi_i^{n+1} + \varphi_{i-1}^{n+1}}{\Delta x^2}\right]$$

Order by index of i

$$\varphi_{i-1}^{n+1}\left(-\frac{\Delta t U}{2\Delta x} - \frac{\Delta t\,\Gamma}{\Delta x^2}\right) + \varphi_i^{n+1}\left(1 + \frac{2\Delta t\,\Gamma}{\Delta x^2}\right) + \varphi_{i+1}^{n+1}\left(\frac{\Delta t U}{2\Delta x} - \frac{\Delta t\,\Gamma}{\Delta x^2}\right) = \varphi_i^n$$

Time: 1st order backward Euler (implicit)

$$\varphi^{n+1} = \varphi^n + f(t_{n+1}, \varphi^{n+1})\Delta t$$

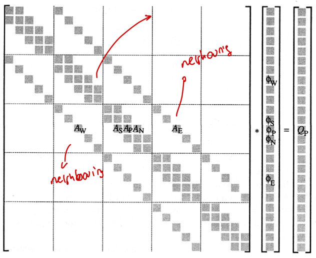
Space: 2nd order central differences (uniform grid)

$$\left(\frac{\partial\varphi}{\partial x}\right)_i = \frac{\varphi_{i+1} - \varphi_{i-1}}{2\Delta x} \qquad \left(\frac{\partial^2\varphi}{\partial x^2}\right)_i = \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{\Delta x^2}$$

central difference scheeme

The sumation of all advection and diffusion terms leads to a linear algebraic system

$$A_P\varphi_P^{n+1} + \sum_i A_i\varphi_i^{n+1} = Q_P$$



matrix structure for
5 x 5 grid

Set of algebraic eq. for every conservation law and every cell.

Index P: point where we approx PDE

Index i: runs over all cells

Ap, Ai: depend on grid, geometry and fluid properties.

Qp: includes all known terms at previous time level.

LINEARIZATION: How to treat nonlinear PDEs

$$\frac{\partial \underline{u}}{\partial t} + \nabla\cdot(\underline{u}\underline{u}) + \frac{1}{\rho}\nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{u} = 0$$

$$\nabla\cdot\underline{u} = 0$$

Navier Stokes are a system of nonlinear PDEs

Interaction of vortices and turbulence result from the quadratic nonlinearity.

Implicit time discretization methods are applied to a linearized version of N-S eqn.

$$A_P\varphi_P^{n+1} + \sum_i A_i\varphi_i^{n+1} = Q_P$$

Possible choices for linearizing:

(1) $u^{n+1}u^{n+1} \approx u^n u^{n+1}$

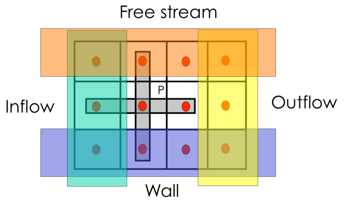
-> Error $(u^{n+1} - u^n)\, u^{n+1} = O(\Delta t)$

(2) $u^{n+1}u^{n+1} \approx 2\,u^n u^{n+1} - u^n u^n$

-> Error $(u^{n+1} - u^n)^2 = O(\Delta t^2)$

BOUNDARY CONDITIONS

Discrete operators approx the flow evolution in a cell as a function of the solution in neighbor cells. But, what do we do at the domain boundaries?



- N-S describe initial boundary value problems
- A well posed problem requires correct initial conditions and correct boundary conditions

TYPES OF BOUNDARY CONDITIONS:

1. Dirichlet boundary condition:

   Impose value of value (φ) on boundary

2. Neumann boundary condition:

   Impose gradient of variable $\left(\frac{\partial\varphi}{\partial n}\right)$ on the boundary

3. Robbin boundary condition:

   Combination of Dirichlet and Neumann

4. Periodic boundary condition:

   Chose bonding surface such that values of variable should be the same at two opposite walls.

Example: No slip wall

Value of fluid velocity is prescribed,
that is, Dirichlet boundary condition for velocity

$$\boxed{\underline{u}_{\text{fluid}} = \underline{u}_{\text{wall}}}$$

Advection/convection through wall is Null

No-slip condition plus incompressible continuity equation yield in relative wall coordinates.

$$\left.\frac{\partial u}{\partial x}\right|_{wall} = 0 \quad \rightarrow \quad c.eq.: \quad \cancel{\frac{\partial u}{\partial x}} + \frac{\partial v}{\partial y} = 0 \quad \rightarrow \quad \boxed{\left.\frac{\partial v}{\partial y}\right|_{wall} = 0}$$

Pressure gradient normal to wall is Null
Neumann boundary condition for pressure.

Viscous stress:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

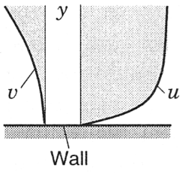
Viscous normal:

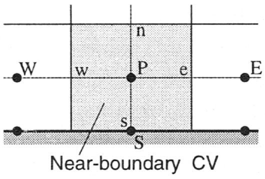
$$\boxed{\tau_{yy} = 2\mu\left.\frac{\partial v}{\partial y}\right|_{wall} = 0}$$

Viscous tangential: one side diffeences approxmation.

$$F_s^d = \int_{S_s} \tau_{xy}\, dS = \int_{S_s} \mu \frac{\partial u}{\partial y} dS \approx \mu S_s \frac{u_P - u_S}{y_P - y_S}$$

We can also use wall-stress model to approx viscous shear. Incorporate modeled shear stress as source term.





Example: Symmetry

Assume that a mirrored solution contines on the othe side of the bondary:

$$\boxed{\left.\frac{\partial p}{\partial y}\right|_{sym} = 0 \qquad \left.\frac{\partial u}{\partial y}\right|_{sym} = 0 \qquad v\big|_{sym} = 0}$$

$$\left.\frac{\partial v}{\partial y}\right|_{sym} \neq 0$$

No flow through the boundary

Tangential viscous shear stress of u-comp disapears.

Normal viscous is approximated as before

Example: Inflow

Supersonic:

Single-species N-S eq, 5 independent variables have to be specified as Dirichlet boundary conditions.

- density, 3x momentum, and energy
- pressure, temperature, velocity vector

Subsonic:

4 independent variables as Dirichlet boundary

- Density and 3xmomentum
- P_0, T_0, flow direction.

Example: Outflow

Supersonic:

No upstream influence. No dirichlet b.c.
Only Neumann b.c.

Subsonic / Incompressible

1 variable as Dirichlet b.c. usually static pressure.

Best Practice for Boundary Conditions

- Boundary conditions approximate the reality
- To limit unphysical effects, they should be located as far away as possible from region of interest.
- Interference flow can lead to unphysical oscillations and reflections.
- The choice of boundary conditions can significantly affect the convergence rate.
- Avoid "opening conditions" and never over-determine the problem.

Pressure Velocity Coupling

Compressible N-S eq.

Mass conservation (continuity equation)

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\underline{u}) = 0$$

Momentum conservation

$$\frac{\partial \rho \underline{u}}{\partial t} + \nabla\cdot(\rho\underline{u}\underline{u}) = -\nabla p + \nabla\cdot\underline{\tau} + \underline{F}$$

Energy conservation, with total energy $E = e + \frac{1}{2}\underline{u}^2$

$$\frac{\partial \rho E}{\partial t} + \nabla\cdot(\underline{u}\rho E) = -\nabla\cdot(\underline{u}p) + \nabla\cdot(\underline{u}\cdot\underline{\tau}) - \nabla\cdot(\underline{q}) + \underline{u}\underline{F}$$

Pressure (and temperature) follow from the equation of state,

e.g., for a perfect gas $p(\rho, e) = \rho RT = \rho R\frac{e}{c_V} = \rho e(\gamma - 1)$

Incompressible N-S eq.

Mass conservation (continuity equation)

$$\nabla\cdot\underline{u} = 0$$

Momentum conservation (constant density, no external forces F)

$$\frac{\partial \underline{u}}{\partial t} + \nabla\cdot(\underline{u}\underline{u}) = -\frac{1}{\rho}\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\underline{u}$$

The pressure p is not an independent variable anymore.
It is determined by the **Poisson equation**

$$\nabla^2 p = f(\underline{u})$$

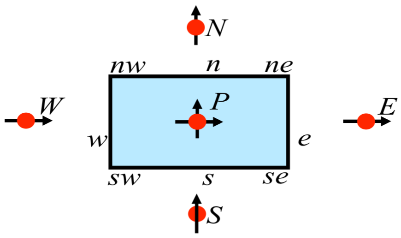
Discretization of the Pressure Poisson

Pressure term

$$-\int_S p\underline{n}\,\mathrm{d}S$$

Momentum equation x-dir

$$-\int_S p\underline{n}\,\mathrm{d}S = -p_e A_e + p_w A_w$$



Central differences scheme on uniform grid

$$\boxed{p_w - p_e = \frac{p_W + p_P}{2} - \frac{p_P + p_E}{2} = \frac{p_W - p_E}{2}}$$

Pressure is effectively discretized on coarser grid

Velocity and pressure of cell P are decoupled.

Pressure Poisson equation with CDS

$$\nabla^2 p = f(\underline{u})$$

$$\frac{p_{EE} - 2p_P + p_{WW}}{4\Delta x^2} + \frac{p_{NN} - 2p_P + p_{SS}}{4\Delta y^2} = f(\underline{u})$$

Neither mass nor momentum eq. couple pressure and speed at any point. Oscillations on pressure can occur and remain undetected by the numerical scheme!

RHIE & CHOW INTERPOLATION

> Pressure oscillations can occur and are not damped by the numerical scheme if we discretize pressure and velocity with CDS on a common grid.

> Coupling is restored by:

- different grids (staggered) for pressure and velocity.
- Special interpolation techniques for the pressure. ⟶ Rhie & Chow:

METHOD RHIE & CHOW:

Taylored interpolation method for the mass flux through cell surface.

Include the pressure gradient into interpolation rules for the advection velocity across finite-volume cell faces... NOT important probably.

EXAMPLE

Continuity equation is discretized as

$$\frac{\partial u}{\partial x} = 0$$

$\boxed{\text{CFX}}$

$$\frac{u_E - u_W}{2\Delta x} + C_{RC}\frac{p_{EE} - 4p_E + 6p_P - 4p_W + p_{WW}}{\Delta x^4} = 0$$

Thus we actually solve the modified equation

$$\frac{\partial u}{\partial x} + C_{RC}\left(\frac{\partial^4 p}{\partial x^4}\right) = 0$$

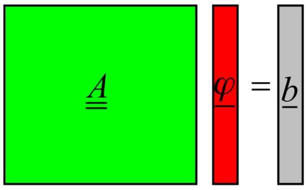
The additional term “detects“ oscillations. One can show that this term is dissipative and smoothes oscillations.

Disadvantage: the order of the method is reduced.

SOLUTION ALGORITHMS FOR LARGE LINEAR SYSTEMS

LINEAR ALGEBRAIC SYSTEM

Discretization leads to a linear algebraic system



A – coefficient matrix
φ – solution vector
b – constant „right hand side“

$$\underline{\underline{A}}\cdot\underline{\varphi} = \underline{b} \qquad \varphi = A^{-1}\cdot b$$

two solution methods
direct or iterative

Direct solution method

Invert the matrix A to obtain solution

$\underline{\varphi} = \underline{\underline{A}}^{-1} \cdot \underline{b}$ Problem: Memory requirements

Size of inverse matrix A^{-1},

A is typically sparse: a lot of zeros in the entries

but A^{-1} does not have to be.

Gauss-elimination

Multiply the first row of A with A_{21}/A_{11} and substract from second row.

➔ $\boxed{\underline{\underline{U}} \cdot \underline{\varphi} = \underline{q}}$

➔ **U** := upper triangular matrix

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix} \qquad U = \begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ 0 & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_{nn} \end{pmatrix}$$

try

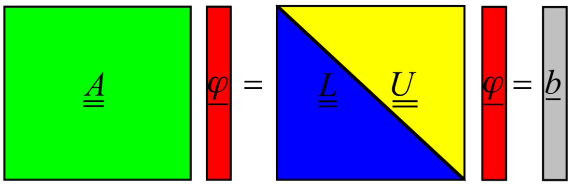
Last line of U has only one non zero entry

➔ back substitution $\boxed{\varphi_n = \frac{q_n}{U_{nn}}}$

Requires $O(N^3)$ operation for dense (non-sparse) matrices

LU (lower-upper) factorization

Matrix A can be decomposed in product of 2 triangular matrices L and U.



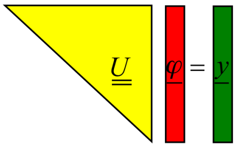
$\underline{\underline{A}}$
$\underline{\varphi}$
=
$\underline{\underline{L}}$
$\underline{\underline{U}}$
$\underline{\varphi}$
=
$\underline{b}$

$$\boxed{\underline{\underline{A}}\,\underline{\varphi} = \underline{\underline{L}}\,\underline{\underline{U}}\,\underline{\varphi} = \underline{b}}$$

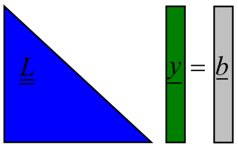
$$L_{ij} = A_{ij} - \sum_{k=1}^{j-1} L_{ik} U_{kj} \qquad U_{ij} = \frac{1}{L_{ii}}\left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} U_{kj}\right)$$

Can be splitted into two systems: we can solve a first and then b.

$(a) \Rightarrow \boxed{\underline{\underline{U}}\,\underline{\varphi} = \underline{y}}$



$(b) \Rightarrow \boxed{\underline{\underline{L}}\,\underline{y} = \underline{b}}$



Advantage: decomposition is independent of right hand side

Disadvantage: L and U are dense also if A is a sparse matrix

Direct methods

Exact solution with maximum „computer accuracy"

High computational cost

- Large memory requirements, e.g. O(N^2)
- Large operation count, e.g. O(N^3)

There is no need to solve the system exactly, because modeling and discretization errors are much larger than computer round-of error

Iterative Methods

Compute approximate solution through iteration

$$\underline{\varphi}^{n+1} = \text{function}\left(\underline{\varphi}^{n}\right)$$

Iterative Solution Methods

Split A as a part N that can be inverted easily and a part P for which computing the inverse is difficult

$$\underline{\underline{A}} = \underline{\underline{N}} - \underline{\underline{P}}$$

The linear system is re-written accordingly

$$\underline{\underline{N}} \cdot \underline{\varphi} = \underline{\underline{P}} \cdot \underline{\varphi} + \underline{b}$$

Use simple iteration scheme

$$\underline{\varphi}^{n+1} = \underline{\underline{N}}^{-1} \cdot \left(\underline{\underline{P}} \cdot \underline{\varphi}^{n} + \underline{b} \right)$$

Alternative iteration scheme, which is better in terms of round-off errors:

$$\underline{R}^{n} = \underline{\underline{A}}\underline{\varphi}^{n} - \underline{b} \quad \rightarrow \quad \Delta\underline{\varphi}^{n+1} = \underline{\underline{N}}^{-1} \cdot \underline{R}^{n} \quad \rightarrow \quad \underline{\varphi}^{n+1} = \underline{\varphi}^{n} + \Delta\underline{\varphi}^{n+1}$$

Residuum:

Deviation of the equation that is solved by the iterative solution from the exact equation (readily available)

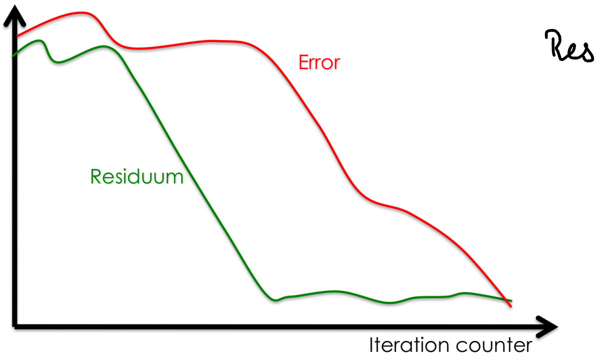
$$\underline{R}^{n} = \underline{\underline{A}} \cdot \underline{\varphi}^{n} - \underline{b}$$

Error:

Deviation from exact solution (exact solution is generally unknown)

$$\underline{\varepsilon}^{n} = \underline{\varphi}^{n} - \underline{\varphi}$$

They are related: $\underline{R}^n = \underline{\underline{A}} \cdot \underline{\varepsilon}^n$ can be solved, but it will be as complex as the problem.



Residuum converges before

Iteration matrix G: example

$$\underline{\underline{G}} = \underline{\underline{N}}^{-1}\underline{\underline{P}}$$

$$\underline{\varphi}^{n} = \underline{\underline{G}} \cdot \underline{\varphi}^{n-1} + \underline{\underline{N}}^{-1} \cdot \underline{b}$$

Change of error:

$$\underline{\varepsilon}^{n} = \underline{\varphi}^{n} - \underline{\varphi} = \underline{\underline{G}} \cdot \underline{\varepsilon}^{n-1} = \left(\underline{\underline{G}}\right)^{n} \cdot \underline{\varepsilon}^{0}$$

Necessary for convergent is that spectral radius of G is smaller than 1. $\rho\left(\underline{\underline{G}}\right) < 1$ faster the less radius.

Jacobi Method

Decomposition of A

- Diagonal elements into N
- All the rest goes into P

$$\underline{\varphi}^{n+1} = \underline{\underline{N}}^{-1} \cdot \left(\underline{\underline{P}} \cdot \underline{\varphi}^{n} + \underline{b} \right)$$

Spectral radius of iteration matrix for Poisson eq. on uniform grid

$$\rho\left(\underline{\underline{G}}\right) \approx 1 - \frac{\pi^2}{4}\left(\frac{1}{N^2} + \frac{1}{M^2}\right)$$

Characteristics:

Simple and robust

Very slow convergence for large problems.

Gauss-Seidel Method

Decomposition of A:

- Diagonal elements into N
- Elements in lower triangle so formally into N, however, not inverted but multiplied with φ^{n+1}
- Elements in upper triangle are in P and will be multiplied with φ^n.

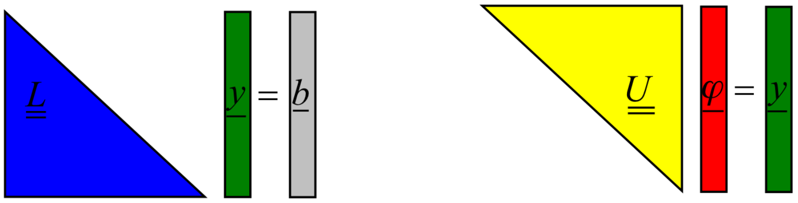
Implementation same as Jacobi, but need to remember φ^n

Spectral radius same as Jacobi

Accelerate it faster with successive over-relaxation

$$\rho\left(\underline{\underline{G}}\right) \approx 1 - \frac{\pi^2}{2}\left(\frac{1}{N^2} + \frac{1}{M^2}\right)$$

Incomplete LU factorization

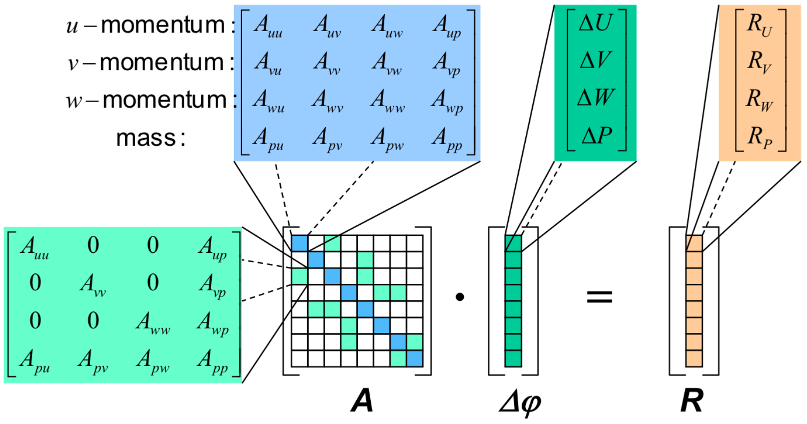

$\underline{\underline{L}}$
$\underline{y} = \underline{b}$
$\underline{\underline{U}}$
$\underline{\varphi} = \underline{y}$

- A is approximated by product of L and U.
- ILU only computes elements that are non-zero in A.
- L and U are sparse if A is sparse.

Compared with complete factorization: L and U are sparse as well. Memory is manageable.

CFX

Coupled ILU solver for u, v, w and p


$u-$momentum:
$v-$momentum:
$w-$momentum:
mass:
A_{uu} A_{uv} A_{uw} A_{up}
A_{vu} A_{vv} A_{vw} A_{vp}
A_{wu} A_{wv} A_{ww} A_{wp}
A_{pu} A_{pv} A_{pw} A_{pp}
ΔU ΔV ΔW ΔP
R_U R_V R_W R_P
A_{uu} 0 0 A_{up}
0 A_{vv} 0 A_{vp}
0 0 A_{ww} A_{wp}
A_{pu} A_{pv} A_{pw} A_{pp}
·
=
A
$\Delta\varphi$
R

Requires more memory and operations than uncoupled methods

More robust and faster convergence. Less iterations.

Multigrid Methods

OBSERVATION: Adapt the grid to match the error length and decrease computational time.

Iterative solvers quickly reduce errors with a small wave length, which is on the order of the cell size.

Errors with large wave length converge very slowly.

IDEA:

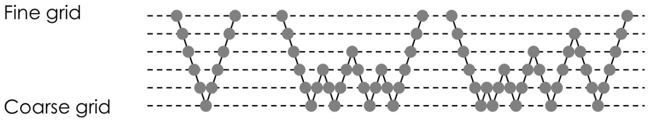
Converge acceleration by adapting grid to error wave length:

- Solve for part with large wave length on a coarser grid
- Interpolate solution back to fine grid.
- Solve for part with short wave length on fine grid.

Multigrid Method (continued)

- By projeting the error on a coarser grid, large wave lengths appear smaller to the solver.
- Algebraic system in coarser grid is much bigger thus computational time is decreased.
- Direct solvers are used on coarsest grid level.
- Information exchange:
  - Fine → coarse := Restriction
  - Coarse → fine := Prolongation

Many ways for walking through the grid hierarchy. Most popular are the V and W cycles.



- Method:

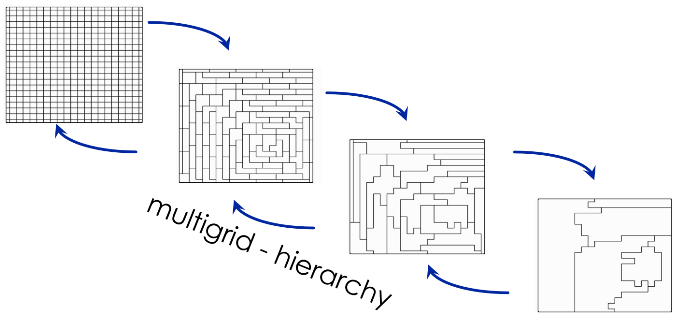
1. (one) iteration on original (fine) grid
2. Compute residuum, compare with convergence threshold
3. Restrict residuum to coarser grid(s)
4. Iteration of correction equation on coarse grid(s)
5. Prolongation of correction to fine grid
6. Update solution on fine grid
7. Back to step 2.

Multigrid Methods:

- Geometric Multigrid Coarsening is based on a user defined or automatically generated grid
- Algebraic Multigrid Coarsening is based on coefficient matrix.

Algebraic Multigrid

- The discretization is done only once on the original (fine) grid
- The discrete equations (coefficient matrix) for the coarser grids are obtained by summation from the fine grid.
- Absolute value of coefficients determines which lines and columns (cells) are merged



Irregular shapes

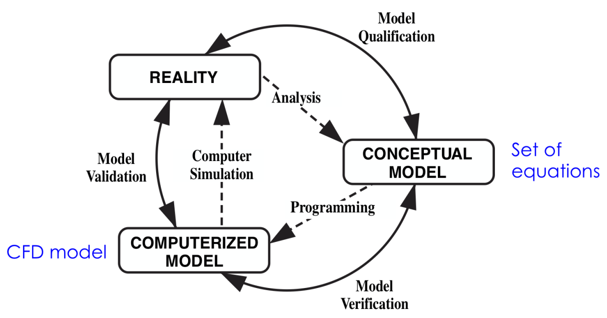
Verification & Validation

Verification comparison with known solutions

- Is the code doing what is supposed to do?
- Influence of numerical approximations
- Mathematical issues

Validation: comparisons with experimental data

- Simulation good representation of reality?
- Influence of model assumptions
- Physical issue.



Modelling error (validation)

$$
\begin{aligned}
\Delta &= u_{\text{nature}} - u_{\text{discrete}}\\
&= \underbrace{(u_{\text{nature}} - u_{\text{exp}})}_{E_1} + \underbrace{(u_{\text{exp}} - u_{\text{exact}})}_{E_2} + \underbrace{(u_{\text{exact}} - u_{\text{discrete}})}_{\substack{\|\\ (u_{\text{exact}} - u_{u_{h,\tau\to 0}}) + (u_{u_{h,\tau\to 0}} - u_{u_{h,\tau,I,c}})\\ E_3}}
\end{aligned}
$$

Numerical error (verification)

Types of Errors

Sources Numerical errors:

- Insufficient spatial discretisation convergence
- Insufficient temporal discretisation convergence
- Insufficient convergence of an iterative process
- Computer round-off

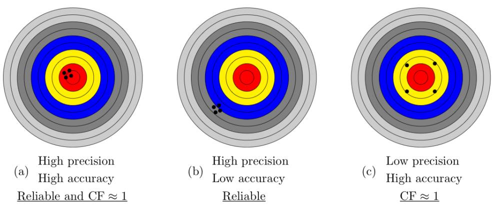
} Solution verification (acknowledged errors)

- Programming errors ⟶ Code verification (unacknowledged errors)

Therminology

Bias error: systematic offset
→ can be calibrated
Accuracy when the bias error is small

Precision error: random
→ can cancel when statistical averages are performed
Reliability when the precision error is small



(a) High precision High accuracy Reliable and CF ≈ 1
(b) High precision Low accuracy Reliable
(c) Low precision High accuracy CF ≈ 1

Convergence Error

- Iterative resolution of the system of equations results in an approximate solution with an unknown error
- The residual is only an indirect indication of the convergence error
- Iterative resolution of the system of equations results in an approximate solution with an unknown error
- The residual is only an indirect indication of the convergence error
- The error usually decreases slower than the residual
- Local time step accelerates the convergence but is insidious
- Better to do some iterations with a global ("physical") timestep

Residual 10^{-3} are not enough

Methodology for verification

1. Requires comparison with:
   - analytical solutions
   - semi analytical solutions
   - benchmark solutions
2. Code comparisons = verification activity!
3. Accuracy req. is often more stringent that in validation activities.
4. Document verification
5. Evidence gathered from the users.
6. Must be repeated for every change in the code

Consistent schemes $\|u_{\text{exact}} - u_{h,\tau}\| = \mathcal{O}(h^p, \tau^p)$

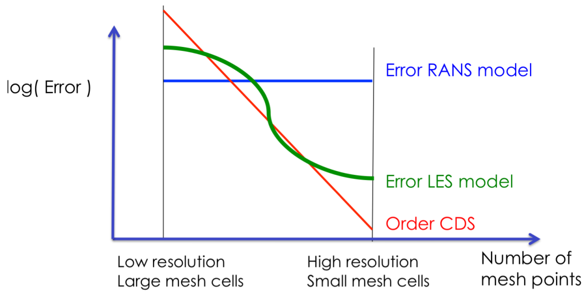
Discretization Error numerical diffusion.

Depends on grid quality, cell size and time step.

If possible select a high order method depends cell resolution

Run the simulation for different mesh resolutions and critically compare the results

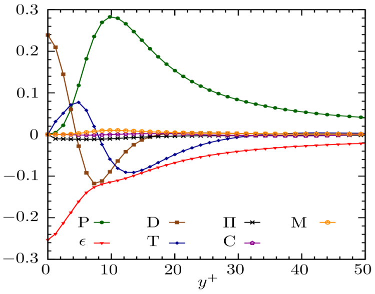
Grid convergence study enables the separation btw numerical truncation and error of physical model.



Computational Grid

The resolution of the boundary layers is important for:

- Friction coefficient
- Flow separation
- Transition and turbulence



The gradient of all relevant quantities should be resolved

The geometry should be resolved. When this is not possible, simplify the geometry before meshing.

The quality of the grid is critical for:

- Accuracy of discretization
- Convergence behaviour
- Timestep size

CFX:

Wall functions switched on if:

Laminar sub-layer is unresolved

An epsilon-based turbulence model is selected.

Critically analyse value of y^+, use $y^+ < 1$ at wall.

Error Estimation

Asymptotic convergence should be verified using at least 3 solutions $\Delta, \Delta/2, \Delta/4$

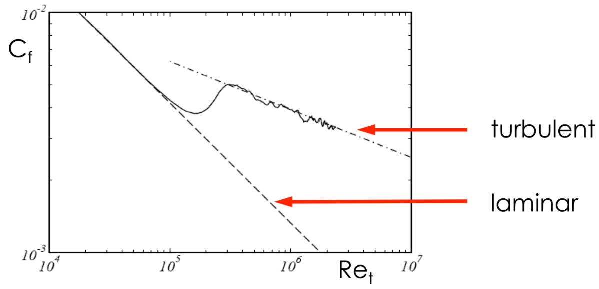
Singularities and discontinuities complicate the analysis.

Methodology for Validation

- Requires experimental data.
- Can be performed on subsystems and unit tests.

Modelling Error Turbulence:

Choice of model can affect the results. LES is good



The transition point btw laminar and turbulent flow should be well captured.

Summary

CFD can produce very good or very bad results depending on the way it is used (also CFX!)

Check model assumptions, including material parameters

Compare different turbulence models

Use y+=1 at the wall

Perform a grid convergence analysis

Critically analyse the boundary conditions

Computational domain as large as possible

Be sure that the simulation is converged, rather wait a little longer

Think about the result before starting the simulation

Gain experience and always be critical

Modelling Error: check consistency with model assumptions

Continuum hypothesis with micro/nano fluid flows

Incompressible flows:
maximum speed is smaller than a third of the speed of sound?

Compressible flows:
Equation of state for ideal gas valid? Dissociated gas?

Euler:
No shear layer effects, is friction really unimportant?

Stationary flow:
No significant slow, large scale phenomena?

Modelling Error - Thermodynamics

- The standard parameters are valid under dry and lukewarm conditions.
- In reality, humid air and temperature changes.
- Empirical laws have a working range on temp
- Ideal Gas also has a range

Domain Size and Boundary Conditions

CFD b.c. are not realistic

Experiments have errors

Inflow and far field critical

Choice of b.c influences convergence