Introduction to Abaqus/CFD

SIMULIA

Day 1

- Lecture 1 Review of CFD Fundamentals
- Lecture 2 Introduction
- Lecture 3 Getting Started with Abaqus/CFD
- Workshop 1 Unsteady flow across a circular cylinder
- Lecture 4 CFD Modeling Techniques – Part 1

SIMULIA

Day 2

- Lecture 5 CFD Modeling Techniques – Part 2
- Lecture 6 Getting Started with FSI Using Abaqus/CFD
- Workshop 1 Unsteady flow across a circular cylinder (continued)
- Lecture 7 FSI Modeling Techniques
- Workshop 2 Heat transfer analysis of a component-mounted electronic circuit board
- Lecture 8 Postprocessing CFD/FSI Analyses
- Workshop 2 Heat transfer analysis of a component-mounted electronic circuit board (continued)

Legal Notices

Revision Status

Lecture 1	5/10	New for 6.10
Lecture 2	5/10	New for 6.10
Lecture 3	5/10	New for 6.10
Lecture 4	5/10	New for 6.10
Lecture 5	5/10	New for 6.10
Lecture 6	5/10	New for 6.10
Lecture 7	5/10	New for 6.10
Lecture 8	5/10	New for 6.10
Workshop 1	5/10	New for 6.10
Workshop 2	5/10	New for 6.10

Notes

Notes

Review of CFD Fundamentals

Lecture 1

Overview

- Computational Solid Mechanics (CSM) versus Computational Fluid Dynamics (CFD)
- CFD Basics
- Governing Equations
- Heat Transfer in Fluid Mechanics
- Non-dimensional Quantities in CFD
- Initial and Boundary Conditions
- Solution of Governing Equations
- Turbulence Modeling

Overview

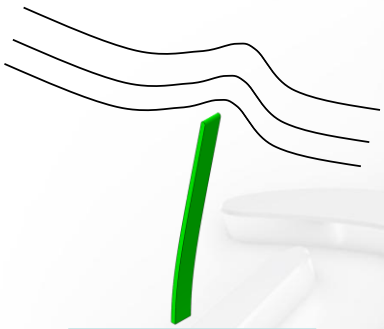
- **This lecture is optional.**
- **It aims to introduce the necessary fluid dynamics concepts and quantities that are relevant to the Abaqus functionality that is presented in the subsequent lectures.**
  - If you are already familiar with these concepts, this lecture may be omitted.

Computational Solid Mechanics (CSM) versus Computational Fluid Dynamics (CFD)

© DASSAULT SYSTEMES

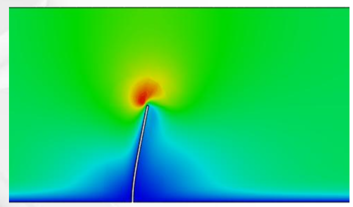
CSM vs. CFD

Engineering problem → Computational tools → Answers



Baffle in wind

Computational solid mechanics (CSM)



Computational fluid dynamics (CFD)

Deformation of structure, Structural integrity

⇕

Flow patterns

CSM vs. CFD

Computational Solid Mechanics

Physics | Material properties | Computational method | Framework | Boundary conditions, Loads | Analysis types

Physics

Governing equations
- Equations of motion
  - Elasticity

Sources of nonlinearity
- Material
  - Nonlinear elasticity
  - Plasticity
  - Material damage
- Geometric
  - Large deformations
  - Large rotations
  - Structural instability (buckling)
- Boundary
  - Contact

Material properties

Elasticity (linear, nonlinear)
Plasticity
Conductivity
Specific Heat

Computational method

Method
Finite element

Discretization
FE mesh
"Stress concentrations"
Interpolation order

Model sizes
Maximum ~ 2-3 M
Routine analysis < 0.5 M

Solution of discretized equations
Direct solvers

Framework

Primarily Lagrangian
- Mesh deforms with material

Arbitrary Lagrangian-Eulerian (ALE)
- Mesh smoothing for large deformation

Eulerian
- Extreme deformation cases (CEL)

Boundary conditions, Loads

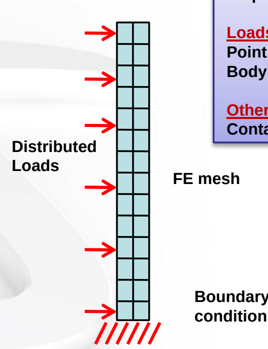
Boundary conditions
Displacement, velocity, acceleration

Loads
Point and distributed loads
Body forces

Other loading
Contact interactions

Analysis types

Static
Dynamic
Linear perturbation
- Frequency



CSM vs. CFD

Computational Fluid Dynamics

Physics

Governing equations
Equation of motion
Navier-Stokes (N-S) equations
Incompressible or Compressible
Specialized versions
Stokes flow

Sources of nonlinearity
N-S equations are fundamentally nonlinear due to advective terms

Material nonlinearity
- Non-Newtonian viscosity

Boundary nonlinearity
- FSI

Material properties

- Viscosity (Newtonian, Non-Newtonian)
- Conductivity
- Specific Heat

Computational method

Method:
Finite volume
Finite difference
Finite element
Hybrid methods

Discretization:
Finite volume mesh
“Boundary layers”
Treatment of various terms in equations

Model sizes
Maximum ~ 1000M
Routine analysis ~ 0.5-1.5 M

Solution of discretized equations
Iterative solvers

Framework

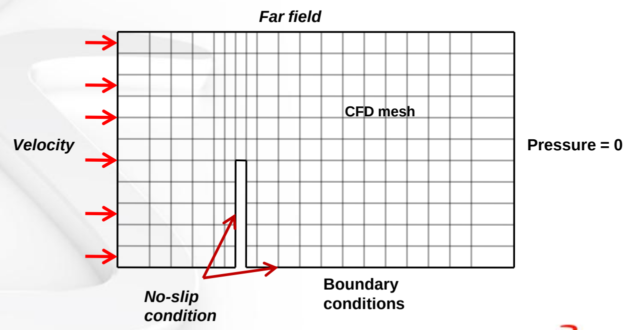
Primarily Eulerian (Mesh is fixed)
Arbitrary Lagrangian-Eulerian (ALE)
Fluid-structure interaction

Boundary conditions, Loads

Velocity, pressure BCs
Body forces

Analysis types

Steady-state
Transient



CSM vs. CFD

- **Key differentiators**

Feature		Computational solid mechanics	Computational fluid mechanics
Physics	Equations	General equations of motion	Equations of motion reduce to incompressible/compressible Navier-Stokes equations
	Computational Framework	Lagrangian, Arbitrary Lagrangian-Eulerian, Eulerian	Eulerian, Arbitrary Lagrangian-Eulerian
	Nonlinearity	Material, geometric and boundary	Fundamentally nonlinear (advective terms), non-Newtonian viscosities, Fluid-structure interaction
	Material properties	Solids: Elasticity, plasticity, etc.	Fluids: Viscosity
Computational method	Method	Finite element	Finite volume, Finite element, Hybrid
	Feature	Stress concentration, Interpolation order	Boundary layers, Treatment of various terms in N-S equations
	Model sizes	Maximum ~ 2-3 M Routine analysis < 0.5 M	Maximum ~ 1000M Routine analysis ~ 0.5-1.5 M
	Equation solution method	Primarily direct, Iterative	Primarily iterative

CFD Basics

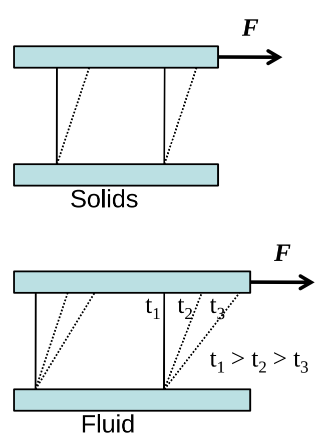
SIMULIA

CFD Basics

Fluid Mechanics
- Discipline within the field of applied mechanics concerned with the behavior of fluids

Fluids
- Liquids or gases
- Deform continuously under the application of shear stresses

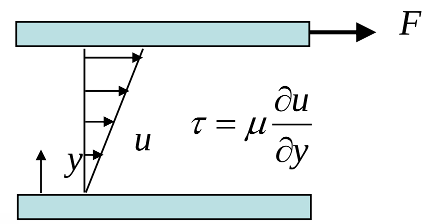
Computational Fluid Dynamics
- Like structural mechanics, very few fluid mechanics problems can be solved analytically
- Computational (i.e., numerical) method for fluid mechanics



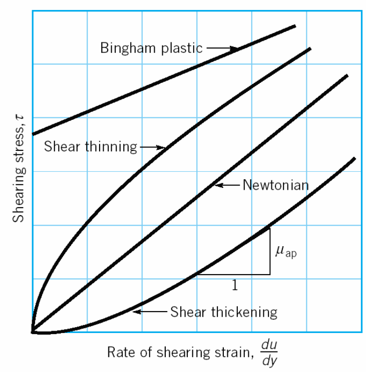
SIMULIA

CFD Basics

- **Viscosity**
  - Fluid property that relates shear stress to the rate of deformation

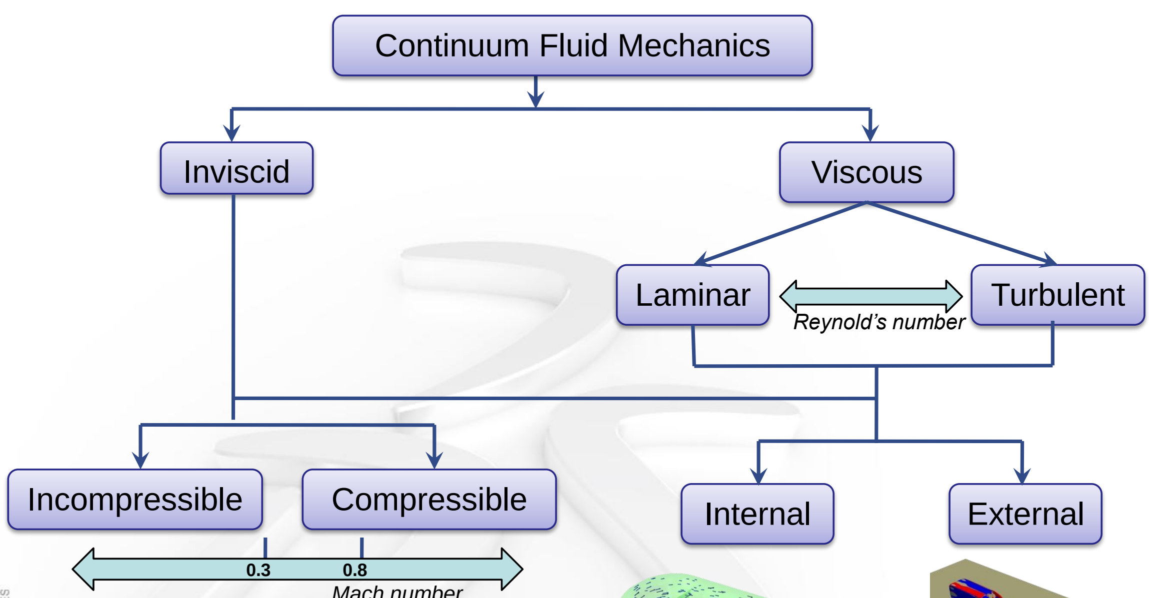


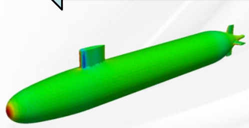
- **Newtonian/Non-Newtonian Fluids**
  - Newtonian fluids:
    - Shearing stress varies linearly with the rate of shearing strain
    - E.g. air and water
  - Non-Newtonian fluids
    - Distinguished by how viscosity changes with shearing rate
    - E.g. blood or alcohol

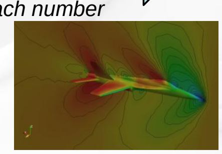


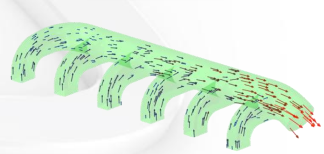
Munson, B. R., D. F. Young, and T. H. Okiishi, *Fundamentals of Fluid Mechanics*, John Wiley & Sons, 2002

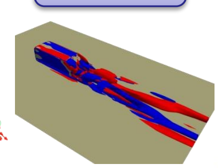
CFD Basics











CFD Basics

- **Inviscid vs. viscous flows**

Inviscid flows	Viscous flows
• Effect of viscosity is neglected	• Effect of viscosity is included • Especially important in flows close to a solid boundary

Reynolds number

$$\mathrm{Re} = \frac{Inertial\ forces}{Viscous\ forces} = \frac{\rho V L}{\mu}$$

L: Characteristic length scale of the flow
V: Characteristic velocity

Increasing Reynolds number →

Viscous effects dominate

Stokes flow, Re << 1

Inertial effects dominate

SIMULIA

CFD Basics

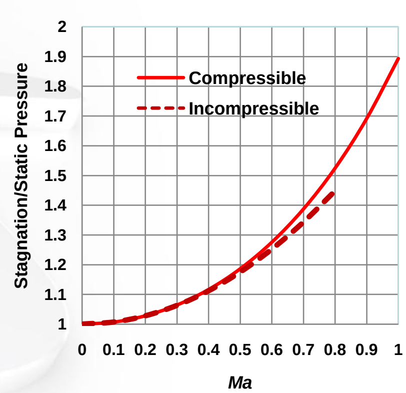
- **Incompressible vs. compressible flows**

Incompressible flows	Compressible flows
• Velocity field is divergence free • Energy contained in acoustic waves is small relative to the energy transported by advection • Example: Flow of liquids are often treated as incompressible	• Density variations within the flow are not negligible • Example: Flow of gases are often compressible

Mach number

$$Ma = \frac{Flow\ speed}{Local\ speed\ of\ sound} = \frac{V}{c}$$

- For $Ma < 0.3$, the variation is < 2 %
- For $Ma < 0.45$, the variation is < 5 %
- $Ma \approx 0.3$ **is considered the limit for incompressible flow**

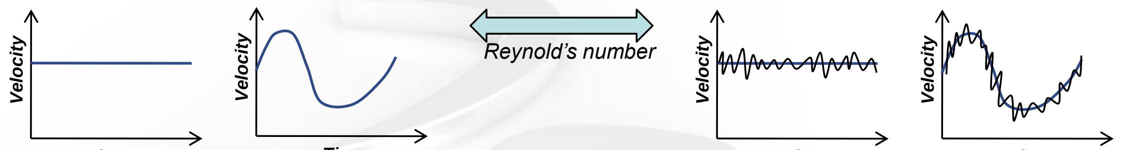


SIMULIA

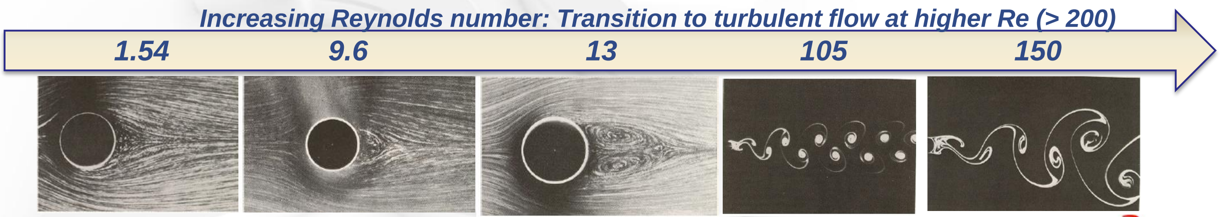
CFD Basics

- **Laminar vs. turbulent flows**

Laminar flow	Turbulent flows
• Smooth motion in layers (laminae) • No gross mixing of flows (slow dispersion due to molecular motion only)	• Random three dimensional motion • Macroscopic mixing • Unsteady (mean flow can be steady or unsteady)



Steady laminar flow — Unsteady laminar flow — Steady turbulent flow — Unsteady turbulent flow

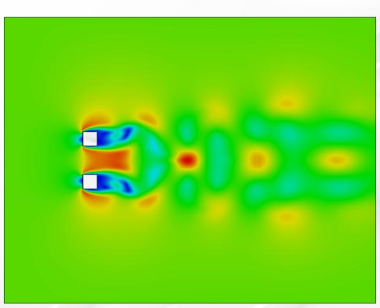


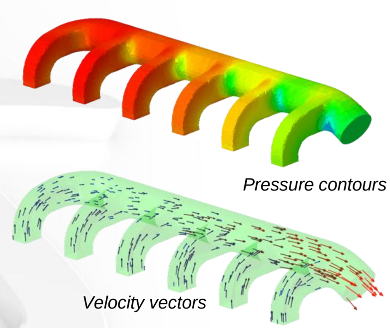
Munson, B. R., D. F. Young, and T. H. Okiishi, *Fundamentals of Fluid Mechanics*, John Wiley & Sons, 2002

CFD Basics

- **Internal vs. external flows**

External flows	Internal flow
• Fluid flow over external surface of an object	• Fluid flow that passes through confined solid boundaries


Velocity contours



Flow around Obstacles — ***Flow inside Engine Manifold***

Governing Equations

SIMULIA

Governing Equations

Fundamental Principles of Mechanics

Conservation of Mass | Conservation of Momentum | First and Second Law of Thermodynamics

Constitutive Assumption for Fluids

1) Newtonian fluid (viscosity is constant)
2) Non-Newtonian fluid

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\left(\boldsymbol{D} - \frac{1}{3}(\nabla\cdot\mathbf{v})\mathbf{I}\right)$$

$$\boldsymbol{D} = \frac{1}{2}\left((\nabla\mathbf{v}) + (\nabla\mathbf{v})^T\right)$$

Rate of deformation tensor

Fluid Mechanics ⇨

- Equation of Continuity
- Navier-Stokes Equation
  1) Generalized - Non-Newtonian fluids
  2) Compressible and incompressible forms
- Energy Equation

SIMULIA

Governing Equations

- **Continuity Equation**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\, \mathbf{v}) = 0$$

$$\nabla \cdot \mathbf{v} = 0$$ *Incompressible flows*

- **Conservation of momentum**

Mechanics

Constitutive law

Fluid Mechanics

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = \underbrace{\nabla\cdot\boldsymbol{\sigma}}_{\text{Surface forces}} + \underbrace{\rho\mathbf{g}}_{\text{Body forces}}$$

$$\boldsymbol{\sigma} = -\left(p + \frac{2}{3}\mu(\nabla\cdot\mathbf{v})\right)\mathbf{I} + 2\mu\boldsymbol{D}$$

Newtonian fluid (Constant viscosity)

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu\nabla^2\mathbf{v} + \frac{\mu}{3}\nabla(\nabla\cdot\mathbf{v}) + \rho\mathbf{g}$$

Compressible Navier-Stokes Equation (Viscous fluid with constant fluid viscosity)

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu\nabla^2\mathbf{v} + \rho\mathbf{g}$$

Incompressible Navier-Stokes Equation (Viscous fluid with constant fluid viscosity)

Governing Equations

- **Conservation of energy**

$$\frac{\partial e}{\partial t} + \mathbf{v}\cdot\nabla e = \boldsymbol{\sigma} : \boldsymbol{D} + \rho r - \nabla\cdot\boldsymbol{q}$$

Mechanics

Constitutive law

Fluid Mechanics

$$\boldsymbol{q} = -k\nabla\theta$$ *Fourier law for heat conduction*

$$F(p, \rho, \theta) = 0$$ *Equation of state*

$$e = e(\theta, \rho)$$

$$p = \rho R\theta$$

$$e = c_v\theta$$

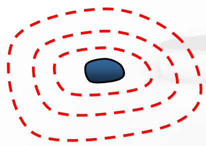
Ideal gas

Governing Equations

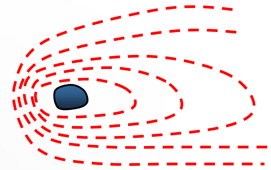
- **Some terminology**

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\underbrace{\mathbf{v}\cdot\nabla\mathbf{v}}_{\text{Advective terms}}\right)=-\nabla p+\underbrace{\mu\nabla^2\mathbf{v}}_{\text{Diffusive terms}}+\rho\mathbf{g}$$

Incompressible Navier-Stokes Equation



*Diffusion (in still flow, **V** = 0)*



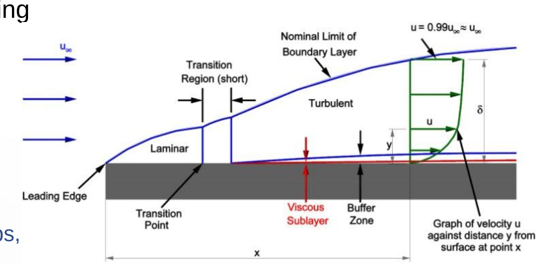
Diffusion and Advection

- Advective terms make Navier-Stokes equations fundamentally nonlinear
  - Except in the special case of creeping flows (Stokes flow), this cannot be neglected
- Diffusion dominates at low Reynolds number while advection dominates at high Reynolds number

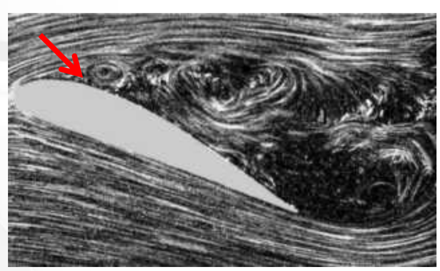
SIMULIA

Flow Features

- **Boundary layers**
  - Layer of fluid in the immediate vicinity of a bounding surface
  - Fluid velocity changes from zero at the surface to free stream velocity in this thin layer
  - Can be laminar or turbulent (depends on Re)
  - Affects engineering quantities of interest
    - Drag on bodies, wall shear stresses, pressure drops, heat transfer



- **Flow separation**
  - Occurs when a boundary layer travels far enough against an adverse pressure gradient such that the speed of the boundary layer falls to zero
    - Detached flow in forms of eddies & vortices
    - Increased drag on bodies
    - Delaying the onset of flow separation is a design challenge



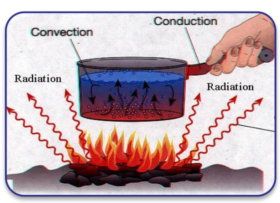
SIMULIA

Heat Transfer in Fluid Mechanics

SIMULIA

Heat Transfer in Fluid Mechanics

Conduction
- Diffusion process; heat transfer through direct contact



Radiation
- Heat transfer through electromagnetic waves

Convection
- Associated with fluid motion
- Natural convection – "Hot gas rises"
- Forced convection – Forced fluid motion

Modeling Natural Convection

Temperature differential causes change in density; lighter fluid rises	Requires compressible N-S equations **Source term** proportional to density and gravity	**Approximate** and model as incompressible flow (Boussinesq approximation)	Linearize the source term and relate change in density to change in temperature

$$\mathbf{S} = \rho \mathbf{g}$$

$$\beta = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_P$$

$$(\rho - \rho_0)\mathbf{g} \approx -\rho_0 \beta (T - T_0)\mathbf{g}$$

SIMULIA

Non-dimensional Quantities in CFD

Non-dimensional Quantities in CFD

Name	Definition	Physical significance	Area of applicability	Notes
Biot number (Bi)	hL/k_{solid}	Ratio of internal thermal resistance of solid to fluid thermal resistance	Heat transfer between solid and fluid	• Bi <<1: Heat conduction inside the body is much faster than the heat convection away from its surface; use temperature BC at solid walls • Bi >> 1 : Need to consider spatial variation of temperature within solid; include conduction in solid
Grashof number (Gr)	$\frac{gL^3\beta\Delta T}{\nu^2}$	Ratio of buoyancy to viscous forces	Natural convection flows	• Indicates strength of natural convection and also limit for transition to turbulent flows for natural convection
Mach number (Ma)	V/c	Ratio of velocity of flow to velocity of sound	Compressible flows	• Indicates if the flow is compressible • Ma < 0.3: Incompressible • 0.3 < Ma < 0.8 Weakly compressible • Ma > 0.8 Compressible

Non-dimensional Quantities in CFD

Name	Definition	Physical significance	Area of applicability	Notes
Nusselt number (Nu)	hL/k_{fluid}	Ratio of convective heat transfer to conduction in a fluid slab of length *L*	Convective heat transfer	• Increased convection and heat removal at higher Nu • Nu ~1: Laminar flow • Higher values for turbulent flows
Prandtl number (Nu)	$\mu c_p/k$	Ratio of molecular momentum and thermal diffusivity	Forced and natural convection	• Pr << 1: heat diffuses quicker than velocity • Relative thickness of thermal and velocity boundary layers
Rayleigh number (Ra)	$Gr\,\mathrm{Pr}$	Modified Grashof number	Natural convection	• Higher value indicates vigorous natural convection
Reynolds number (Re)	$\dfrac{\rho VL}{\mu}$	Ratio of inertial to viscous forces	Dynamic similarity	• Transition from laminar to turbulent flow • Dynamic similarity between experiments
Strouhal number (Sr)	$\dfrac{Lf}{V}$	Ratio of velocity of vibration LF to the velocity of the fluid	Vortex shedding	• Oscillating flows, vortex shedding

Initial and Boundary Conditions

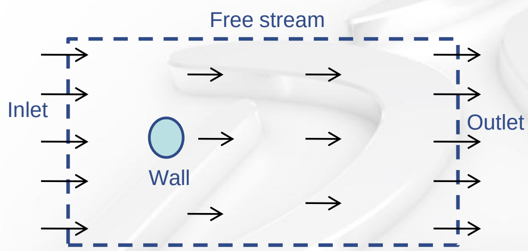
Initial and Boundary Conditions

- Governing equations require initial and boundary conditions
- Initial conditions define conditions at start up (required for transient problems)
  - Pressure (only compressible flows)
  - Velocity
  - Temperature
  - Turbulence quantities
- Boundary conditions define conditions at solution domain boundaries
  - Pressure
  - Velocity
  - Temperature
  - Quantities specific to turbulence models
    - Turbulent viscosity
    - Wall-normal distance function

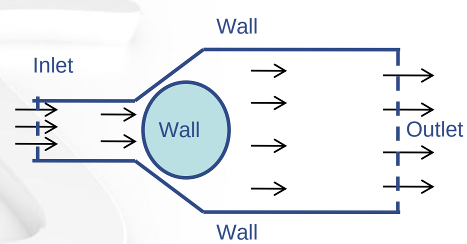
Initial and Boundary Conditions

Where do I need boundary conditions?

Flow inlet or outlet regions	**Physical wall (stationary or moving) region**	**Far field region**	**Zones where flow is abstracted**	
Fluid enters or leaves the flow solution domain	Fluid is constrained to "stick" to an obstruction, "No-Slip, no-penetration"	Imaginary flow solution domain boundaries	Approximating three dimensional flow as two-dimensional	Symmetry conditions



External flow



Internal flow

Solution of Governing Equations

Solution of Governing Equations

Differential equations

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\mathbf{v} + \mathbf{f} = 0$$

$$\nabla\cdot\left(\frac{1}{\rho}\nabla p\right) = \nabla\cdot\left(\mathbf{f} - \mathbf{v}\cdot\nabla\mathbf{v}\right)$$

Pressure Poisson's Equation (PPE)

- Solving the PPE and momentum equations together is the same as solving the momentum and continuity equations if initial and boundary conditions are compatible

Hybrid finite-element/finite-volume discretization

$$\oint_{\forall}\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} + \frac{1}{\rho}\nabla p - \frac{\mu}{\rho}\nabla^2\mathbf{v} - \mathbf{f}\right)\varphi_i d\forall = 0$$

$$\int_{\forall}\left(\nabla\cdot\left(\frac{1}{\rho}\nabla p\right) - \nabla\cdot\left(\mathbf{f} - \mathbf{v}\cdot\nabla\mathbf{v}\right)\right)\psi_i d\forall = 0$$

- A weight function of 1 in the momentum equation recovers finite volume formulation
- Locally conservative

Spatial discretization

$$M\dot{\mathbf{v}} + A(\mathbf{v})\mathbf{v} + K\mathbf{v} + Gp = \mathbf{F}$$

$$D\mathbf{v} = g$$

$$M\dot{\mathbf{v}} + A(\mathbf{v})\mathbf{v} + K\mathbf{v} + Gp = \mathbf{F}$$

$$L\mathbf{p} = DM^{-1}\left(\mathbf{F} - K\mathbf{v} - A(\mathbf{v})\mathbf{v}\right) - \dot{g}$$

- Flux evaluations using least squares method

M : Mass matrix, A(**v**) : Advection operator, K : Viscous diffusion operator
G : Gradient operator, D : Divergence operator, L: Pressure-Poisson operator, **F** : body force

Solution of Governing Equations

Temporal discretization

$$\frac{\mathbf{v}_{n+1}-\mathbf{v}_n}{\Delta t} = (1-\theta)\dot{\mathbf{v}}_n + \theta\dot{\mathbf{v}}_{n+1}$$

$\theta = 1$: Backward Euler

$\theta = \frac{2}{3}$: Galerkin

$\theta = 0.5$: Trapezoid (Crank - Nicolson)

Velocities,
Transport quantities (Temperature, turbulence & species transport)

Solution quantities

1. Velocity (three components)
2. Pressure
3. Temperature (if thermal effects are included)
4. Transport quantities (turbulence, species concentrations, etc.)

Linear system of equations, Iterative linear solvers to find solution

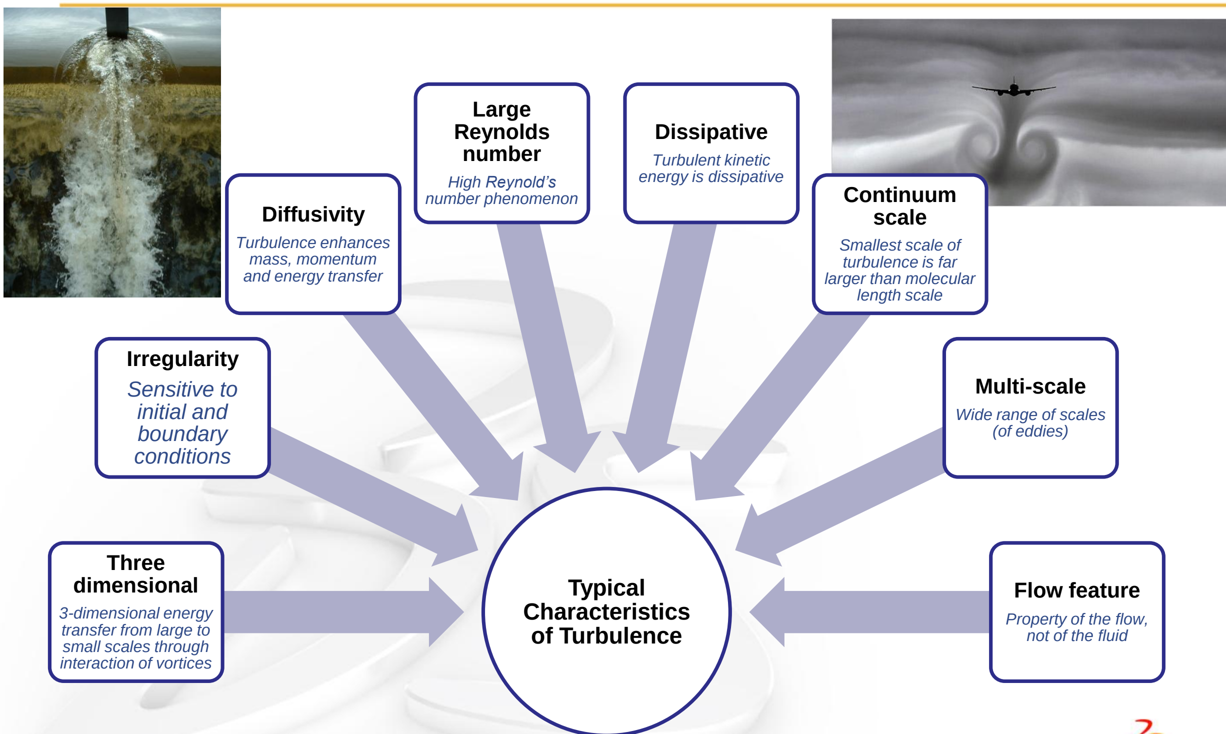
$$Ax = b$$

$$\frac{r}{\|b\|} < \delta \qquad r = \|b - Ax^{i+1}\|$$

Convergence criterion

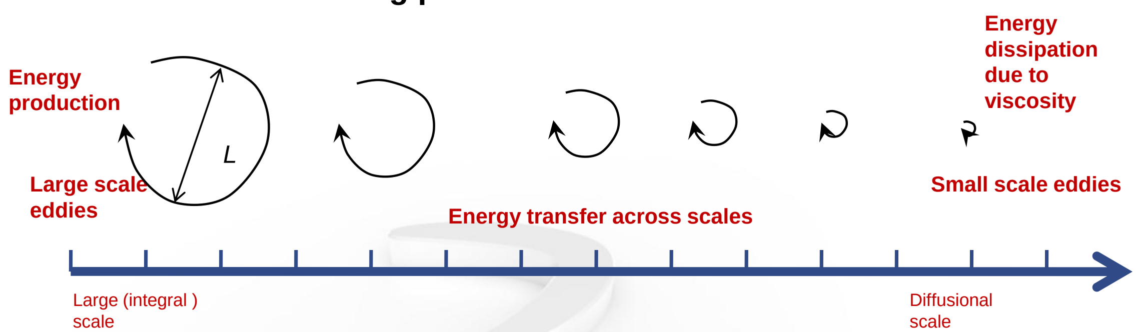
Turbulence Modeling

Turbulence Modeling



Turbulence Modeling

- **Turbulent cascading process**



- Turbulence consists of a continuous spectrum of scales ranging from large to small
  - Local swirling motion of eddies of varied sizes (length scales *L*)
- Turbulent cascading process: Kinetic energy is transferred from larger eddies to smaller eddies, where it is dissipated as heat through molecular viscosity
- Resolving various turbulent length scales (*L*) is key to turbulence modeling
  - Navier-Stokes equations can resolve whole spectrum of turbulence scales but will need a very fine mesh (mesh size ~ length scale resolved)

Turbulence Modeling

- **Approaches to Turbulence Modeling**

Reynolds Averaged Navier-Stokes (RANS)	Large Eddy Simulation (LES)	Hybrid RANS/LES approach	Direct Numerical Simulation (DNS)

• Solve averaged N-S equations
• Models most scales
• Computationally efficient
• Widely used for industrial applications
• Unsteady RANS
• Examples:
  • Eddy viscosity models
  • Reynolds stress models
	• Resolves large scale eddies and models small scale eddies
	• Computationally more expensive than RANS approach
		• RANS approach close to solid boundaries
		• LES away from walls (for detached eddies)
		• Example: Detached eddy simulation (DES)
			• N-S equations can resolve all turbulence scales but would require very fine meshes
			• Computationally intensive and hence not often used for industrial applications

Increasing computational cost

→

More turbulence scales resolved

Turbulence Modeling

- **Reynolds Averaged Navier-Stokes (RANS) Approach**

Introduce Reynold's decomposition of variables into the flow equations

↓

Perform time averaging

↓

Obtain RANS equations for incompressible flow

$$u_i(\mathbf{x},t) = U_i(\mathbf{x}) + u_i'(\mathbf{x},t)$$

Mean velocity *Fluctuating velocity*

$$U_i(\mathbf{x}) = \lim_{t\to\infty} \frac{1}{T}\int_t^{t+T} u_i(\mathbf{x},t)\,dt,$$

$$\overline{u_i'}(\mathbf{x},t) = 0$$

$$\frac{\partial U_i}{\partial x_i} = 0$$

Turbulence Modeling

- **Reynolds Averaged Navier-Stokes (RANS) Approach (cont'd)**

Obtain RANS equations for incompressible flow

↓

New "**Stress like**" terms arise

↓

Turbulent closure problem - more unknowns (three velocities, pressure and six Reynolds stresses) than equations (three momentum and continuity equation)

↓

Need ***additional*** *equations* to model Reynolds stresses

$$\rho \frac{\partial U_i}{\partial t} + \rho U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \rho \overline{u'_j u'_i}\right)$$

$$\tau_{ij} = -\overline{u'_i u'_j}$$

Specific Reynolds-stress tensor

Model Reynolds stresses

1. Eddy viscosity models
2. Reynolds stress models

Turbulence Modeling

- **Eddy Viscosity Models**
  - Model Reynolds stresses from an eddy viscosity and mean strain rate - **"Boussinesq Approximation"**
  - Isotropic
  - Eddy viscosity is a function of turbulence length and time scales
    - Not constant

Effective turbulent viscosity

$$\tau_{ij} = -\rho\overline{u'_i u'_j} = 2\mu_t S_{ij} - \frac{2}{3}\rho k \delta_{ij}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \quad 2\rho k = \tau_{ii} = -\rho\overline{u'_i u'_i}$$

k: Like "normal stress"

Eddy Viscosity Models			
Algebraic Turbulence Models	One Equation Turbulence Models	Two Equation Turbulence Models	Others
Examples: Prandtl Mixing Length Models, Cebeci-Smith Model, Baldwin-Lomax Model	Examples: **Spalart-Allmaras**, Baldwin-Barth	Examples: k-ε model, k-ω model	

Turbulence Modeling

- **Spalart-Allmaras Turbulence Model**
  - One equation turbulence model
  - Differential equation to determine $\tilde{\nu}$

$$\left\{\begin{array}{l}\nu_t : \text{Effective turbulent kinematic viscosity} \\ \tilde{\nu} : \text{Turbulent kinematic viscosity}\end{array}\right.$$

$$\nu_t = \tilde{\nu} f_{v1}\left(\tilde{\nu}/\nu\right), \quad f_{v1}(\chi) := \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}$$

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{v}.\nabla \tilde{\nu} = \frac{1}{\sigma}\underbrace{\left(\nabla \cdot \left((\tilde{\nu} + \nu) \nabla \tilde{\nu} \right) + c_{b2} \left|\nabla \tilde{\nu}\right|^2 \right)}_{\text{Diffusion}} + \underbrace{c_{b1} \tilde{S}(\tilde{\nu}) \tilde{\nu}}_{\text{Production}} - \underbrace{c_{w1} f_w(\tilde{\nu}) \left(\frac{\tilde{\nu}}{d}\right)^2}_{\text{Dissipation}}$$

Normal distance from wall is needed

$$\tilde{S}(\tilde{\nu}) = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}\left(\tilde{\nu}/\nu\right), \quad S = \left\|\nabla \times \mathbf{v}\right\|_2$$

$$f_{v2}(x) := 1 - \frac{x}{1 + x f_{v1}(x)}, \qquad f_{v1}(x) := \frac{x^3}{x^3 + c_{v1}^3}$$

$$f_w(g) := g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}, \quad g(r) := r + c_{w2}\left(r^6 - r\right), \quad r(\tilde{\nu}) = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$$

Constants

$c_{b1} = 0.1355$

$c_{b2} = 0.622$

$c_{v1} = 7.1$

$\sigma = 2/3$

$c_{w1} = \dfrac{c_{b1}}{\kappa} + \dfrac{1 + c_{b2}}{\sigma}$

$c_{w2} = 0.3$

$c_{w3} = 2$

$\kappa = 0.41$

$c_{v2} = 5$

Turbulence Modeling

- **Spalart-Allmaras Turbulence Model**
  - Required boundary conditions for $\tilde{\nu}$
    - At walls: $\tilde{\nu} = \mathbf{0}$
    - Inlet turbulence needs to be specified
  - Initial conditions required for $\tilde{\nu}$ for transient problems
  - Does not require near-wall treatment
    - Formulation ensures correct near-wall behavior when integrated down to wall
  - Captures accurate boundary layers if near-wall meshes are resolved ($y^+ \approx \mathbf{3}$)
  - Usage:
    - Attached flows with no or mild separation
    - Can not be used with highly rotational flows
    - Primarily developed for external aerodynamics

Turbulence Modeling

- **Near-wall treatment in turbulent flows**

Why do walls affect turbulent flows?

No-slip condition at the walls

Main source of turbulence (large gradients in temperature and velocity field) occur near the wall

Why bother?

Affects engineering quantities of interest such as:
- Drag on bodies
- Wall shear stresses
- Pressure drops
- Heat transfer from walls

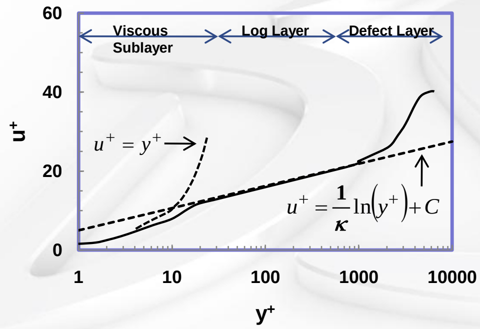
Near-wall turbulent flow structure

Law of the wall

1) Viscous sublayer

2) Log layer

3) Fully turbulent core



Typical velocity profile for a turbulent boundary layer – "Law of the Wall"

Turbulence models should capture the near wall behavior for accurate wall modeling

Turbulence Modeling

- Two approaches for capturing near-wall behavior

Wall function approach

- Uses logarithmic law-of-the-wall at the walls as boundary conditions
  - Near-wall flow is not solved
  - Alleviates the need for very fine near-wall mesh resolution
  - Computationally efficient
- Some turbulence models *require* wall functions
  - k-ε RNG model is not valid in viscous sub-layer
- Other turbulence models do not require wall function (however, wall functions can be used when near-wall mesh resolution is coarse)
  - Spalart-Allmaras, k-ω SST

Low Reynold's number approach

- Requires fine near-wall mesh to resolve the flow
- Computationally expensive when accurately resolving near-wall flow
- Turbulence models appropriately capture near-wall behavior. For example:
  - Spalart-Allmaras and k-ω SST models ensure correct near-wall behavior when integrated down to wall while k-ε RNG model does not

References

References

- **Fluid Mechanics**
  - Introduction to Fluid Mechanics
    - Robert W. Fox & Alan T. McDonald
- **Turbulence Modeling**
  - Turbulence Modeling for CFD
    - David C. Wilcox
  - Turbulent Flows
    - Stephen B. Pope
- **Continuum Mechanics**
  - Introduction to the Mechanics of a Continuous Medium
    - Lawrence E. Malvern

Notes

Notes

Introduction

Lecture 2

Overview

- SIMULIA
- Multiphysics
- Abaqus/CFD
- Fluid-Structure Interaction (FSI)
- Native FSI using Abaqus
- Target Applications
- System and Licensing Requirements
- Execution Procedure

SIMULIA

SIMULIA

SIMULIA is the Dassault Systèmes brand that delivers a scalable portfolio of Realistic Simulation solutions

- The Abaqus product suite for **Unified FEA**
- **Multiphysics** solutions for insight into challenging engineering problems
- **Lifecycle management** solutions for managing simulation data, processes, and intellectual property

Multiphysics

SIMULIA

Multiphysics

- **Multiphysics involves the inclusion of multiple physical representations to capture the diversity of behavior present in real-world problems**
  - Multiphysics solutions offered by SIMULIA broadly falls into three different areas

Abaqus Multiphysics

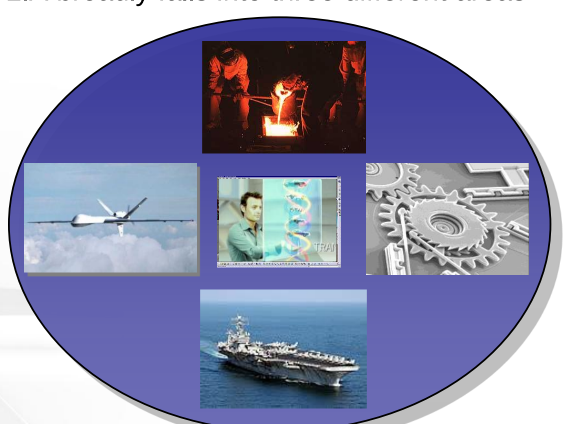
- Native multiphysics capabilities available in Abaqus
- Broad range of physics

Extended Multiphysics

- Extended multiphysics capability
- CEL in Abaqus/Explicit
- **Abaqus/CFD**

Multiphysics Coupling

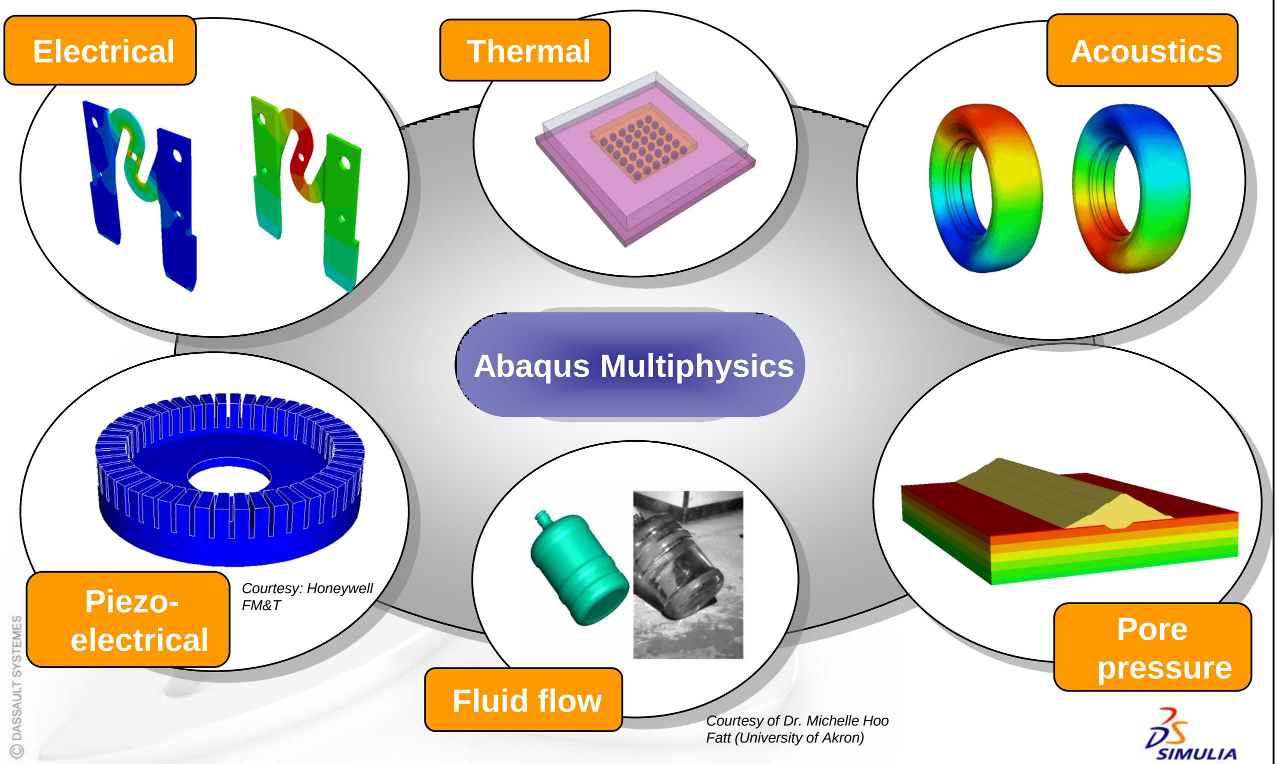
- Open scalable platform for partners and customers
- **Co-simulation engine**
- **Native FSI capability**
- **Coupling with third-party CFD codes**



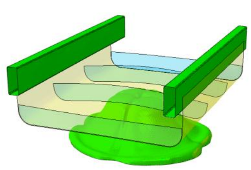
- Multiphysics simulation requirements spans *every industry* and *nearly every real application*
- Multiphysics simulation is often *required* for *Realistic Simulation*

SIMULIA

Abaqus Multiphysics



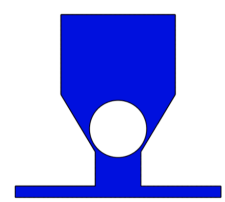
Extended Multiphysics



Blast loaded structure

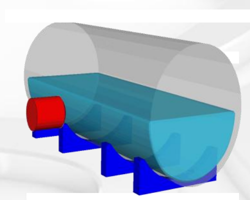
Tire Hydroplaning

Aortic Aneurysm



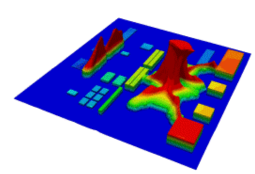
Automotive Brake Valve

Airbag inflation



Tank sloshing

Flow in Arteries



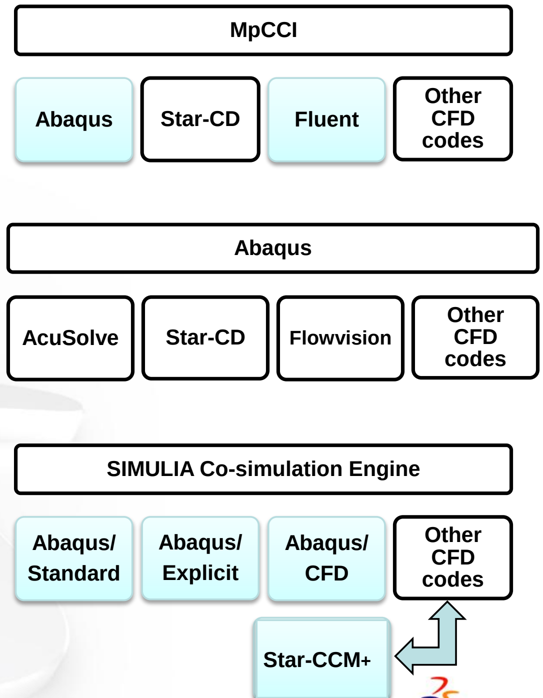
Electronic Cooling

Coupled Eulerian-Lagrangian (CEL) technology

Abaqus/CFD

Multiphysics Coupling

Independent code coupling interface	• Enabled through MpCCI from Fraunhofer SCAI • Allows coupling Abaqus with all codes supported by MpCCI
SIMULIA Direct Coupling	• Enables Abaqus to couple directly to 3rd party codes • Currently in maintenance mode
Co-simulation Engine (CSE)	• SIMULIA's next generation open communications platform that seamlessly couples computational physics processes in a multiphysics simulation • Physics-based conservative mapping technology • Superior coupling technology



SIMULIA

Multiphysics Coupling

- **FSI using Abaqus in 6.10:**

  Abaqus/Standard, Abaqus/Explicit 6.10 ←CSE→ Abaqus/CFD 6.10

  - *This is covered in the current training class*

- **FSI using Abaqus and third-party CFD codes**
  - Product support with 6.10:

    Abaqus 6.10 ↔ MpCCI 4 ↔ Fluent 12

    Abaqus 6.10 ←CSE→ Star-CCM+ 5

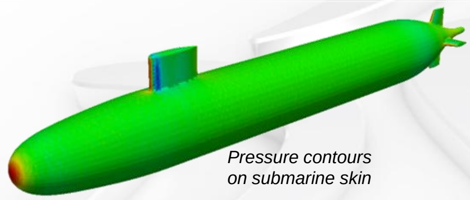
  - SIMULIA Direct Coupling Interface (DCI)
    - DCI is in maintenance mode
    - Coupling with Flowvision and AcuSolve is supported by respective third parties
  - More information is available in training class "FSI Simulation Using Abaqus and Third-party CFD Codes"

SIMULIA

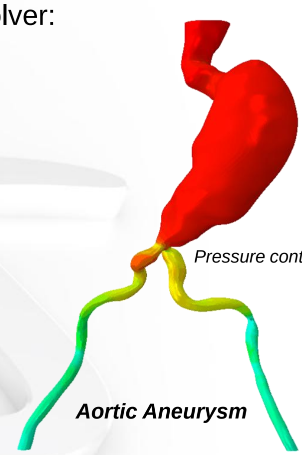
Abaqus/CFD

Abaqus/CFD

- **Abaqus/CFD is the computational fluid dynamics (CFD) analysis capability offered in the Abaqus product suite to perform fluids analysis**
  - *Scalable* CFD solution in an *integrated* FEA-CFD multiphysics framework
  - Based on hybrid finite-volume and finite-element method
  - Incompressible, pressure-based flow solver:
    - Laminar & turbulent flows



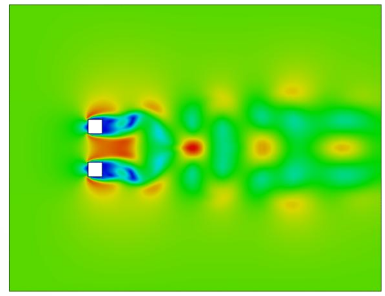
Submarine



Aortic Aneurysm

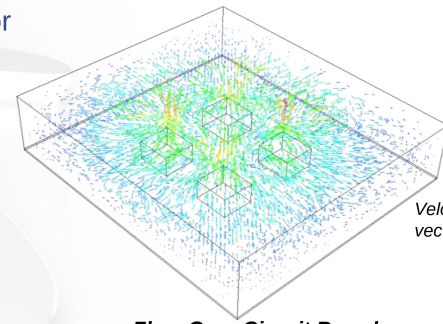
Abaqus/CFD

- **Incompressible, pressure-based flow solver:**
  - Transient (time-accurate) method
    - 2nd-order accurate projection method
    - Steady-state using time marching and backward Euler method
    - 2nd –order accurate least squares gradient estimation
    - Unsteady RANS approach (URANS) for turbulent flows
  - Energy equation for thermal analysis
  - Buoyancy driven flows (natural convection)
    - Using Boussinesq approximation



Velocity contours

Flow Around Obstacles (Vortex Shedding)



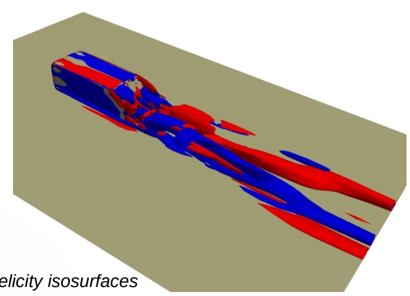
Velocity vectors

Flow Over Circuit Board

Buoyancy driven flow due to heated chips

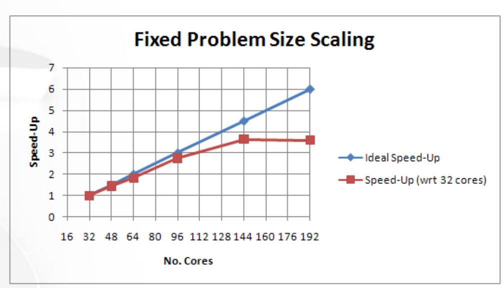
Abaqus/CFD

- **Turbulence models**
  - Spalart-Allmaras
    - Both steady-state and transient, i.e., URANS
  - ILES (Implicit Large-Eddy Simulation)
    - Transient by nature
- **Iterative solvers for momentum, pressure and transport equations**
  - Krylov solvers for transport equations
    - Momentum, turbulence, energy, etc.
  - Algebraic Multigrid (AMG) preconditioned Krylov solvers for pressure-Poisson equations
- **Fully scalable and parallel**

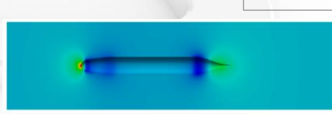


Helicity isosurfaces

Prototype Car Body (Ahmed's body)

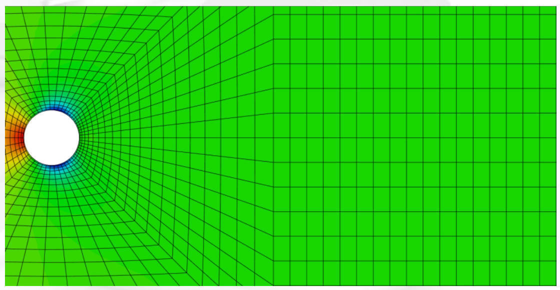


88 % efficiency (fixed work per processor at 64 cores)



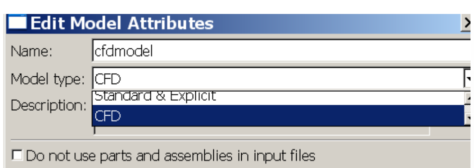
Abaqus/CFD

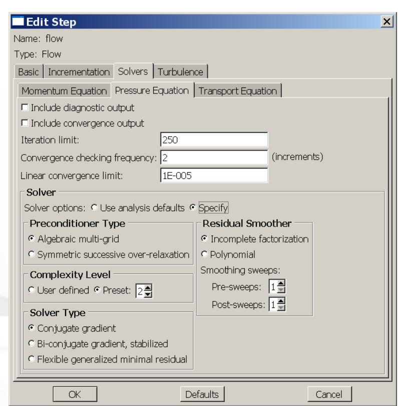
- **Other features**
  - Fluid material properties
    - Newtonian fluids only
  - CFD-specific diagnostics and output quantities
  - Arbitrary Lagrangian-Eulerian (ALE) capability for moving deforming mesh problems
    - Prescribed boundary motion, Fluid-structure interaction
    - “hyper-foam” model, total Lagrangian formulation



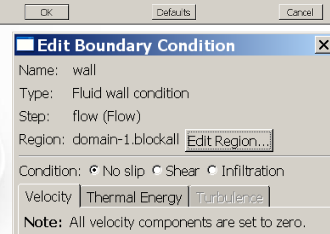
Abaqus/CFD

- **Abaqus/CAE support**
  - Concept of “model type” in Abaqus/CAE
    - Model type “CFD” enables CFD model creation





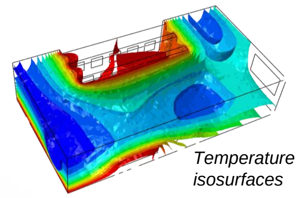
  - Support for CFD-specific attributes
    - Step definition
    - Initial conditions
    - Boundary conditions and loads

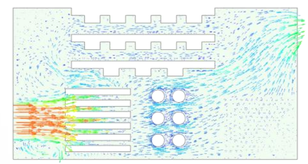


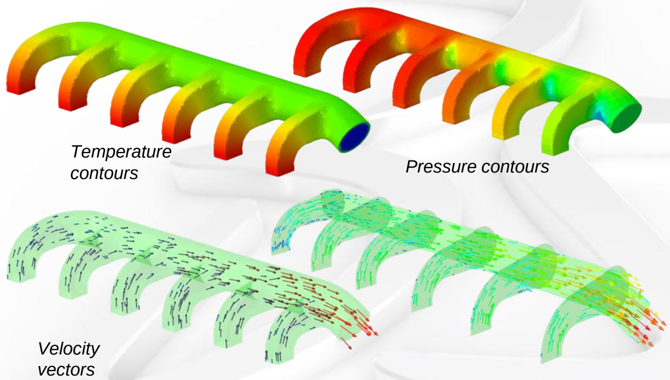
  - Job submission, monitoring etc.

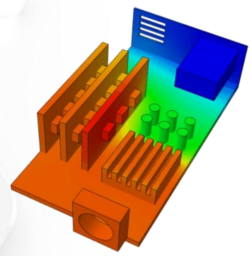
Abaqus/CFD

- **Abaqus/Viewer support for Abaqus/CFD**
  - CFD output database
  - Isosurfaces
  - Multiple cut-planes
  - Vector plots
    - Multiple cut-planes





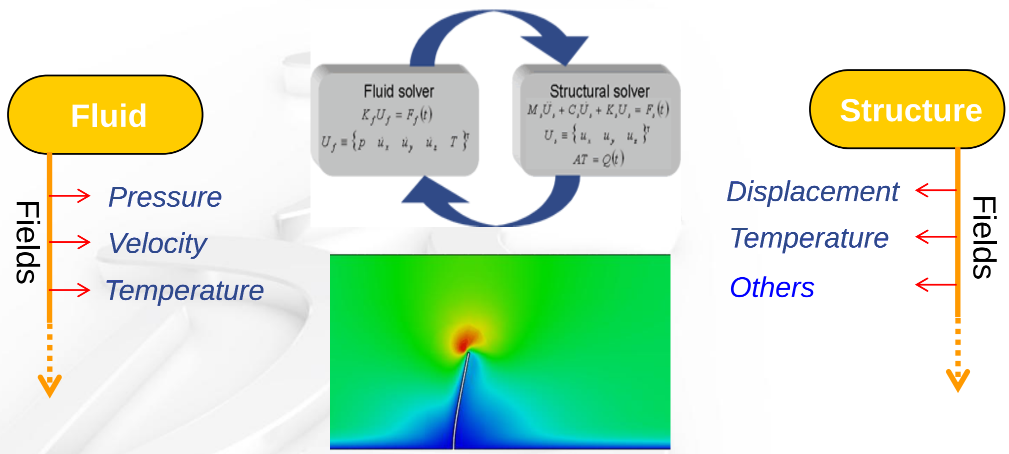




Fluid-Structure Interaction

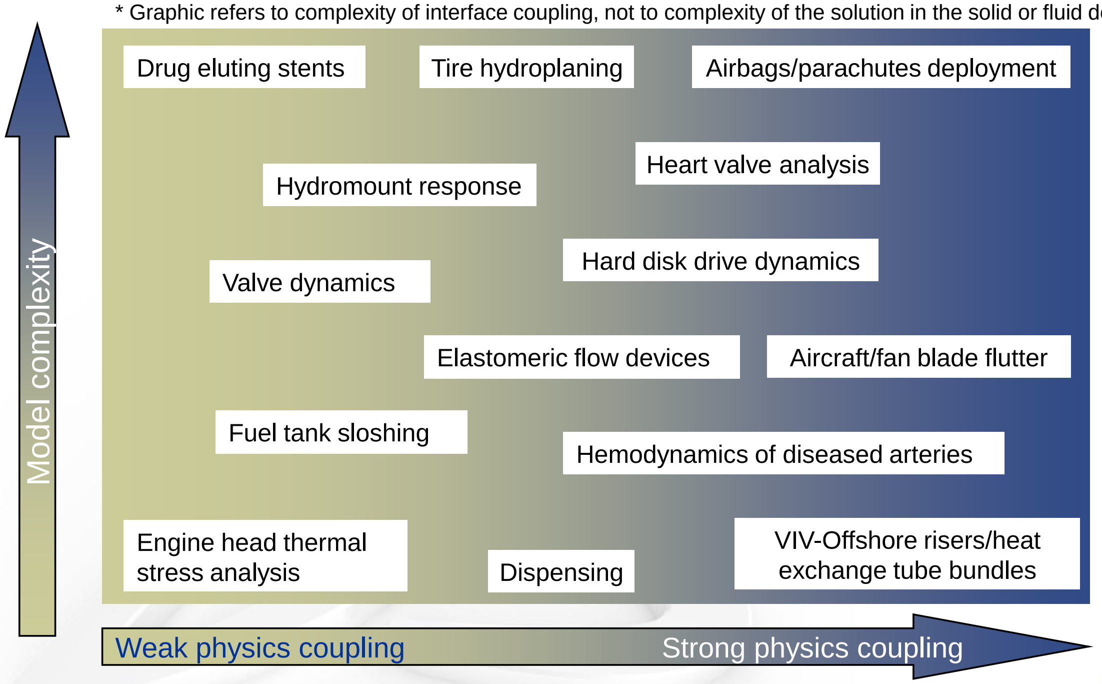
What is Fluid-Structure Interaction or FSI?

- **FSI represents a class of multiphysics problems where fluid flow affects compliant structures, which in turn affect the fluid flow**
  - Partitioned solution approach is widely used
  - One of more fields may be of interest



FSI Coupling Spectrum

* Graphic refers to complexity of interface coupling, not to complexity of the solution in the solid or fluid domain.



Current FSI Technology Spectrum

Increasing solution complexity

6-DOF solver	Simple FSI	Staggered Approach (Explicit/Implicit)	Specialized techniques	Monolithic approach
Structure represented in the fluids code as a 6 DOF entity	Compliance matrix/eigen value approach to solving the structural problem inside a fluids code	Structure and fluid equations solved separately with code coupling and mapping at the interface	1.SPH: Meshless method 2.Immersed Boundary Techniques 3.CEL	Single set of equations for the fluid and structural domains
Suitable for rigid body motions in a fluid.	Suitable for linear structural problems	Suited for weak to moderately strong coupling physics problems. Implicit coupling well suited for tackling unstable FSI problems	Suitable for problems where structural modeling is too complex or deformations are significant	Suited for all coupling physics problems
Examples: IC engines, rigid valve movement	Examples: Sloshing, vortex-induced vibrations	Examples: Pulsatile blood flow, dispensing	Examples: Tire hydroplaning	Examples: All

Current FSI Technology Spectrum

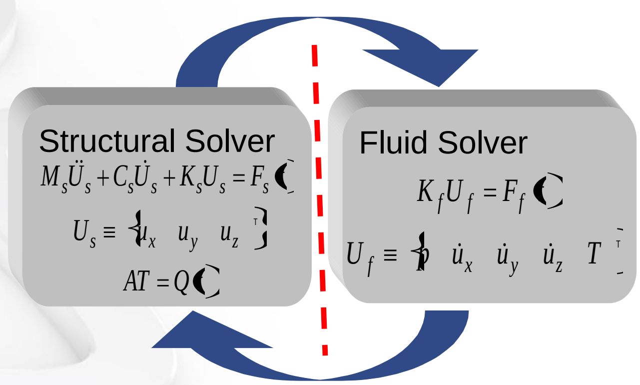
- **Linear Structures Approach**
  - Assumption 1: Linear solid/structural deformation
  - Assumption 2: Eigenmodes sufficient to represent the dynamic behavior
  - Projection of dynamic system onto the eigenspace

$$\mathbf{Ma} + \mathbf{Cv} + \mathbf{Kd} = \mathbf{F}$$

$$(\mathbf{K} - \lambda_i \mathbf{M})\mathbf{S}_i = 0 \qquad i = 1, \ldots, n_{\text{modes}}$$

$$\mathbf{m}\ddot{\mathbf{y}} + \mathbf{c}\dot{\mathbf{y}} + \mathbf{k}\mathbf{y} = \mathbf{f}$$

- **Partitioned Approach**
  - Structural and fluid equations solved independently
  - Interface loads and boundary conditions exchanged after a converged increment



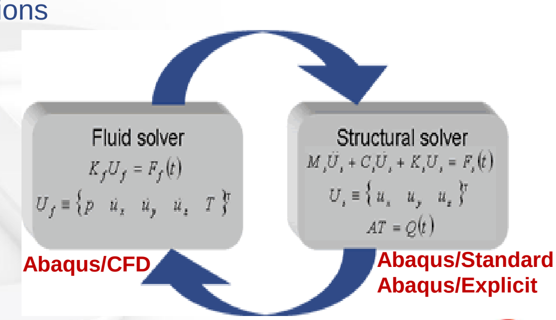
- **Specialized Techniques**
  - Coupled Eulerian-Lagrangian

Abaqus native FSI capability is based on a partitioned approach

Native FSI using Abaqus

Native FSI using Abaqus

- Abaqus/CFD couples with Abaqus/Standard and Abaqus/Explicit through the co-simulation engine
  - The co-simulation engine operates in the background (no user intervention required)
    - Physics-based conservative mapping on the FSI interface
- Significantly expands the set of FSI applications that SIMULIA can address independently
  - Fluid-structure interaction
  - Conjugate heat-transfer applications
- Based on partitioned approach
  - Explicit coupling between codes
  - Conditionally stable

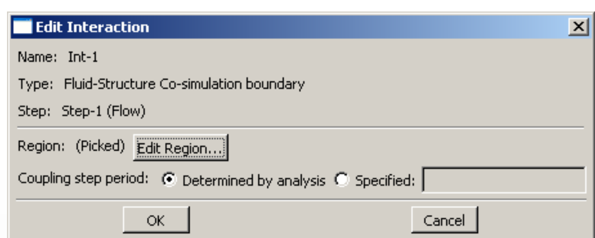


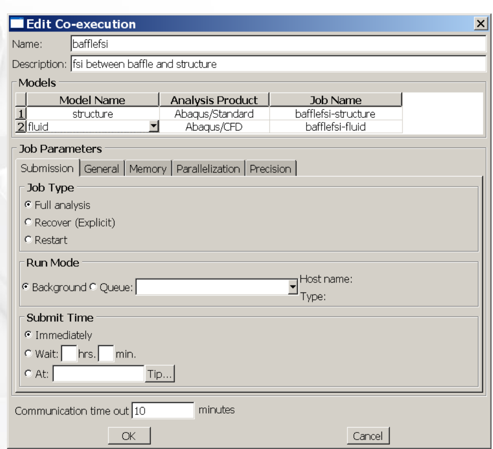
Native FSI using Abaqus

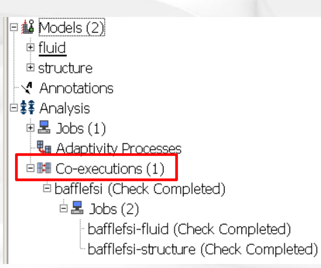
Coupling	Abaqus/Standard + Abaqus/CFD	Abaqus/Explicit + Abaqus/CFD
Fluid structure interaction	✓	✓
Conjugate heat-transfer	✓	⊘

Native FSI using Abaqus

- Supported though Abaqus/CAE
  - Support for creating “FSI” interactions in
    - Structural analysis (in Abaqus/Standard or Abaqus/Explicit)
    - CFD analysis (in Abaqus/CFD)
  - FSI jobs launched through co-execution framework







Native FSI using Abaqus

- The native FSI capability in Abaqus addresses weak to moderately coupled FSI problems
  - For problems where "added-mass" effects are important, this approach may lead to numerical instabilities
    - Occurs when fluid density is close to the density of the structure
    - Examples:
      - In interactions with water, added-mass effect is important
      - In interactions with air, added-mass effect can often be ignored
  - Not a limitation of Abaqus but a common limitation of explicit FSI coupling based on a partitioned approach
  - Conjugate heat-transfer problems are also conditionally stable but the stability envelope is much larger
    - The stability limit is encountered in rare circumstances

Target Applications

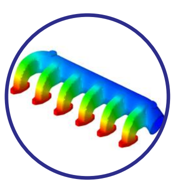
Target Applications

Application attributes

Time-domain response		**Distinct fluid and structure domains**	**Geometric consistency of models**		
FSI is transient	Steady-state response for quasi-static problems	Interaction between the domains is through user-identified surfaces	Co-located models	Consistent geometric idealizations of fluid and structure	Consistent units

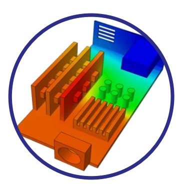
SIMULIA

Target Applications

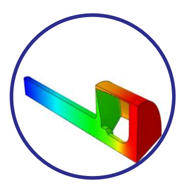


Automotive
- Engine, exhaust manifold, cooling jackets
- Hydraulic engine mounts
- Disc brake system
- ABS, shock absorbers

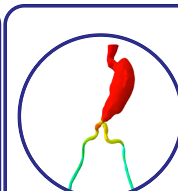
Power/gas
- Pipelines, risers
- Heat exchangers (nuclear power)



Electronics
- Cooling of electronic components
- Manufacturing of integrated circuits



Industrials
- Flow limiters, seals



Medical
- Heart valves
- Blood flow

SIMULIA

Target Applications

- **Applications not targeted**
  - Vibroacoustics
    - More effectively treated by frequency domain methods
  - Structures modeled with rod, beam, truss, cable elements
    - Inconsistent geometry idealizations
  - Injection molding, casting, superplastic forming
    - Indistinct or changing fluid/structure interface
  - Rupture, penetration, fragmentation
    - Variable fluid region topology

System and Licensing Requirements

System and Licensing Requirements

- **Abaqus/CFD requires MPI installation**
  - Even for a single-cpu run
  - MPI configuration:
    - Windows / x86-32 - HPMPI
    - Windows / x86-64 - MSMPI
    - Linux / x86-64 - HPMPI
- **Platforms supported:**
  - Supported on all 6.10-supported platforms
    - Windows / x86-32
    - Windows / x86-64
    - Linux / x86-64
- **More information:**
  - http://www.simulia.com/support/sup_systems_info.html

System and Licensing Requirements

- **Running Abaqus/CFD requires a “CFD” license feature in the license file**
- **Additionally, running FSI using Abaqus/CFD requires a co-simulation engine license feature in the license file**

Execution Procedure

Execution Procedure

- **From within Abaqus/CAE**
  - Abaqus/CFD jobs can be run from within Abaqus/CAE as regular Abaqus jobs
  - FSI jobs can be launched from within Abaqus/CAE as a co-execution job
- **From the command line**
  - Abaqus/CFD jobs

    ```
    abaqus -job <job name> -cpus <# of cpus>
    ```
  - FSI jobs

    ```
    abaqus -job <job 1 name> -listenerPort 11111 -remoteConnections <hostname>:22222
    ```

    ```
    abaqus -job <job 2 name> -listenerPort 22222 -remoteConnections <hostname>:11111
    ```
    - So if you were running Job-1 on a machine named **blue** and Job-2 on machine named **red**, the commands would be:

      ```
      abaqus -job Job-1.inp -listenerPort 11111 -remoteConnections red:22222
      ```

      ```
      abaqus -job Job-2.inp -listenerPort 22222 -remoteConnections blue:11111
      ```

Notes

Notes

Getting Started with Abaqus/CFD

Lecture 3

Overview

- CFD Simulation Workflow
- Setting up CFD Analyses
- Case Study 1: Flow around a Rigid Circular Cylinder
- Case Study 2: Flow around an Oscillating Rigid Circular Cylinder
- Modeling Heat Transfer
- Modeling Turbulence

CFD Simulation Workflow

CFD Simulation Workflow

CFD Simulation Workflow

- **CAD Geometry**
  - Abstraction and clean up
  - Volume creation to represent flow volume
- **CFD Meshing**
  - Surface meshing followed by volume mesh
  - Volume meshing of flow volume
  - Mesh topology
  - Tetrahedral, Hexahedral, Wedges, Pyramids, Mixed boundary layer mesh
- **CFD Solver**
  - Steady or transient
  - Laminar or turbulent
  - Numerical scheme
  - Initial & boundary conditions
- **Postprocess**
  - Contours, vector plots, streamlines

CFD Simulation Workflow

- **CFD Simulation Workflow in Abaqus/CAE**

Part module:
Create a part representing the flow domain

Property module:
Define fluid properties; create and assign fluid section

Assembly module:
Instance and position the parts

Step module:
Define fluid analysis step, solver controls, turbulence models, etc.

Interaction module:
Define interactions for FSI problems

Load module:
Apply boundary conditions, body forces, fluid reference pressure etc.

Mesh module:
Create CFD mesh

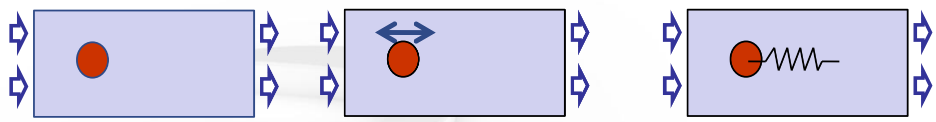
Job module:
Create and submit CFD job

Visualization module:
Postprocess results

Setting up CFD Analyses

Setting up CFD Analyses

- **Case study introduction**
  1. Flow around a rigid circular cylinder
  2. Flow around an oscillating rigid circular cylinder
  3. Flow around a spring-loaded rigid circular cylinder (***Covered in Lecture 6***)



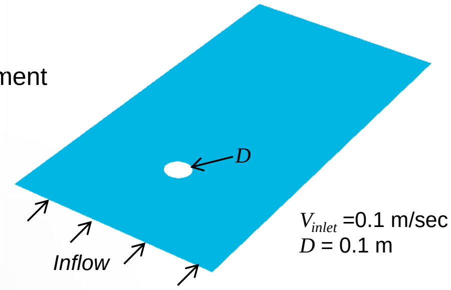
Problem description	Flow around a rigid circular cylinder	Flow around an oscillating rigid circular cylinder	Flow around a spring-loaded rigid circular cylinder
Flow domain	Around the cylinder	Around the cylinder but domain *changes* due to cylinder's oscillation	Around the cylinder but domain *changes* due to cylinder's oscillation
How do I model it?	1) Model fluid flow 2) Mesh is fixed	1) Model fluid flow 2) Allow mesh at cylinder surface to accommodate displacements (ALE)	1) Model fluid flow 2) Allow mesh at cylinder surface to accommodate displacements (ALE) 3) Model the cylinder and the spring in structural solver (co-simulation)
Cylinder motion	None	Modeled in Abaqus/CFD as a boundary condition	Determined by structural analysis (two separate models)

Case Study 1: Flow around a Rigid Circular Cylinder

Introduction

- Flow around a rigid circular cylinder is often used as a CFD benchmark case
  - Characteristic length scale: D (cylinder diameter)
  - Model the flow as 3-dimensional but with one element through the thickness and symmetry boundary conditions on the front and back faces to enforce 2-dimensional conditions

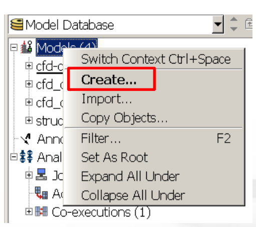
$$\mathrm{Re} = \frac{\rho V D}{\mu}$$

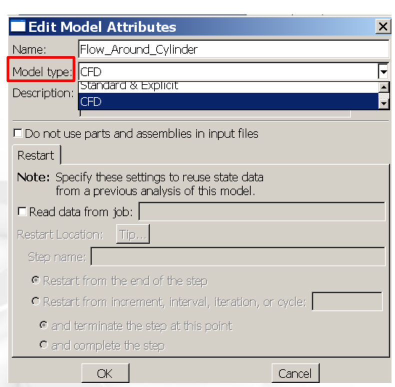


Reveals interesting flow features depending on the Reynolds number

Defining the CFD Model

1. Create a "CFD" model in Abaqus/CAE



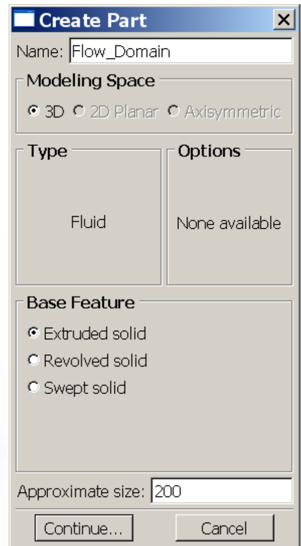


- CFD-specific model attributes are only accessible for Abaqus/CAE models of type "CFD"
- The model type, once set, cannot be changed
  - Parts, instances, material properties etc. can be copied between "Standard & Explicit" and "CFD" model types

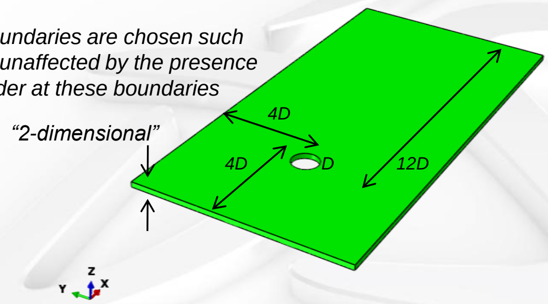
Defining the CFD Model

2. Define a part representing the flow domain

- Only 3-dimensional parts can be modeled.
  - Use a 3D sector for axisymmetric models
  - Use a 3D part with one element through the thickness for 2-dimensional models
- Orphan CFD meshes can be imported
  - Tetrahedral and hexahedral elements only
- All Abaqus/CAE features for geometry creation are accessible



Far field boundaries are chosen such that flow is unaffected by the presence of the cylinder at these boundaries



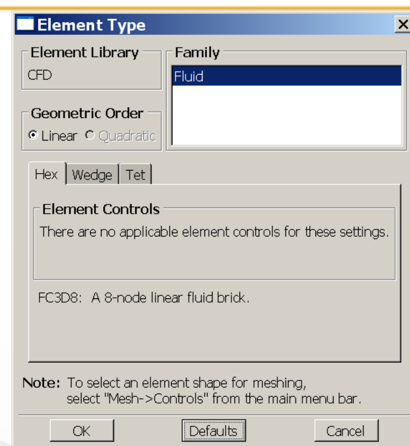
- For detailed information about geometry creation in Abaqus/CAE, please refer to
  - ***Abaqus/CAE User's Manual***
  - ***Introduction to Abaqus/CAE*** lecture notes

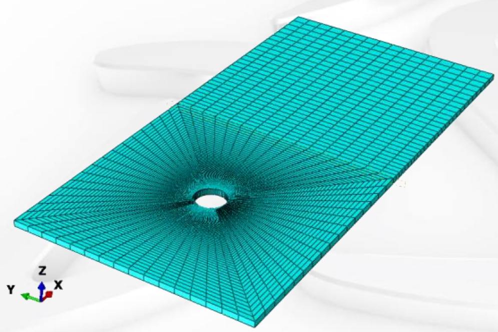
SIMULIA

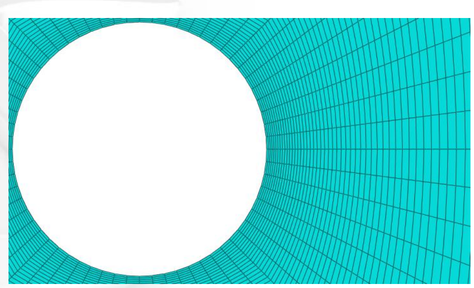
Defining the CFD Model

3. Generate the mesh

- Hexahedral (FC3D8) and tetrahedral (FC3D4) element types are available
  - No mixed meshes are allowed
- Proper modeling of the boundary layer requires a fine mesh near the cylinder surface to resolve the velocity gradient
  - Fluid velocity is zero at the cylinder surface (no-slip, no-penetration condition)
  - Typically, the CFD mesh is refined at no-slip walls and coarsens as we move away from walls





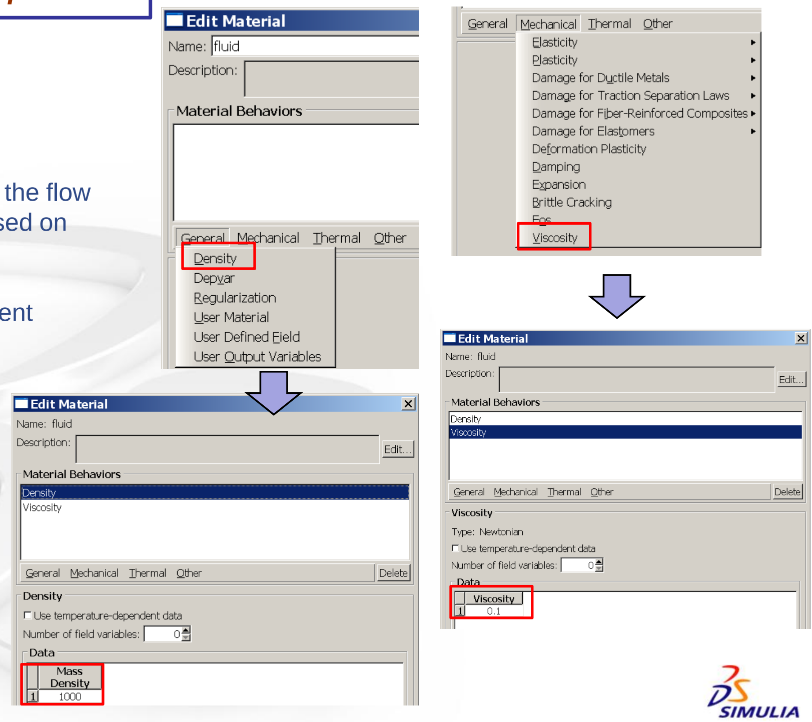


SIMULIA

Defining the CFD Model

4. Define fluid material properties

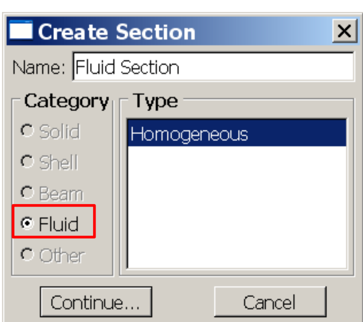
- Newtonian fluid
  - Density: 1000 kg/m^3
  - Viscosity: 0.1 Pa.sec
- Properties are chosen to set the flow Reynolds number at 100 based on cylinder diameter
- Temperature or field dependent quantities are not supported

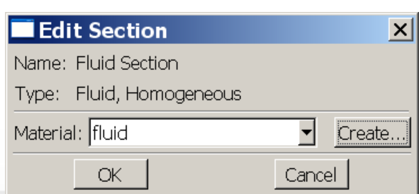


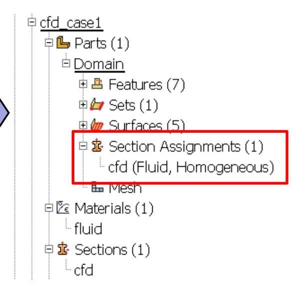
SIMULIA

Defining the CFD Model

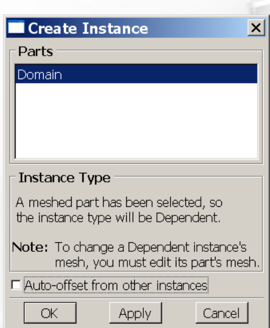
5. Create and assign a fluid section and instance the part







- Section category “Fluid” is the only choice available when the CFD model type is chosen
- Elements without a section assignment will cause a processing error



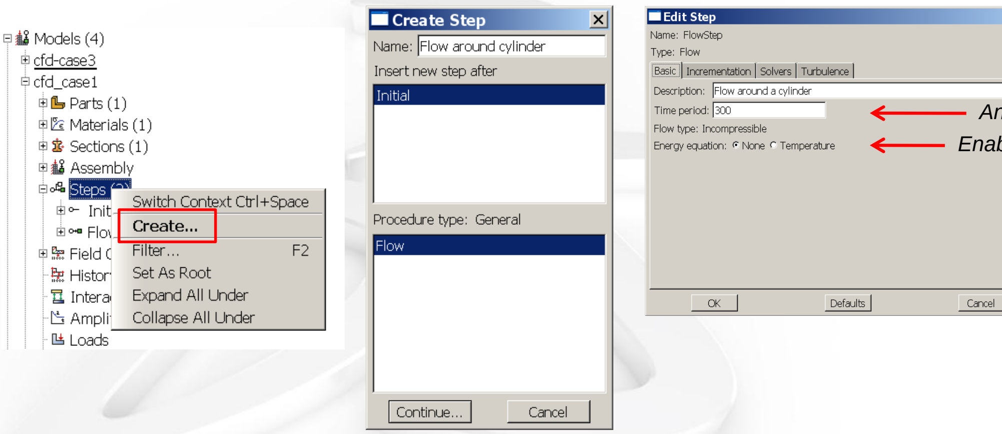
- Instance the part (or orphan mesh) and position it in the desired geometric location

SIMULIA

Defining the CFD Model

6. Define an incompressible flow analysis

- **Transient analysis attributes**
  - Time period
  - Energy equation (off by default)
  - Time incrementation (automatic or fixed)
  - Solver controls
  - Laminar or turbulent flows



Analysis time period

Enable heat transfer

Defining the CFD Model

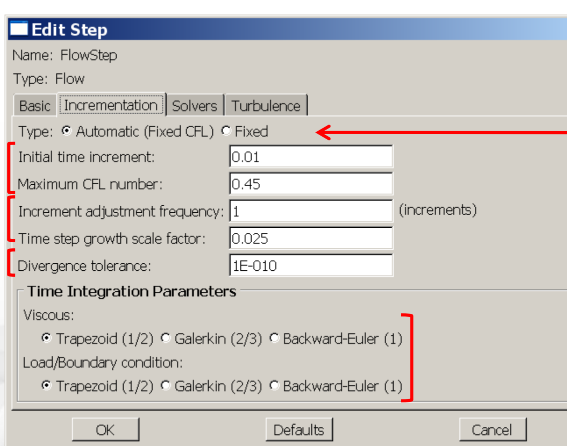
6. Define an incompressible flow analysis (cont'd)

Time incrementation

Automatic time integration parameters
• Initial time increment
• CFL limit

Time increment growth

Controls pressure update



Automatic and fixed time incrementation

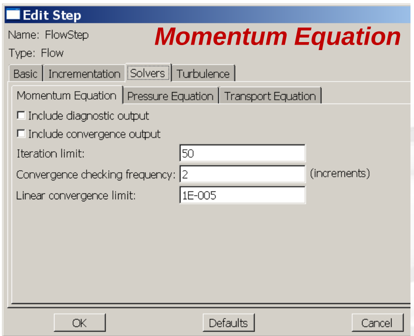
Time integration accuracy
• 2nd order accuracy for trapezoid scheme

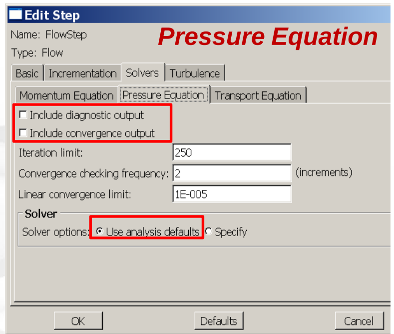
- CFL limit: Courant-Fredrichs-Levy (CFL) condition
  - Arises due to the explicit treatment of advective terms in the Navier-Stokes equations

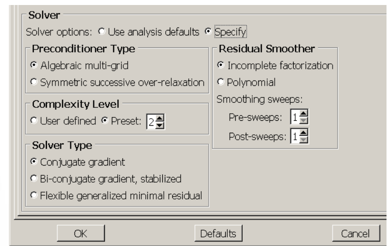
Defining the CFD Model

6. Define an incompressible flow analysis (cont'd)

Solver controls





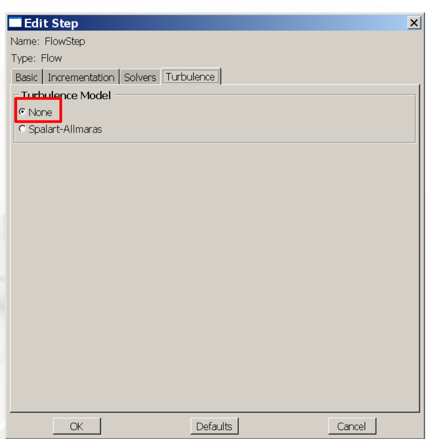


- Primary solution quantities
  - Velocity components
  - Pressure
- Convergence and diagnostic output for each of the equations is not written by default, but they can be toggled on
- Many solver choices are available for the Pressure Poisson's Equation
  - Use preset levels (default 2)

Defining the CFD Model

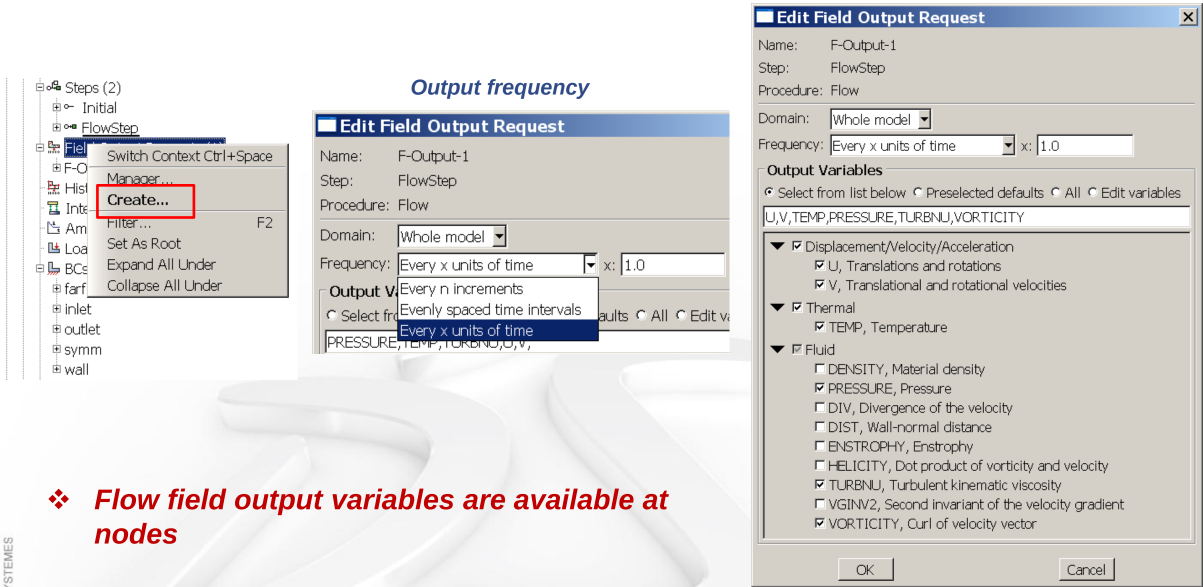
6. Define an incompressible flow analysis (cont'd)

- **Laminar or turbulent flow**
  - Laminar flow (default)
  - Choice of turbulence models



Defining the CFD Model

7. Request output variables

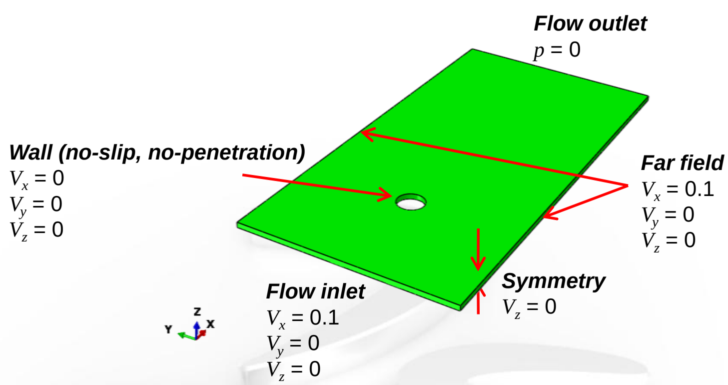


❖ ***Flow field output variables are available at nodes***

SIMULIA

Defining the CFD Model

8. Define boundary conditions



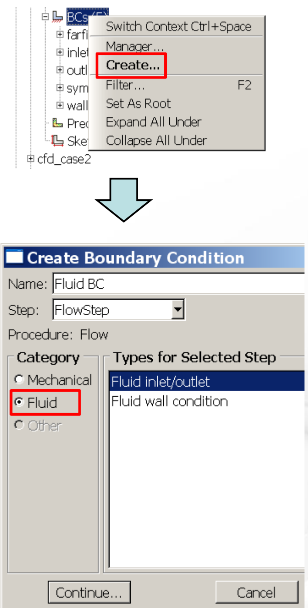
- One element in the thickness direction
  - Symmetry boundary conditions on faces to enforce 2-dimensional conditions
- Reynolds number = 100

❖ ***Fluid boundary conditions are applied on surfaces***

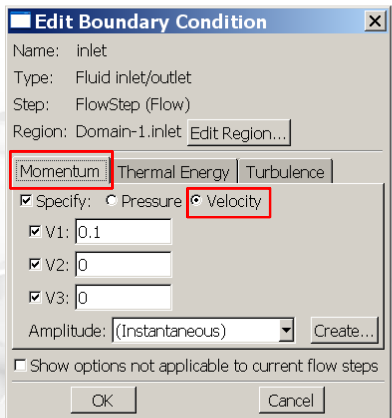
SIMULIA

Defining the CFD Model

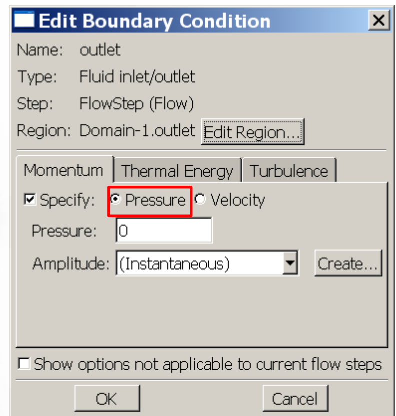
8. Define boundary conditions (cont'd)



Flow inlet



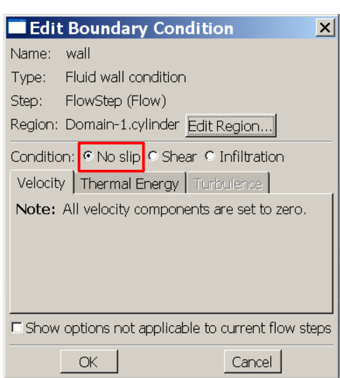
Flow outlet



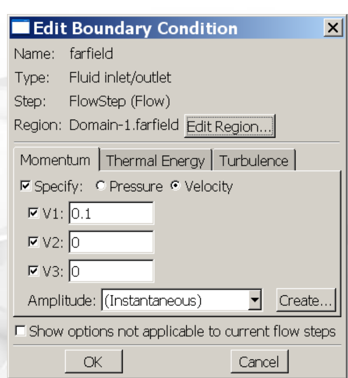
Defining the CFD Model

8. Define boundary conditions (cont'd)

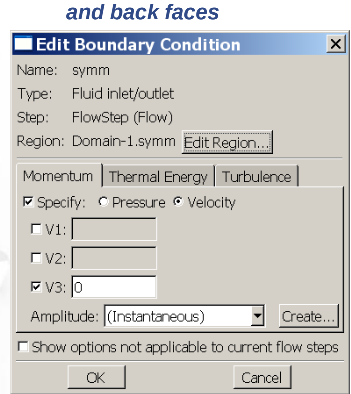
Wall condition



Far field velocities

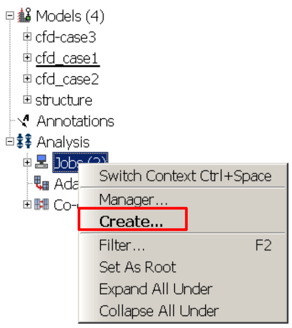


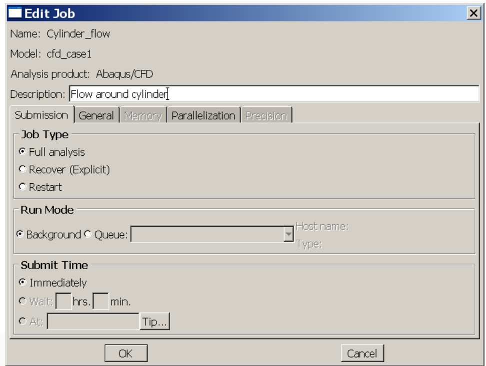
Symmetry at front and back faces

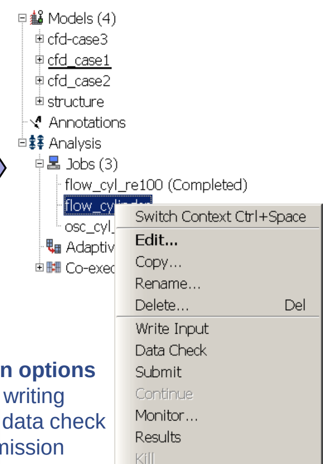


CFD Analysis Execution

CFD analysis job







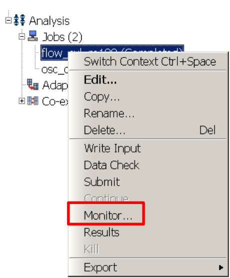
Job execution options

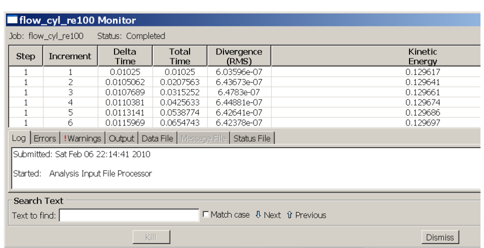
- Input file writing
- Analysis data check
- Job submission
- Job monitoring
- Accessing output database (ODB)
- Job termination

Input files are not supported

(You can write input files and use them for job submission; however, you must use the input file generated by Abaqus/CAE)

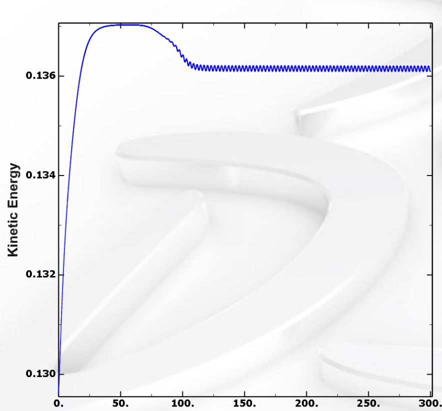
Monitoring the Solution

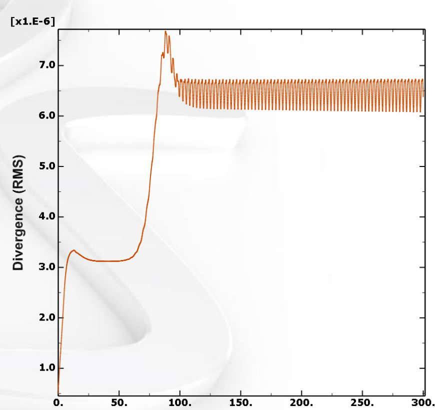




Divergence (RMS)

- ✓ Measure of mass balance
- ✓ Monitor divergence (RMS) to gauge solution convergence
- ✓ Divergence (RMS) < 1e–3
- ✓ High values often indicate incorrect problem set up
- ✓ Mesh dependent (refine mesh if divergence is too high)
- ✓ Volume error = Div*dt





Monitoring the Solution

Analysis output files

- Log (`.log`) file
- Status (`.sta`) file
- Output database (`.odb`) file

Status (`.sta`) file

```
CFD Incompressible Flow
Step  Inc  DT        Time        RMS Div.     KE
------ ------ ---------- -------------- -------------- -------------
   1    0 0.01000  0.000000    4.87625e-07  0.129588
   1    1 0.01025  0.0102500   6.03596e-07  0.129617
   1    2 0.01051  0.0207562   6.43673e-07  0.129641
   1    3 0.01077  0.0315252   6.47830e-07  0.129661
   1    4 0.01104  0.0425633   6.44881e-07  0.129674
   1    5 0.01131  0.0538774   6.42641e-07  0.129686
```

1. All mesh statistics, solver options, boundary conditions and timing summaries are written in the log file
2. Time increments and maximum and minimum velocity components (& temperatures) are output in log file
3. Solver convergence & diagnostics are not printed by default but they can be turned on for each individual equation

Log (`.log`) file (truncated view)

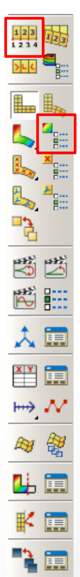
```
 T I M I N G  S U M M A R Y
=========================
open for read .......................................... 0.09375
readHeader ............................................. 0
registerData ........................................... 0
readConnectivity ....................................... 0.015625

 M E S H  S T A T I S T I C S
  ============================
  Problem dimension ...................................... 3
  Number of nodes ........................................ 11448
  Number of elements ..................................... 5632

 M A T E R I A L  M O D E L S
  ============================
Number of input materials .............................. 1
Material ID ............................................ 0
Mass Density ........................................... 1000
Specific Heat at constant pressure ..................... 0

 C O N T R O L  O P T I O N S
  ============================
Deformable mesh options

 D I R I C H L E T  B O U N D A R Y  C O N D I T I O N S
=======================================================
Number of Dirichlet BC's ............................... 11

 I N I T I A L  D I V E R G E N C E  S U M M A R Y
=================================================
Initial divergence ..................................... 0.00020949
Projected divergence ................................... 4.8762e-07
Net volumetric flux balance ............................ 8.0495e-11

 MEMORY USAGE
  ============
Total Memory Used [MBytes] ............................. 2.1932e+01

 Inc.   dt          Time       U-Min.    U-Max.    V-Min.     V-Max.    W-Min.     W-Max.
===== ========== ========== ========== ========== ========== ========== ====
   0  1.00e-02  0.00e+00  2.27e-03  2.00e-01  -9.82e-02  9.82e-02  -1.19e-04  5.98e-05
   1  1.02e-02  1.02e-02  1.34e-03  1.88e-01  -8.80e-02  8.80e-02  -1.08e-04  5.73e-05
  11  1.31e-02  1.28e-01 -1.35e-04  1.75e-01  -7.91e-02  7.91e-02  -2.39e-04  2.39e-04
```

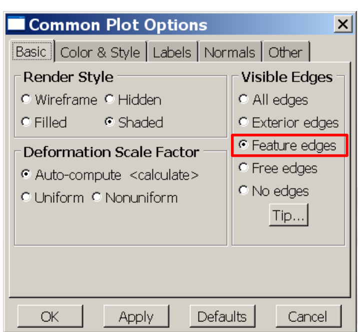

Postprocessing the Results

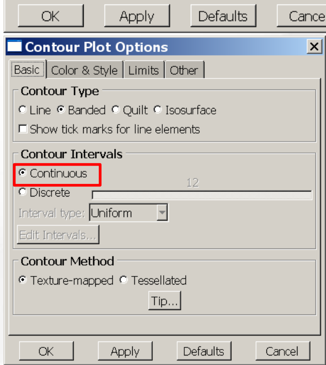
Abaqus/Viewer – Tips for postprocessing CFD output databases

1

2

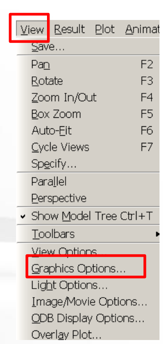






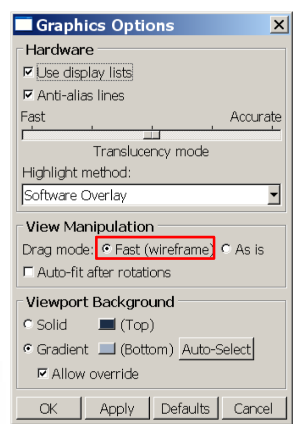
1

Turn off mesh display



2

Use continuous contour intervals



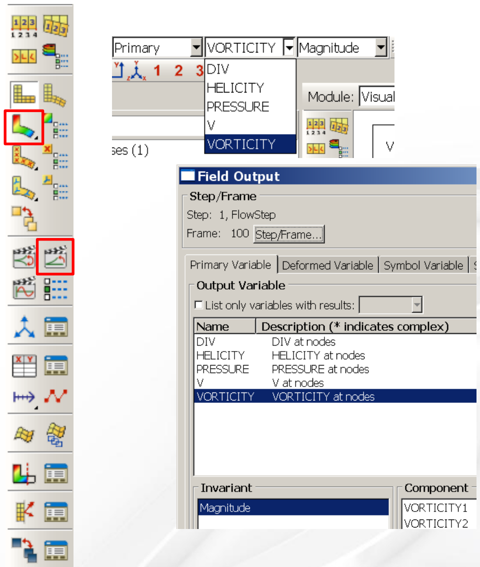
3

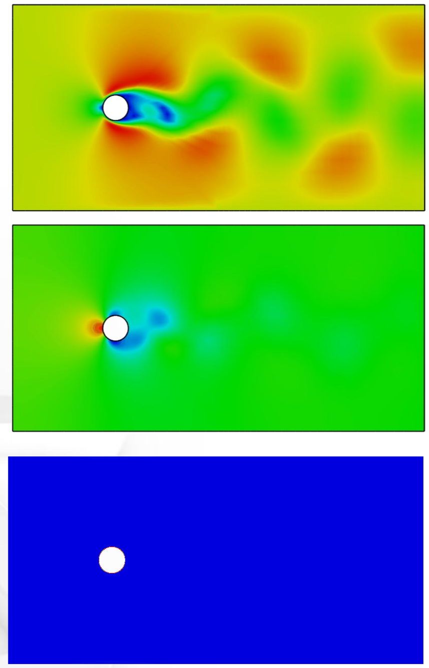
Fast view manipulation

(for large CFD models)

Postprocessing the Results

Contour plots & animations





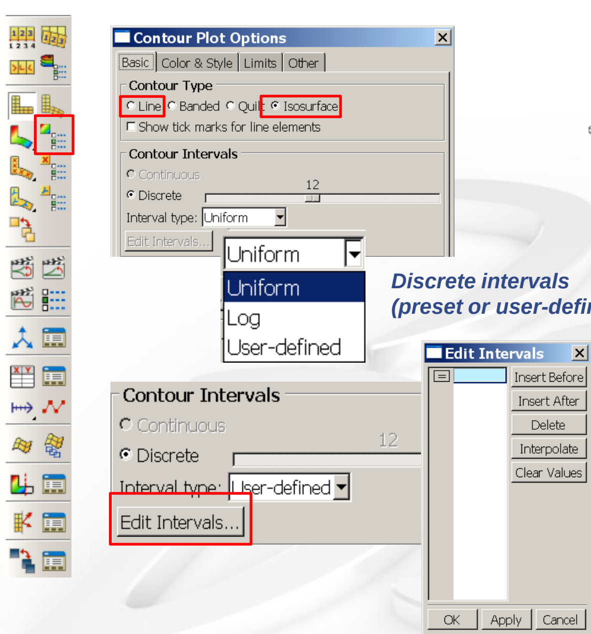
Velocity contour

Pressure contour

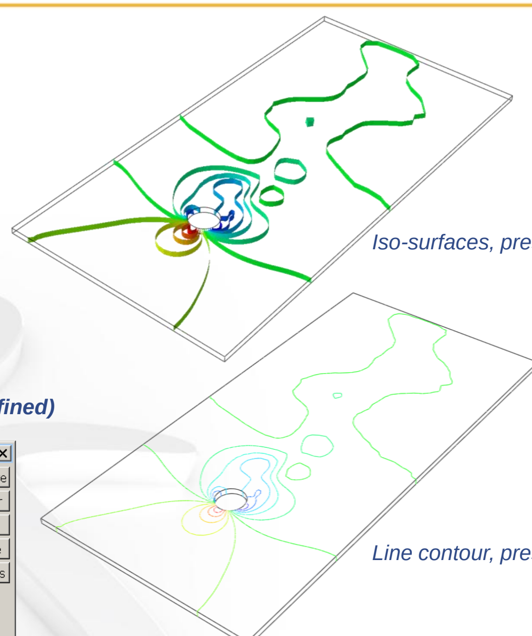
Vorticity animation

Postprocessing the Results

Iso-surfaces and line plots



Discrete intervals (preset or user-defined)



Iso-surfaces, pressure

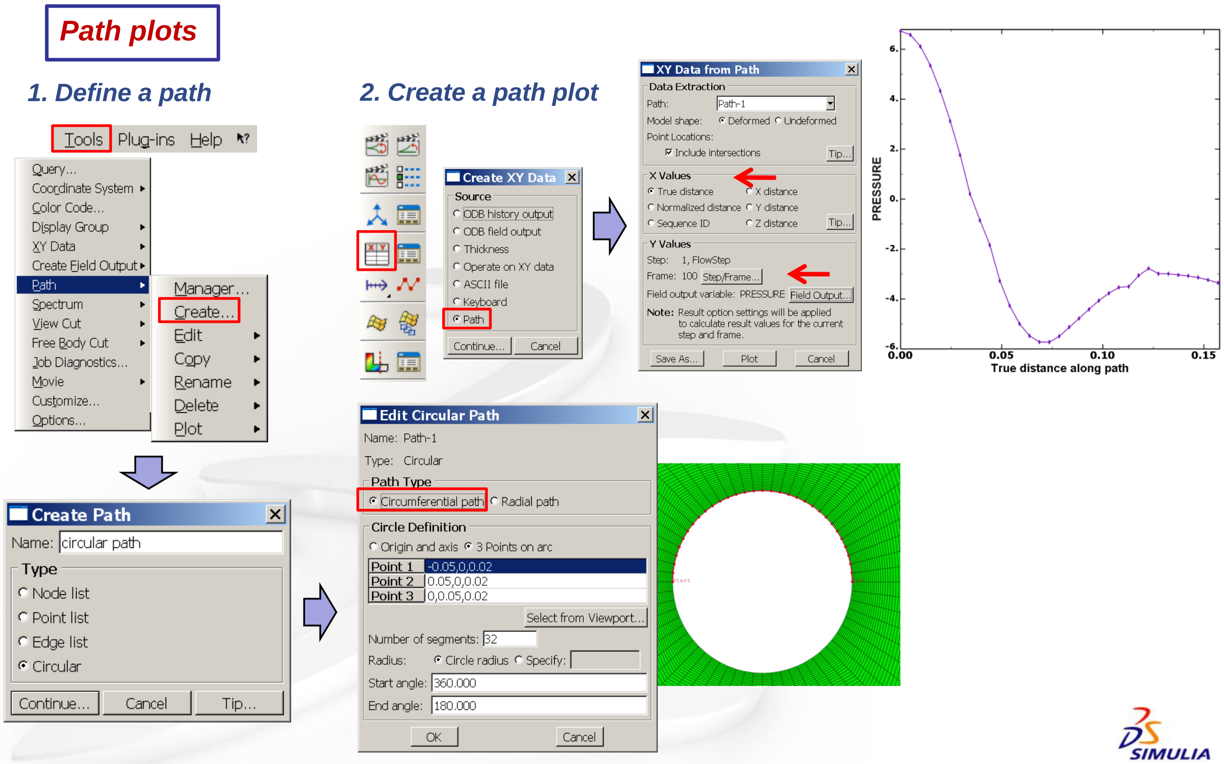
Line contour, pressure

Postprocessing the Results

Path plots

1. Define a path

2. Create a path plot



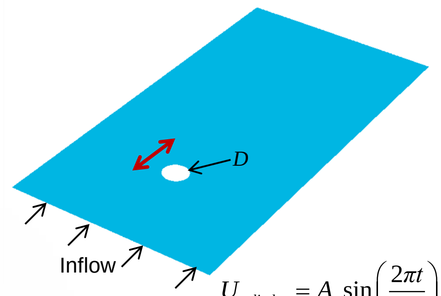
SIMULIA

Case Study 2: Flow around an Oscillating Rigid Circular Cylinder

SIMULIA

Introduction

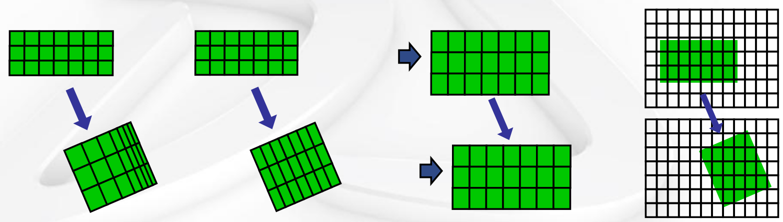
- Flow around an *oscillating* rigid circular cylinder
  - Flow domain is still around the cylinder but the domain *changes* due to cylinder's oscillation
  - Flow in modeled in Eulerian framework
    - Mesh is fixed while the material flows through it
  - Arbitrary Lagrangian-Eulerian (ALE) capability is required within CFD to model fluid flow when the boundary moves due to prescribed motion (boundary condition) or interaction (fluid-structure interaction)
    - "Deform" the mesh
    - *Mesh motion requires additional boundary conditions*



$$U_{cylinder} = A_o \sin\left(\frac{2\pi t}{T}\right)$$

$$V_{cylinder} = \frac{2\pi A_o}{T}\cos\left(\frac{2\pi t}{T}\right)$$

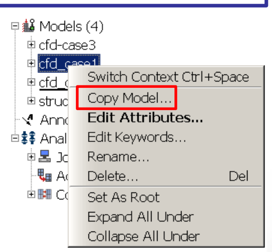
V_{inlet} =0.1 m/sec
D = 0.1 m
A_o = 0.05 m
T = 2 sec



SIMULIA

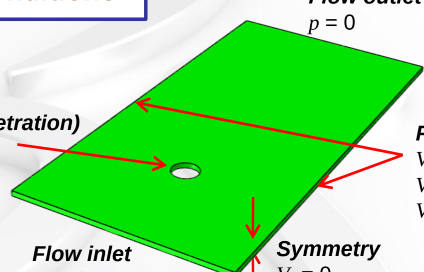
Defining the CFD Model

1. Copy CFD model



- Use the same mesh, fluid properties and analysis attributes
- Only require additional
  - Boundary conditions
  - Output requests

2. Define boundary conditions



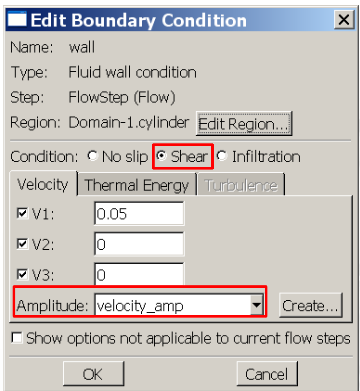
$$V_{cylinder} = \frac{2\pi A_o}{T}\cos\left(\frac{2\pi t}{T}\right)$$

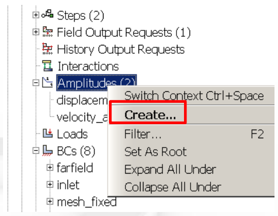
SIMULIA

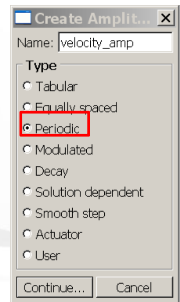
Defining the CFD Model

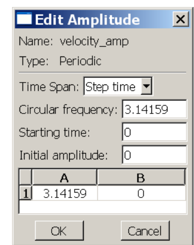
2. Define boundary conditions (cont'd)

Wall condition









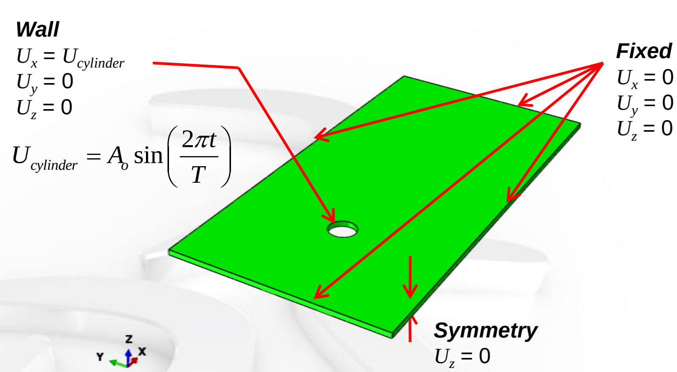
Defining periodic amplitude

❖ *Use wall boundary condition with shear condition to specify wall velocity*

Defining the CFD Model

2. Define boundary conditions (cont'd)

Boundary conditions for mesh motion

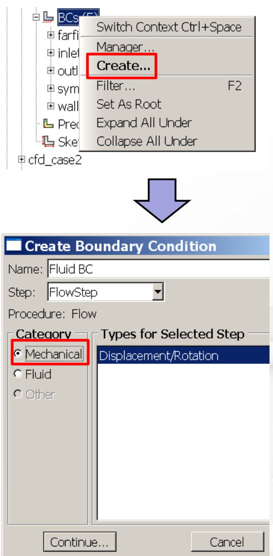


❖ *Mesh motion boundary conditions are applied on nodes*

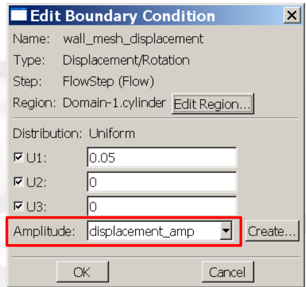
Defining the CFD Model

2. Define boundary conditions (cont'd)

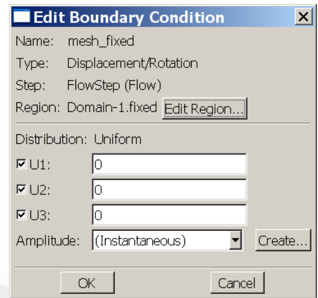
Boundary conditions for mesh motion



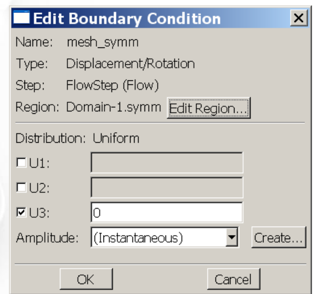
At wall (cylinder surface)



Fixed mesh



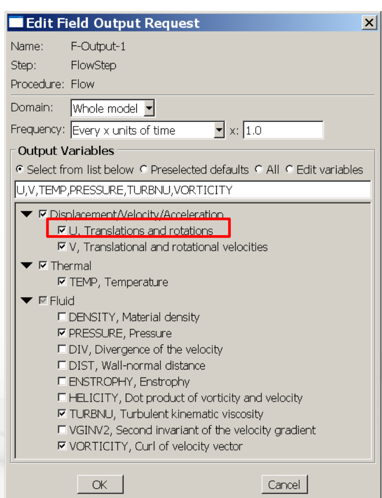
Symmetry



SIMULIA

Defining the CFD Model

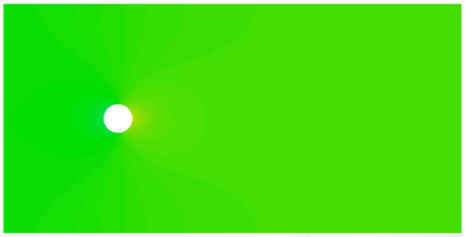
3. Output



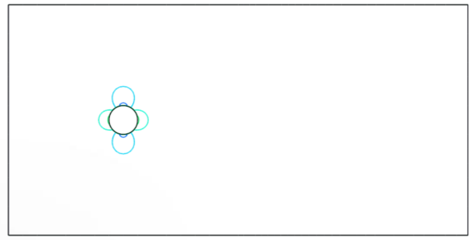
❖ ***Displacement output, U, refers to mesh displacement***

SIMULIA

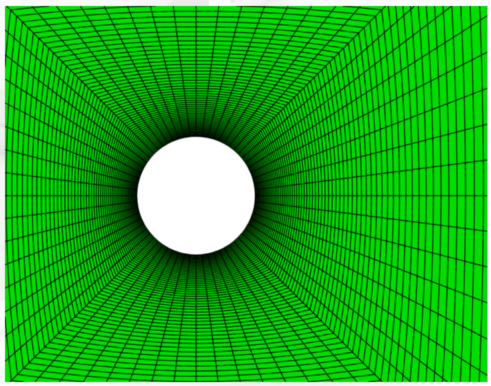
Results



Pressure



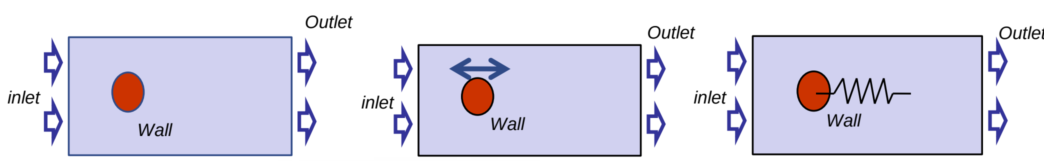
Velocity



Mesh displacement

Setting up CFD Analyses – Summary

Covered in Lecture 6



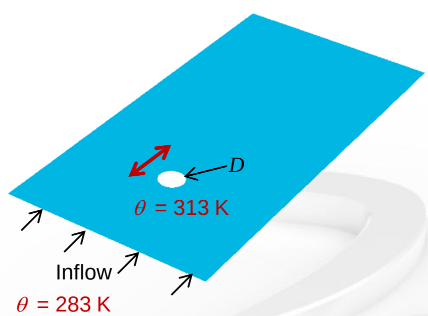
Boundary conditions	On	Flow around a cylinder	Flow around an oscillating cylinder	Flow around a spring loaded cylinder
Wall (Cylinder surface)	Fluid	No-slip, no-penetration **V_{fluid} = 0**	No-slip, no-penetration V_{fluid} = Velocity of the cylinder surface	No-slip, no-penetration V_{fluid} = Velocity of the cylinder surface
Wall (Cylinder surface)	Mesh	None required	U_{mesh} = Displacement of the cylinder surface, **Required for deforming mesh**	U_{mesh} = Displacement of the cylinder surface
Inlet, outlet, top and bottom	Mesh	None required	U_{mesh} = 0 **Required for deforming mesh**	U_{mesh} = 0 **Required for moving deforming mesh**

1. **Do not specify these in the CFD model**
2. **Dictated by structural motion/deformation and FSI interaction**

Modeling Heat Transfer

Introduction

- Flow around an *oscillating* rigid circular cylinder
  - Cylinder surface is now at an elevated temperature while the incoming flow is cooler



$$U_{cylinder} = A_o \sin\left(\frac{2\pi t}{T_p}\right)$$

$$V_{cylinder} = \frac{2\pi A_o}{T_p} \cos\left(\frac{2\pi t}{T_p}\right)$$

V_{inlet} =0.1 m/sec
D = 0.1 m
A_o = 0.05 m
T_p = 2 sec
$\theta_{initial}$ = 283 K

- What do I need to model heat transfer?

Fluid thermal properties
- Specific heat
- Thermal conductivity

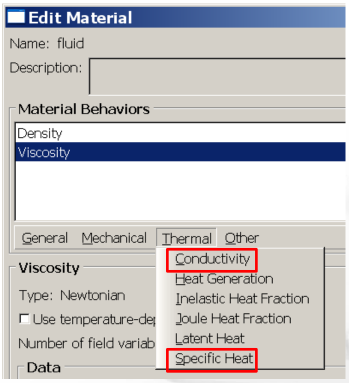
Enable heat transfer
- Linear solver for transport equation

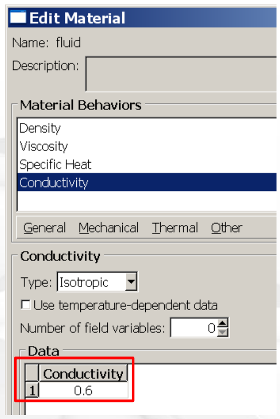
Define thermal boundary conditions and initial conditions
- Temperature at boundaries
- Initial temperature

Relevant output quantities
- Temperature

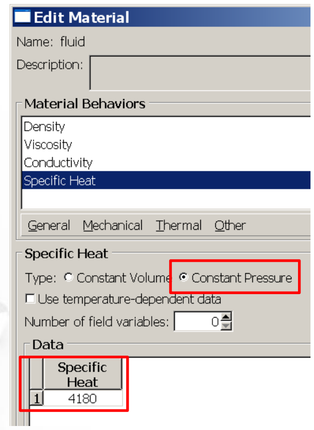
Defining the CFD Model

1. Define thermal properties





Thermal Conductivity

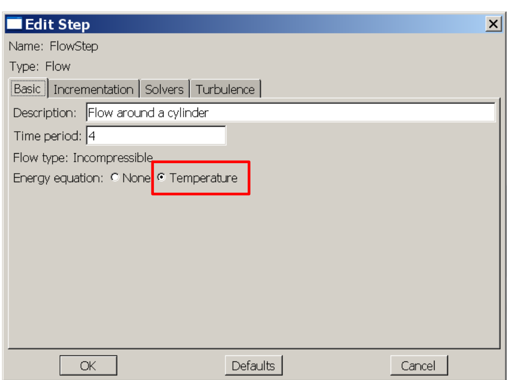


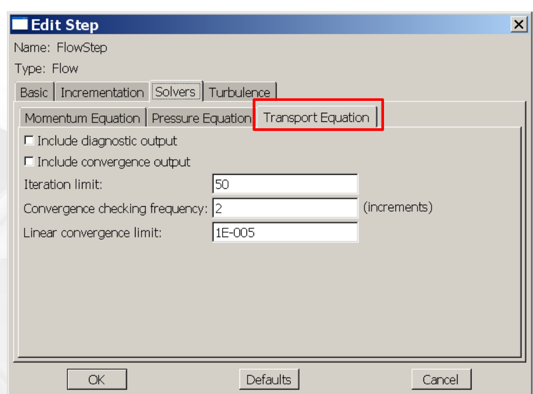
Specific heat at constant pressure

- c_p is required for incompressible flow analysis
- Alternatively, define c_v and universal gas constant R

Defining the CFD Model

2. Activate heat transfer



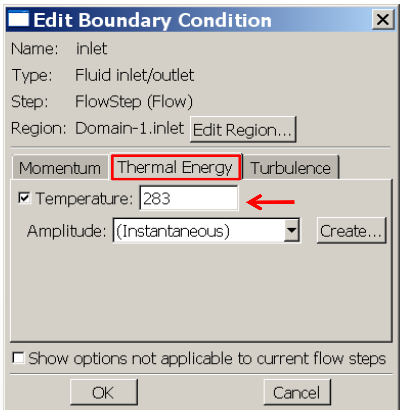


- Activating energy equation enables heat transfer in the flow
  - Requires transport equation solver for temperature
  - Solver options are controlled through transport equation solver controls

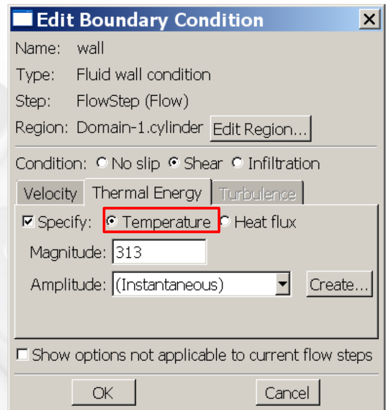
Defining the CFD Model

3. Define boundary conditions

- ***Additional boundary conditions for temperature***
  - Flow inlet
  - Cylinder surface



Define inlet temperature

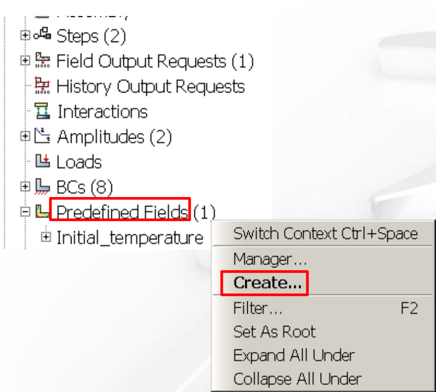


Define cylinder surface temperature

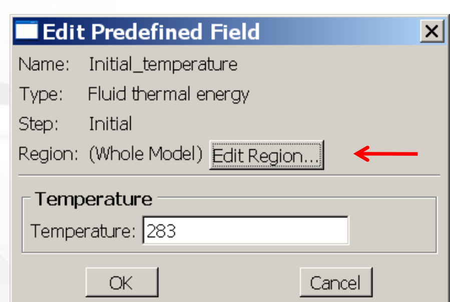
Defining the CFD Model

4. Define initial conditions

- ***Solution of a transient flow problem requires initial conditions***
  - Zero initial velocity is assumed unless non-zero initial velocities are specified
  - Including thermal effects, however, requires specification of initial temperature

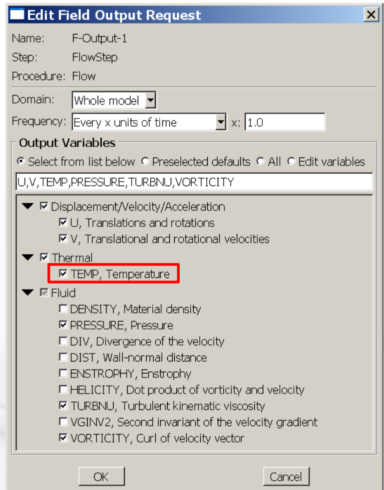


$\theta_{initial}$ = 283 K

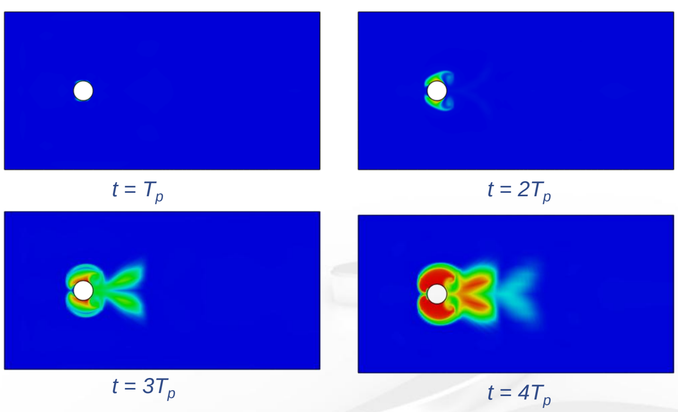


Defining the CFD Model

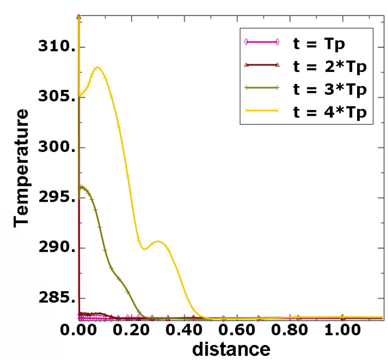
5. Output requests



Results



Temperature contours



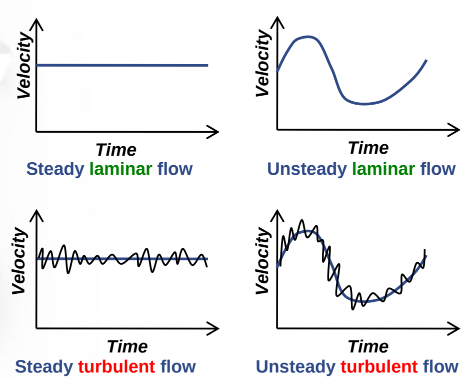
Temperature path plot (Nodes behind the cylinder)

Evolution of Temperature iso-surfaces

Modeling Turbulence

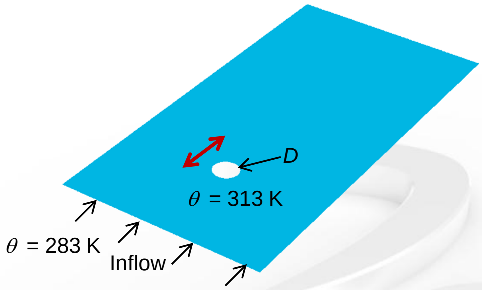
Introduction

- **Many CFD problems are laminar, and do not require the use of a turbulence model**
  - For problems that are truly laminar, use of a turbulence model may yield incorrect results that are too dissipative
- **Before activating a turbulence model, check the Reynolds number for the flow**
  - A very large Reynolds number typically indicates the need for a turbulence model
    - The transition Reynolds number depends on the flow itself, for example:
      - For pipe flows: transition Re ~2300
      - For flow around a rigid circular cylinder: transition Re ~200
  - If you are unsure, try this test:
    1. Run the simulation without a turbulence model activated
    2. Plot kinetic energy and/or the time-history of several flow variables
    3. If there are random oscillations in the results, rerun the simulation with the turbulence model activated to improve the accuracy of the solution



Introduction

- Consider the case of flow around an *oscillating* rigid circular cylinder that is maintained at an elevated temperature
    - But the Reynolds number is now 1000 → Turbulent flow regime

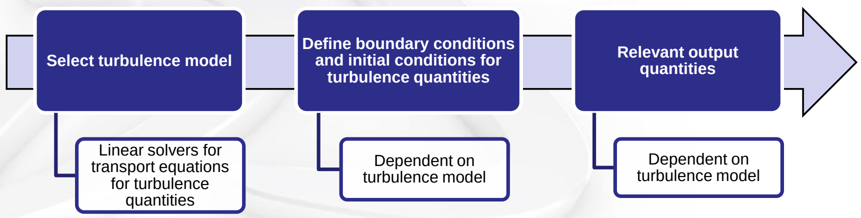


$$U_{cylinder} = A_o \sin\left(\frac{2\pi t}{T_p}\right)$$

$$V_{cylinder} = \frac{2\pi A_o}{T_p} \cos\left(\frac{2\pi t}{T_p}\right)$$

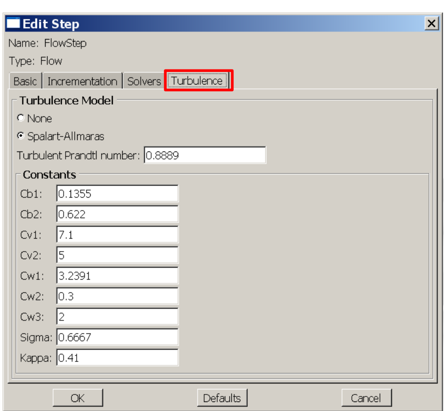
V_{inlet} = ~~0.1 m/sec~~
= 1.0 m/sec
D = 0.1 m
A_o = 0.05 m
T_p = 2 sec
$\theta_{initial}$ = 283 K

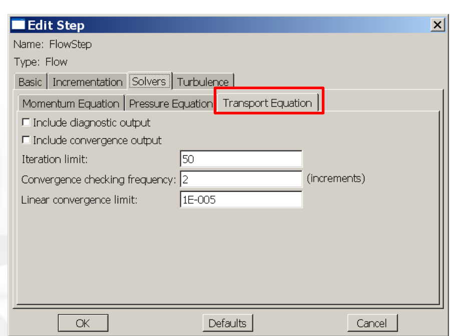
- What do I need to model turbulent flow?



Defining the CFD Model

1. Activate turbulence model





- Turbulence model parameters are characteristic of turbulence models
- Turbulence variables that are solved for depend on particular turbulence model chosen
    - Requires transport equation solver for turbulence variables
    - Solver options are controlled through transport equation solver controls

Defining the CFD Model

2. Boundary conditions

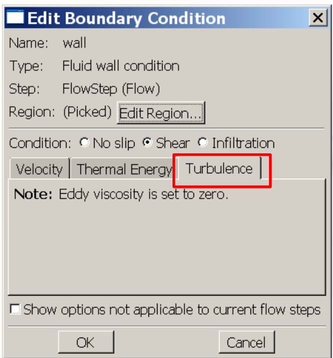
Boundary conditions for turbulence models

- Spalart-Allmaras turbulence model requires specification of modified turbulent kinematic viscosity $\tilde{\nu}$
- Additionally, distance from walls needs to be evaluated

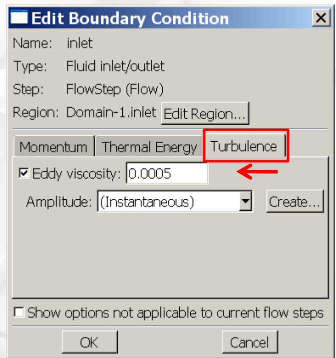
❖ ***Automatically set by Abaqus/CAE at surfaces where wall condition is applied***

At walls :
$\tilde{\nu} = 0$
$d = 0$

At inlet :
Inlet turbulence (ν_t)



Walls



Inlet

❖ ***Inlet turbulence specification methods***

$\nu = \mu / \rho$ ...kinematic viscosity

1. $\nu_t \approx (3-5)\nu$
2. $\nu_t = \sqrt{3/2}u_o Il$

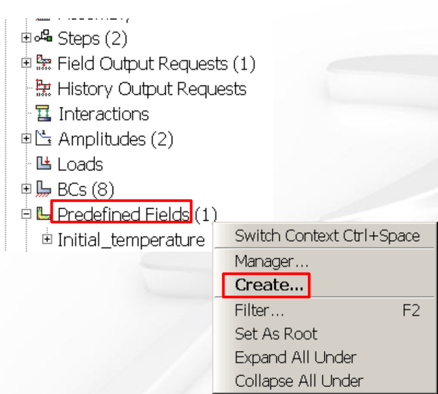
I : Turbulence intensity
l : Turbulence length scale
u_o : Reference velocity

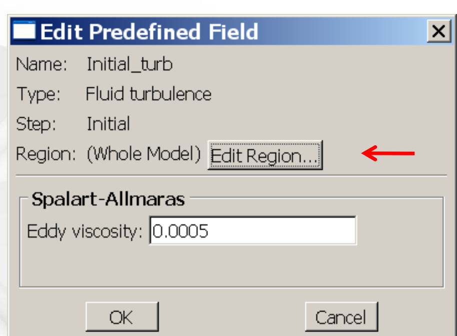
Defining the CFD Model

3. Initial conditions

Initial conditions are required for turbulence variables

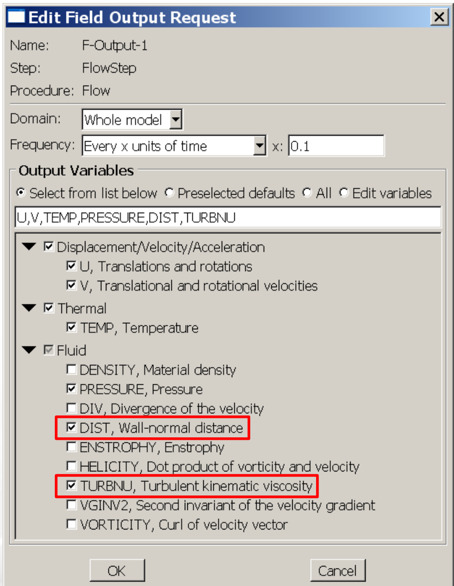
- The Spalart-Allmaras turbulence model requires specification of the initial eddy viscosity
- It is typically specified as being equal to the inlet turbulence





Defining the CFD Model

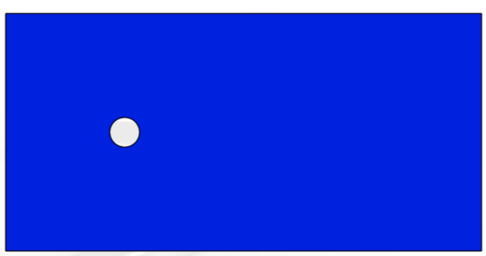
4. Output requests



Distance function output – Distance from walls

Turbulent kinematic viscosity ν_t

Results



Turbulent viscosity

Notes

Notes

CFD Modeling Techniques – Part 1

Lecture 4

Overview

- Material Properties
- Incompressible Flow Analysis Procedure
- Solution Algorithm
- Linear Equation Solvers
- Pressure Equation Solvers
- Momentum Equation Solvers
- Equation Solver Output

Material Properties

Material Properties

Laws of physics + **Material properties (constitutive law for fluids)** = **Fluid physical response**

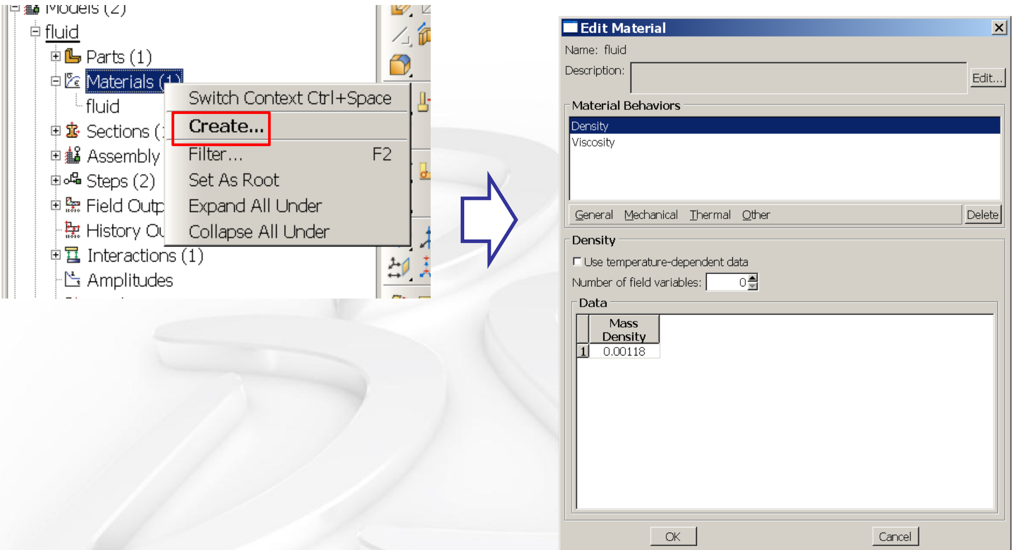
↓

Material properties in fluids analyses

- Density
- Viscosity
- Thermal properties
  - Specific heat, Gas constant
  - Thermal conductivity
  - Thermal expansion coefficient

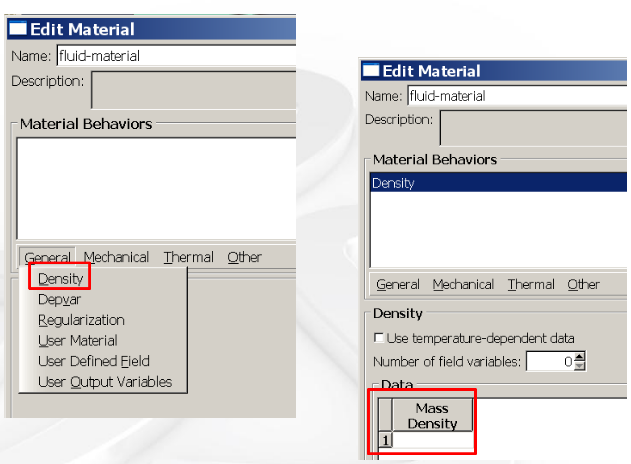
Material Properties

- **Creating fluid materials in Abaqus/CAE**



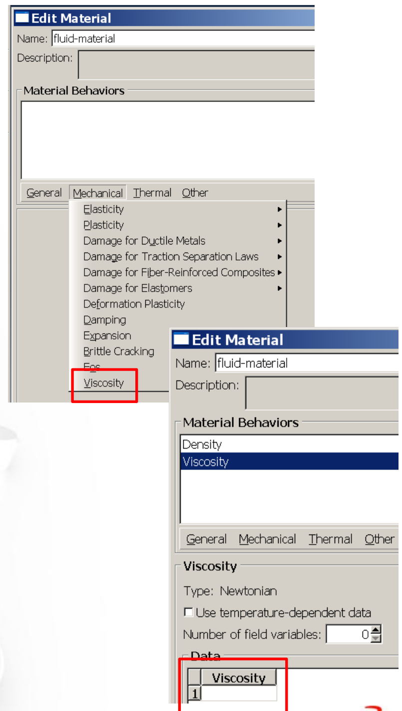
Material Properties

- **Fluid density**
  - *Required* for transient Navier-Stokes computations
  - *Constant* for incompressible flows



Material Properties

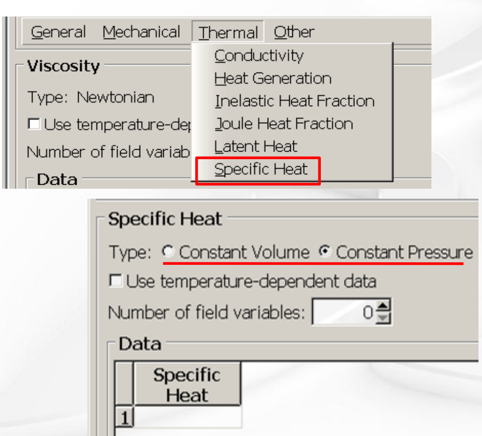
- **Fluid viscosity**
  - Relates fluid shear stress to rate of strain (velocity gradient)
  - Only constant viscosity is supported
    - No support for temperature-dependant or non-Newtonian viscosities
  - Must be specified for *viscous* flows
  - Can also model inviscid flows (zero viscosity)
    - Incompressible Euler equations
    - Viscous effect are ignored
      - High Reynolds number flows away from solid walls
    - Only no-penetration boundary condition is valid at walls
      - No-slip boundary condition is no longer valid at walls

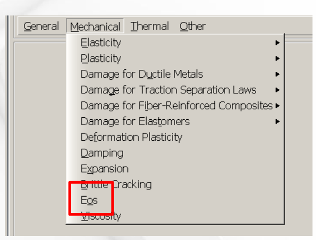


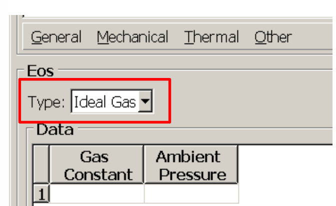
Material Properties

- **Specific heat and gas constant**
  - Specific heat is required for transient Navier-Stokes computations with the energy equation

At constant pressure, c_p	At constant volume, c_v
• Required for incompressible flow • Alternatively, specify c_v and the gas constant, R	• Required for compressible flow



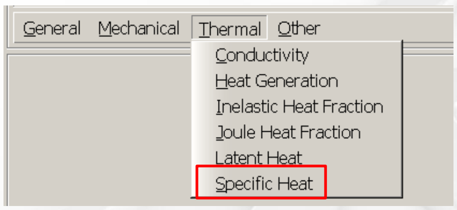


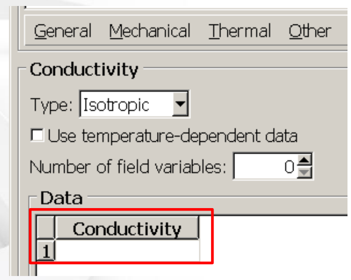


$$c_p - c_v = R$$

Material Properties

- **Thermal conductivity**
  - Required when heat transfer is modeled
    - i.e., when the energy equation is activated
  - Only constant and isotropic thermal conductivity is supported by Abaqus/CFD





Material Properties

- **Thermal expansion coefficient**
  - Required for natural convection flows

Temperature differential causes change in density; lighter fluid rises

Requires compressible N-S equations
Source term proportional to density and gravity

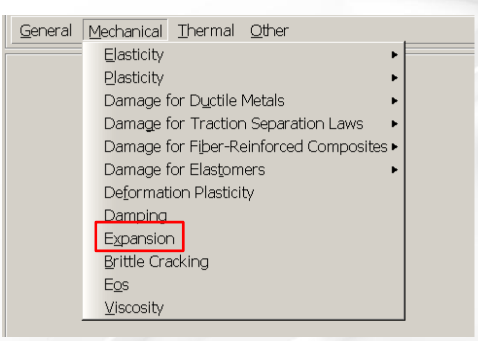
Approximate and model as incompressible flow
(Boussinesq approximation)

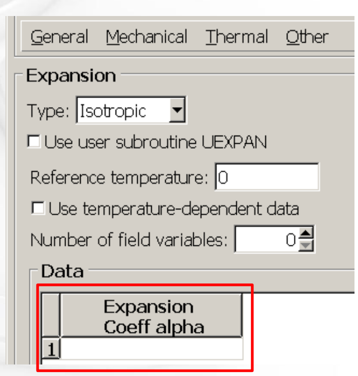
Linearize the source term and relate change in density to change in temperature

$$\mathbf{S} = \rho \mathbf{g}$$

$$\beta = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_P$$

$$(\rho - \rho_0)\mathbf{g} \approx -\rho_0 \beta (T - T_0)\mathbf{g}$$



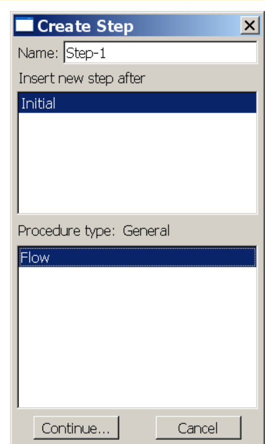


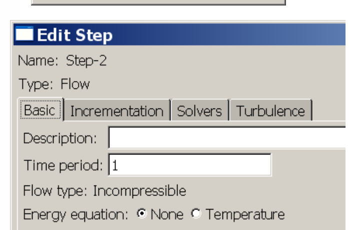
Isotropic thermal expansion coefficient only

Incompressible Flow Analysis Procedure

Incompressible Flow Analysis Procedure

- **The FLOW procedure:**
  - Creates an incompressible fluid analysis
  - Single or multi-step analysis
    - Laminar → Turbulent flow
    - Low Reynolds number flow → High Reynolds number flow
  - Specify:
    - Analysis time
    - Activate energy equation (for modeling heat transfer)
    - Time incrementation
    - Solver settings for momentum, pressure and transport equations
    - Turbulence models





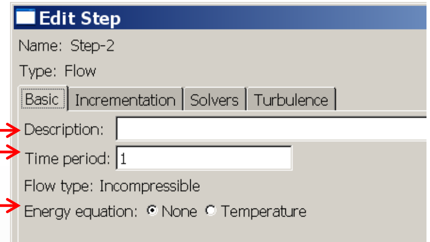
Incompressible Flow Analysis Procedure

- **Basic information**

Specify analysis description

Specify analysis time

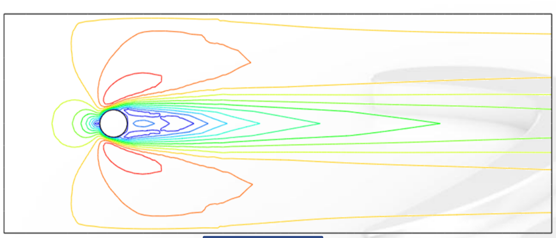
Activate energy equation to include heat transfer



- For analyses reaching steady-state, the time period needs to be chosen so that the monitored variable (kinetic energy, pressure, velocity, or temperature) can reach steady-state
  - Not every CFD problem reaches steady-state!
    - Low Reynolds number flow can go to steady-state
    - Turbulent flows can achieve steady-state in an average sense
- Activating the energy equation includes conduction and convection effects due to heat transfer
  - No support for radiation

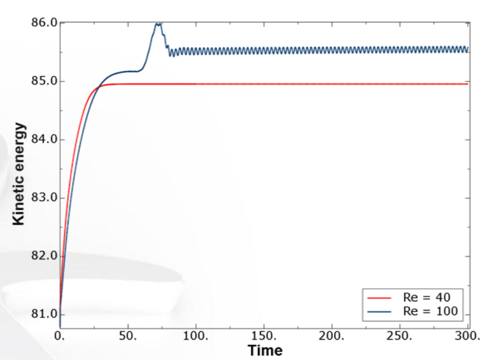
Incompressible Flow Analysis Procedure

- **Tip: Not every flow problem reaches steady-state**
  - Depends on the flow conditions and physics of the flow
  - Example : Vortex shedding



Re = 40

Re = 100



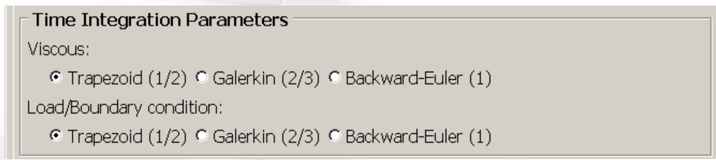
- Re =40 : Steady and symmetric
- Re = 100 : Unsteady, non-symmetric and not yet turbulent!

Incompressible Flow Analysis Procedure

- **Time incrementation**
  - Specify the time integration method for the various terms in the spatially discrete form of the Navier-Stokes equations

$$C^T \mathbf{v} = \mathbf{0}$$
$$M\dot{\mathbf{v}} + A(\mathbf{v})\mathbf{v} + K\mathbf{v} + CP = \mathbf{F}$$

- M : Mass matrix, $A(\mathbf{v})$: Advection operator, K : Viscous diffusion operator
- C : Gradient operator, C^T : Divergence operator

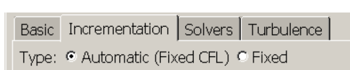


- Recommended method: Trapezoid method with automatic time incrementation
- Backward-Euler method: Use for steady-state problems (transient run reaching steady-state)
- Trapezoid and Galerkin method ~ $O(\Delta t)^2$
- Backward-Euler ~ $O(\Delta t)$

SIMULIA

Incompressible Flow Analysis Procedure

- **Time incrementation control**



Automatic (fixed CFL) time incrementation

- Default and recommended time incrementation method
- Calculates time increment size automatically subject to convective stability limit
  - Courant-Freidrichs-Levy (CFL) condition
  - Arises due to explicit treatment of advective terms in the N-S equations
- Grid Reynolds number and CFL number
- Additionally, avoids numerical oscillations at startup when using trapezoid time integration rule for viscous terms

Fixed time incrementation

- Fixed time increment size
- User has to ensure that the convective stability limit is honored
- May lead to an unstable analysis if the CFL condition is violated

Grid Reynolds number and CFL number

$$\mathrm{Re}_i = \frac{|\mathbf{v}\cdot \boldsymbol{h}_i|}{2\nu}$$

$$CFL_i = \frac{|\mathbf{v}\cdot \boldsymbol{h}_i|\,\Delta t}{\|\boldsymbol{h}_i\|^2}$$

i refers to the element local coordinate directions

SIMULIA

Incompressible Flow Analysis Procedure

- **Time incrementation control (cont'd)**

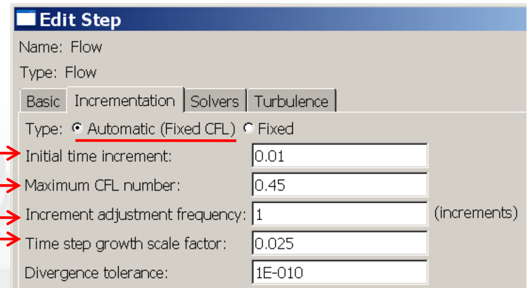
Automatic (fixed CFL) incrementation

User-specified initial time increment (may be violated)

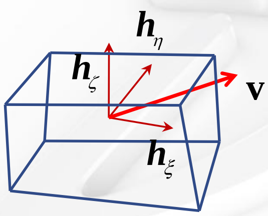
CFL limit

Check and adjust Δt every n increments

Increase Δt by scale factor every n increments

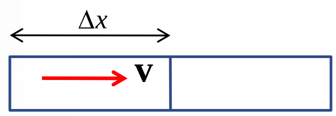


Convective stability limit



$$\Delta t_i \le \frac{CFL \; \|\boldsymbol{h}_i\|}{|\mathbf{v} \cdot \boldsymbol{h}_i|}$$

Minimum over element local directions and over all elements



$$\Delta t \le \frac{CFL \;\; \Delta x}{\mathrm{v}}$$

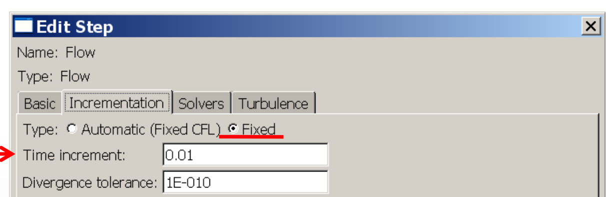
One-dimensional case

Incompressible Flow Analysis Procedure

- **Time incrementation control (cont'd)**

Fixed time incrementation

Fixed Δt

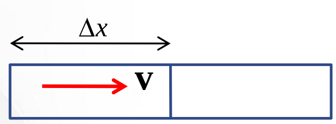


- User-specified, constant time increment
- For stable analysis, choose time increment such that
  - Convective stability limit (CFL condition) is satisfied
  - Avoid numerical oscillations due to viscous (diffusive) terms at startup when using trapezoid time integration rule

Diffusive stability limit

$$\Delta t_i \le \frac{\nu \|\boldsymbol{h}_i\|^2}{1 + \sqrt{1 + \mathrm{Re}_i^{\;2}}}$$

Minimum over element local directions and over all elements

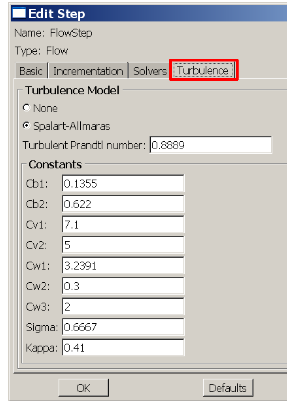


$$\Delta t \le \frac{CFL \;\; \Delta x}{\mathrm{v}} \qquad \Delta t \le \frac{0.5 \;\; \Delta x^{\;2}}{\nu}$$

One-dimensional case

Incompressible Flow Analysis Procedure

- **Turbulence**
  - For turbulent flows, you can activate a turbulence model
    - Turbulence model parameters are characteristic of turbulence models
    - Turbulence solution variables depend on the particular turbulence model chosen
      - Requires transport equation solver for turbulence variables
      - Solver options are controlled through transport equation solver controls
    - If a turbulence model is not chosen for high Reynolds number flow which is expected to be turbulent, implicit large eddy simulation (ILES) ensue.
    - Each turbulence model will require initial conditions and boundary conditions specific to that turbulence model
      - Will be discussed further when discussing initial and boundary conditions (Lecture 5)



Solution Algorithm

Solution Algorithm

- Based on Semi-Implicit Projection Method

$$\rho\frac{\partial \mathbf{v}}{\partial t}+\nabla p=\mathbf{f}+\mu\nabla^2\mathbf{v}-\rho\mathbf{v}\cdot\nabla\mathbf{v}$$ Momentum equation

$$\nabla\cdot\mathbf{v}=0$$ Continuity equation

Step 1 **Momentum Solve**

$$\rho\frac{\tilde{\mathbf{v}}^{n+1}-\mathbf{v}^{n}}{\Delta t}+\cancel{\rho\frac{\mathbf{v}^{n+1}-\tilde{\mathbf{v}}^{n+1}}{\Delta t}+\nabla p^{n+1}}=\Phi\left(\mathbf{v}^{n},\tilde{\mathbf{v}}^{n+1},p^{n},\ldots\right)$$

$$\nabla\cdot\tilde{\mathbf{v}}^{n+1}\neq 0$$

- Ignore pressure gradient at time $n+1$ and calculate an intermediate velocity
- The intermediate velocity $\tilde{\mathbf{v}}^{n+1}$ is not divergence free
- Φ is a function of velocity: its precise form and dependence on time n or $n+1$ depends on how various terms in the N-S equations are treated – explicit, semi-implicit or implicit
- **Linear system of equations if advection is treated explicitly!**

Solution Algorithm

Step 2 **Pressure Solve**

$$\rho\frac{\mathbf{v}^{n+1}-\tilde{\mathbf{v}}^{n+1}}{\Delta t}=-\nabla\left(p^{n+1}-p^{n}\right)$$

Divergence

$$\nabla^2\left(p^{n+1}-p^{n}\right)=\frac{\rho}{\Delta t}\nabla\cdot\tilde{\mathbf{v}}^{n+1}$$

- Poisson's equation for pressure
- Yields correct pressure for next time increment
- Utilize $\nabla\cdot\mathbf{v}^{n+1}=0$

Step 3 **Obtain velocity**

$$\mathbf{v}^{n+1}=\tilde{\mathbf{v}}^{n+1}-\frac{\Delta t}{\rho}\nabla\left(p^{n+1}-p^{n}\right)$$

- Pressure velocity decoupling
- Gives a divergence-free velocity field every time increment
- Additional quantities to solve
  - Energy
  - Turbulence variables including distance function

Solution Algorithm

Log file

```
Inc.    dt        Time      U-Min.    U-Max.    V-Min.    V-Max.    W-Min.    W-Max.    T-Min.    T-Max
===== ========== ========== ========== ========== ========== ========== ========== ========== ========== ==========
  0  1.00e-02  0.00e+00  1.76e-01  1.84e+00  -8.27e-01  8.27e-01  -1.01e-03  5.04e-04  2.83e+02  2.83e+02

 -- X-Momentum --
 --- Solution converged ---

    iterations taken  : 8
    ||r ||/||b|| error: 4.271e-09
    ||dx||/||x|| error: 1.112e-06

 -- Y-Momentum --
 --- Solution converged ---

    iterations taken  : 8
    ||r ||/||b|| error: 2.204e-07
    ||dx||/||x|| error: 2.011e-06

 -- Z-Momentum --
 --- Solution converged ---

    iterations taken  : 8
    ||r ||/||b|| error: 1.106e-06
    ||dx||/||x|| error: 5.638e-06
```

Momentum Solve

```
 -- S-A Transport Model Equation --
 --- Solution converged ---

    iterations taken  : 10
    ||r ||/||b|| error: 6.583e-08
    ||dx||/||x|| error: 5.566e-06
```

Spalart-Allmaras Turbulence Solve

```
 -- Energy Equation --
 --- Solution converged ---

    iterations taken  : 4
    ||r ||/||b|| error: 1.556e-08
    ||dx||/||x|| error: 6.384e-06
```

Energy Solve

```
 -- Pressure Increment –
 --- Solution converged ---

    iterations taken  : 14
    ||r ||/||b|| error: 3.056e-06
    ||dx||/||x|| error: 5.964e-06
```

Pressure Solve

Solution Algorithm

- Each equation is a linear system (explicit advection scheme)
- Require linear equation solvers ($Ax = b$)

Physics	Solution quantities	Number of linear systems
Flow only	Pressure, velocity	4
Flow and energy	Pressure, velocity, temperature	5
Flow and turbulence	Pressure, velocity, wall-normal distance, turbulence quantities	6 or more*
Flow, energy and turbulence	Pressure, velocity, temperature, wall-normal distance, turbulence quantities	7 or more*

- For flow problems, wall-normal distance is calculated once at the beginning of the analysis.
- For flow problems where arbitrary Lagrangian-Eulerian (ALE) is activated (prescribed boundary motion or FSI), wall-normal distance will be updated more frequently
- Number of turbulence variables depends on turbulence model chosen

Linear Equation Solvers

SIMULIA

Linear Equation Solvers

$$Ax = b$$

x: unknown quantity, A: Coefficient matrix, b: right hand side

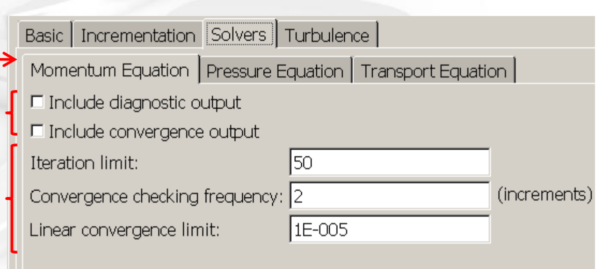
- Due to large problem sizes, Abaqus/CFD uses iterative solvers to solve the linear systems of equations
  - Solve the equation iteratively until convergence or maximum iteration limit

Residual norm $\dfrac{\|b - Ax^{i+1}\|}{\|b\|} < \delta$ $\qquad$ $\dfrac{\|x^{i+1} - x^{i}\|}{\|x^{i+1}\|} < \varepsilon$ **Solution increment**

Equations

Solver diagnostic & convergence output

Iteration limit, convergence checking frequency and linear convergence limit, δ



SIMULIA

Linear Equation Solvers

- **Iterative techniques**
  - An iterative method attempts to solve a system of equations by finding successive approximations to the solution starting with an initial guess

$$Ax = b$$

  - The Krylov subspace method is one such iterative technique
    - Conjugate gradient (CG) method
    - Flexible generalized minimum residual method (FGMRES)
    - Biconjugate gradient method (BiCG)
  - The convergence and robustness of iterative solvers is often accelerated by ***preconditioning*** the linear system of equations
    - Transform the original linear system into one that has the same solution but which is likely to be easier to solve with an iterative solver

$$Ax = b$$

$$AP^{-1} \quad Px = b$$

$$AP^{-1} \quad y = b, \quad x = P^{-1}y$$

Linear Equation Solvers

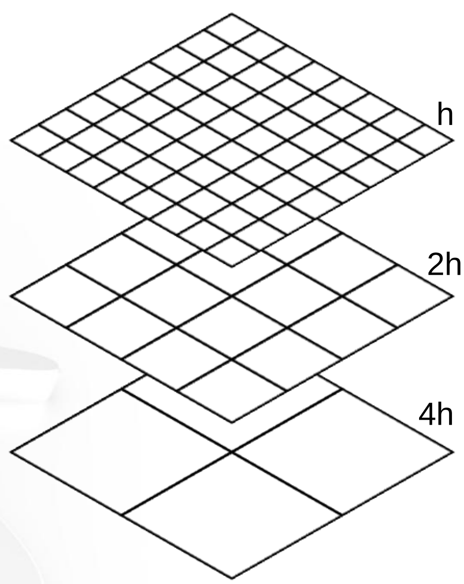
- A system of equations is easier or tougher to solve depending on its condition number
  - Small condition number indicates well-conditioned system
    - Small change in the coefficient matrix or the right hand side results in a small change in the solution vector
    - Easier to solve
  - Large condition number indicates ill-conditioned system
    - Small change in the coefficient matrix or the right hand side results in a large change in the solution vector
    - Tougher to solve

Linear Equation Solvers

- **Algebraic Multigrid (AMG) Technology**
  - Theoretically AMG is a scalable iterative method in the sense that the number of iterations to convergence is not dependent on the mesh refinement for a given problem
    - This in turn ensures the cost of the solution grows proportional to the number of unknowns
  - AMG is most commonly used as a preconditioner for Krylov space iterative solvers
    - Well suited for large systems of equations
    - AMG preconditioners have many settings in general
    - A small subset of these are provided for cases where default settings or presets do not work or in the case performance optimization is desired
      - Smoother type
      - Number of smoother applications at each grid level

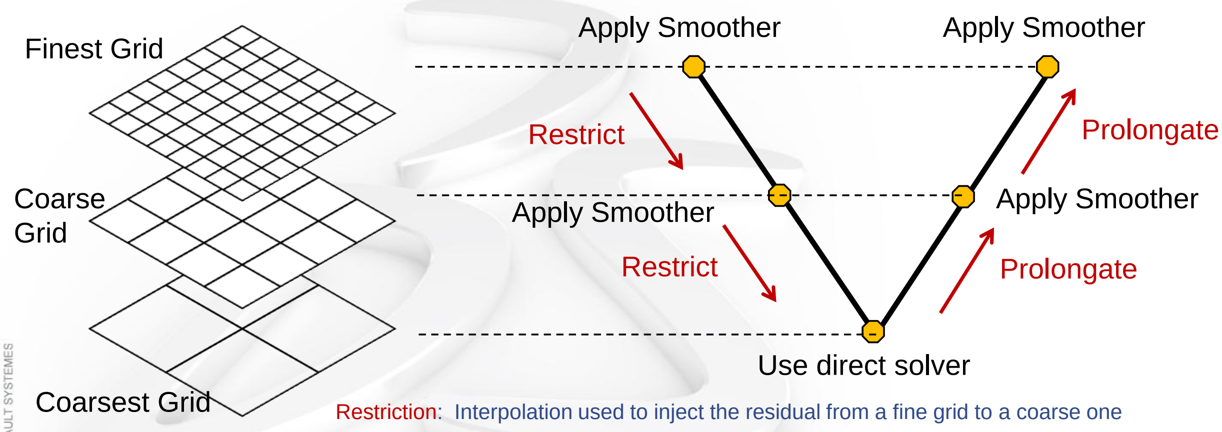
Linear Equation Solvers

- **Why use multiple grids?**
  - Although AMG is purely algebraic (it only requires a matrix A and a RHS vector b), it is much easier to explain the main idea geometrically
  - The main idea is to recursively create coarser versions of a problem where low-frequency errors on a grid level can be represented as high-frequency errors on the next coarse level
    - High-frequency errors are preferred because very inexpensive iterative solvers are available to eliminate them
    - In the context of AMG, these inexpensive iterative solvers used on multigrid levels are called SMOOTHERS (e.g., Incomplete factorization, Chebychev)



Linear Equation Solvers

- **Description of AMG algorithm**
  - Coarsening continues until the grid size is small enough to be solved using a direct solver
    - The number of grids depends on the problem size
  - The most common grid “visiting” scheme is called a V-Cycle and this is used for each solver iteration within Abaqus/CFD when AMG preconditioning is used



Restriction: Interpolation used to inject the residual from a fine grid to a coarse one

Prolongation: Interpolation used to inject the correction from the coarse grid to the finer grid

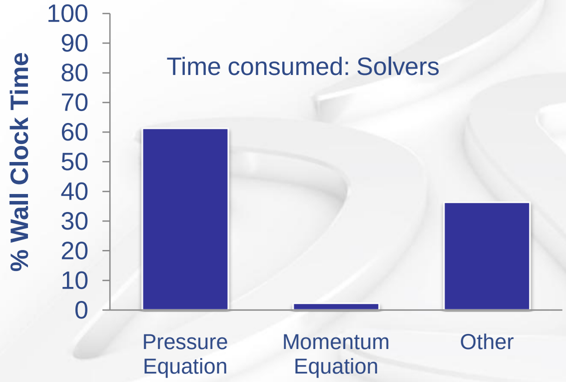
Linear Equation Solvers

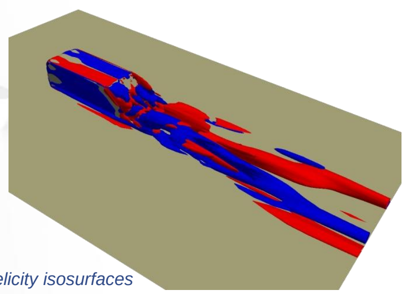
Linear Solvers			
	Preconditioner	Solver	Smoother (AMG only)
Momentum Equation	Diagonally scaled (DS)	• Flexible Generalized Minimal Residual method (FGMRES)	NA
Pressure Equation	• Algebraic Multi-Grid (AMG) • Symmetric Succesive Over-Relaxation (SSOR)	• Conjugate Gradient (CG) • Bi-conjugate Gradient Stabilized (BiCGSTAB) • Flexible Generalized Minimal Residual method (FGMRES)	• Incomplete Cholesky Factorization (ICC) • Polynomial (Chebyshev)
Transport Equation	• Diagonally scaled	• Flexible Generalized Minimal Residual method (FGMRES)	NA

Pressure Equation Solvers

Pressure Equation Solvers

- The solution to the pressure equation is the most time-consuming
  - Poisson's equation for pressure is *global* in nature
  - A range of solver choices are available for solving the pressure equation
- AMG preconditioning is available with all three solver choices: CG, BiCGSTAB and FGMRES
- SSOR preconditioning is only available with CG solver
- Smoothers are only applicable when AMG preconditioning is used





Helicity isosurfaces

Prototype Car Body (Ahmed's body)

16 Cores
Solution physical time = 0.1 sec

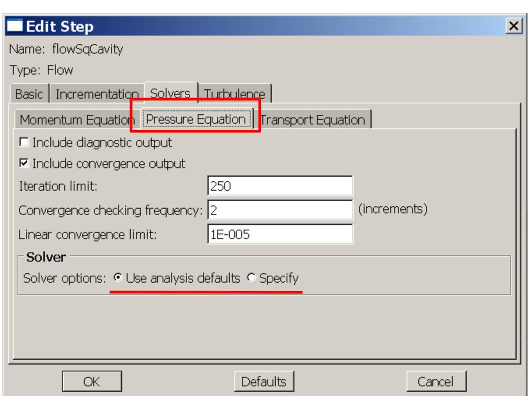
Pressure Equation Solvers

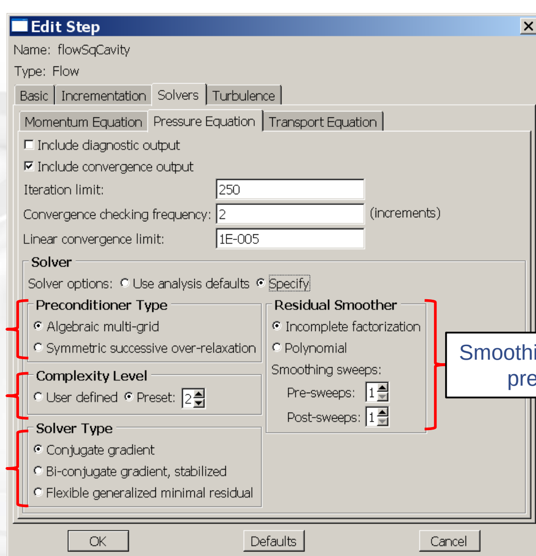
- Each of the equation solvers has controls for
  - Maximum number of iterations allowed
  - Convergence checking frequency
  - Linear convergence tolerance
  - Convergence output
  - Solver diagnostics

Equation	Maximum number of iterations (default)	Convergence checking frequency (default)	Linear convergence tolerance (default)	Convergence output (default)	Solver diagnostics (default)
Momentum	50	2	1e–5	OFF	OFF
Pressure	250 (AMG preconditioned solvers) 1000 (SSOR preconditioned CG solver)	2	1e–5	OFF	OFF
Transport	50	2	1e–5	OFF	OFF

Pressure Equation Solvers

- **Solver presets**





Preconditioner choices

Solver preset levels

Equation solvers

Smoothing for AMG preconditioning

Pressure Equation Solvers

- AMG preconditioned solvers are the recommended and default method for solving the pressure-Poisson equation

Preset 1
- AMG preconditioner
- Chebyshev smoother
- 2 smoothing sweeps
- CG solver

Preset 2 (Default)
- AMG preconditioner
- ICC smoother
- 1 smoothing sweep
- CG solver

Preset 3
- AMG preconditioner
- ICC smoother
- 1 smoothing sweep
- BiCGSTAB solver

Increasing complexity, increasing iteration cost & robustness

- Preset 1 is suitable for problems with good quality meshes
  - Uniform or slightly distorted meshes
  - Chebyshev smoother is less expensive than Incomplete factorization (ICC) but works best with meshes of good quality
    - May have performance benefit over Preset 2

SIMULIA

Pressure Equation Solvers

- Preset 2
  - CG solver is very robust in general but might not converge if mesh is too distorted or has a very high aspect ratio
- Preset 3
  - BiCGSTAB solver is the most robust for meshes that are highly distorted and/or have high aspect ratios
  - Computationally more expensive than CG solver
  - Often used for FSI problems where mesh motion can distort the mesh
- Other choices:
  - SSOR preconditioned CG solver
    - Solver of “last resort”
    - May work for some problems where AMG preconditioning has failed
      - Example: Problems with high aspect ratio elements
    - Requires significantly larger number of iterations
      - Poor performance

SIMULIA

Pressure Equation Solvers

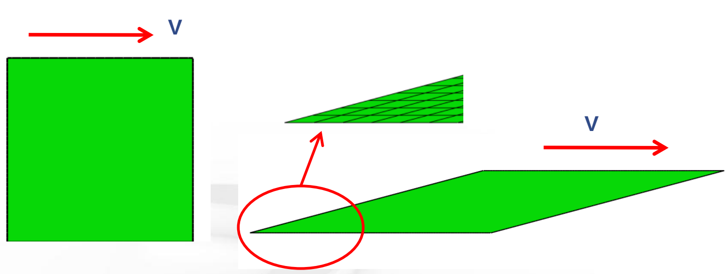
- **Convergence**
  - If the solution ($Ax = b$) to the pressure equation does not converge within a given time increment, a warning is issued but the analysis continues
    - If the solution to the pressure equation fails to converge for 50 continuous time increments, the analysis terminates with an error:

```
***ERROR: Pressure Poisson Equation - Too many non-converged   iterations:
                  Possible diagnostics are: (A) Check the aspect ratio of elements
                  (ratio of element length in the flow direction to that of the
                  length in the flow normal direction). Try increasing the number of
                  iterations for the pressure solver, or changing the preset solver
                  option to 3, or reducing the aspect ratio of the elements in the
                  mesh. (B) Check the validity of the specified boundary conditions.
                  For example, (i) if outflow boundary conditions are specified,
                  check to see if proper surfaces are included, (ii) if gravity is
                  present, check if the hydrostatic head is taken into account for
                  any outflow pressure boundary conditions. (C) If this a
                  Fluid-Structure Interaction problem, the amount of mesh
                  deformation may be excessive. Permitting the mesh boundaries to
                  slip in some areas may reduce element distortion
```

  - Possible reasons for nonconvergence:
    - Large aspect ratio of elements
    - Incorrectly specified boundary conditions
    - Excessive mesh deformation for problems with prescribed displacement or fluid-structure interaction

Pressure Equation Solvers

- **Effect of mesh density and skewness**



Square cavity
(Flow Re = 100)

15º cavity
(Flow Re = 100)

- Uniform mesh for flow in the square cavity
- Uniform but distorted mesh for flow in 15º cavity

How do linear solvers behave?

- Effect of mesh refinement
- Effect of mesh skewness

Pressure Equation Solvers

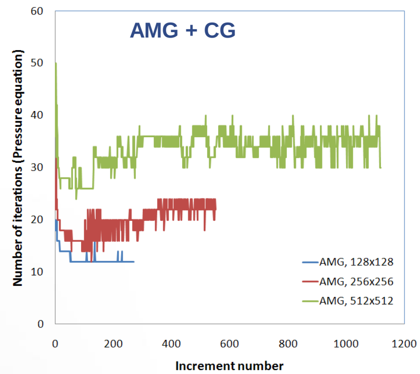
- Effect of mesh refinement (square cavity)

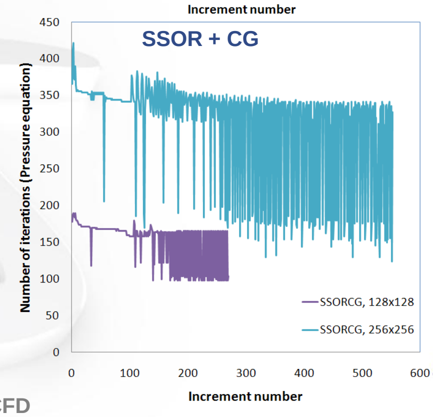
All runs are done on 8 cores
Solution physical time : 1 sec

Increased iteration count with mesh refinement

Mesh Resolution	Average number of Iterations (PPE) AMG + ICC + CG	Average number of Iterations (PPE) SSOR + CG
128x128	12	151
256x256	20	308
512x512	34	-

Significant increase in iteration count going from AMG preconditioning to SSOR preconditioning

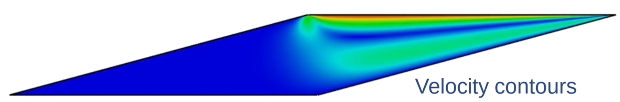




Pressure Equation Solvers

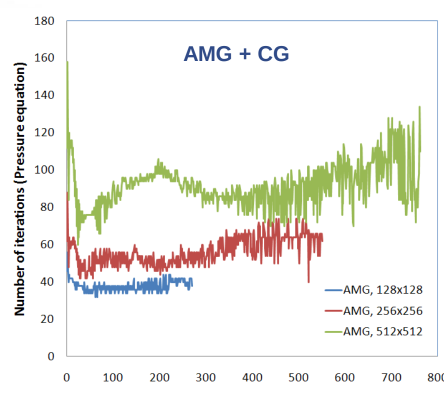
- Effect of mesh refinement (15º cavity)

All runs are done on 8 cores
Solution physical time : 1 sec



Velocity contours

Mesh Resolution	Average number of Iterations (PPE) AMG + ICC + CG	Average number of Iterations (PPE) SSOR + CG
128x128	38	554
256x256	57	Fails to converge in 1000 iterations
512x512	92	-



- Significant increase in iteration count between AMG and SSOR
- Aggravated by mesh skewness

Pressure Equation Solvers

Effect of mesh skewness (square vs. 15º cavity)

Mesh Resolution	Problem	Average number of Iterations (PPE) AMG + ICC + CG	Average number of Iterations (PPE) SSOR + CG
128x128	square cavity	12	151
	15º cavity	38	554
256x256	square cavity	20	308
	15º cavity	57	Fails to converge in 1000 iterations
512x512	square cavity	34	-
	15º cavity	92	-

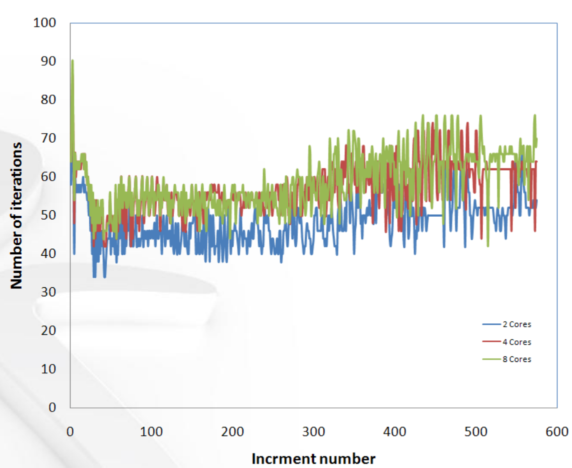
- Problem with skewed mesh requires more iterations to converge
- Effect on SSOR + CG is much more dramatic

Pressure Equation Solvers

- Effect of parallel processing (15º cavity)

Mesh size: 256x256
Solution physical time : 1 sec

# of cores	Average number of Iterations (PPE) AMG + ICC + CG
2	48
4	57
8	59



- Number of iterations may vary slightly with number of cores

Momentum Equation Solvers

SIMULIA

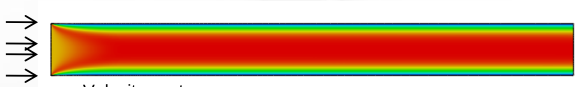
Momentum Equation Solvers

- The momentum equation requires far fewer iterations to solve

```
***WARNING: The x-momentum equation may not have fully converged
```

- If viscous effects are dominant (for low Reynolds number flows), the momentum equation may have trouble converging
  - Increase the number of allowed iterations (default = 50)

Viscous effect (Channel flow)



Velocity contours

Increasing viscous effect

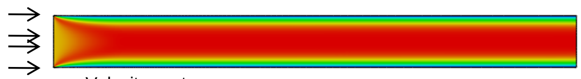
Reynolds number	Average number of iterations (X - momentum equation) (first 5 increments)	Average number of iterations (Y - momentum equation) (first 5 increments)
100	13	16
1	48	50

Increasing number of iterations

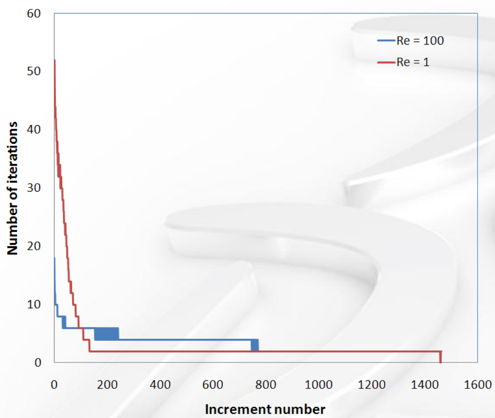
SIMULIA

Momentum Equation Solvers

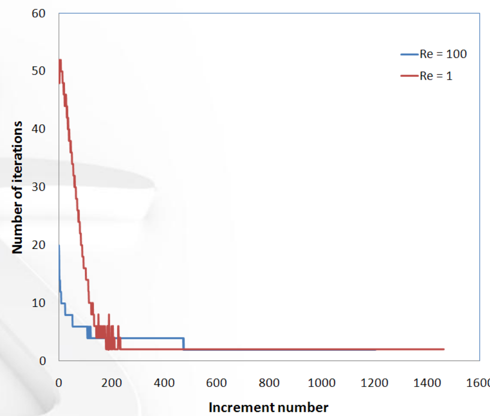
Viscous effect (Channel flow)



X Momentum



Y Momentum



Momentum Equation Solvers

- **Convergence**
  - If the solution to the momentum equation does not converge within a given time increment, a warning is issued but the analysis continues
    - If the solution to the momentum equation fails to converge for 50 continuous time increments, the analysis terminates with an error

```
***ERROR: X-momentum equation - Too many non-converged iterations: Possible
   diagnostics are: (A) The flow may be a highly viscous flow
  (typically, Reynolds number less than 1). If so, try increasing
  the number of iterations for the momentum solvers. (B) Some
  elements in the mesh may be badly distorted. Check the quality of
  the elements. If this is a deforming mesh problem, check to see if
  the displacement boundary conditions for the mesh are physical and
 check the mesh velocities. If the mesh velocities are too high,
 decreasing the time step value may reduce severe mesh distortion.
 (C) Boundary conditions specifications may not be consistent or
 may be erroneous.Check the boundary surfaces and see if the
 specification of a particular boundary condition is meaningful for
 that surface. Example of an improper boundary condition is the
 specification of both velocity and pressure boundary conditions
 for the same surface.
```

- Possible diagnostics
  - Viscous effects dominate
  - Badly distorted elements
  - Incorrectly specified boundary conditions

```
-- X-Momentum --
   !!! Iteration failed to converge in specified number of passes !!!
 reason: -3

      iterations taken  : 52
      ||r ||/||b|| error: 0.0002361
      ||dx||/||x|| error: 0.0001613

  ***WARNING: The x-momentum equation may not have fully
converged
```

- For some problems, the equation may fail to converge at start up but recover afterwards
  - Check the residual to verify if the convergence is acceptable

Equation Solver Output

SIMULIA

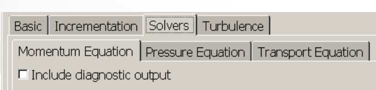
Equation Solver Output

- **Convergence output**

Log file

```
 Inc.       dt            Time          U-Min.     U-Max.     V-Min.     V-Max.     W-Min.       W-Max.     T-Min.       T-Max
===== ========== ========== ========== ========== ========== ========== ========== ========== ========== ==========
    0   1.00e-02   0.00e+00  1.76e-01   1.84e+00  -8.27e-01   8.27e-01  -1.01e-03   5.04e-04   2.83e+02   2.83e+02

    -- X-Momentum --
    --- Solution converged ---

         iterations taken   : 8
         ||r ||/||b|| error: 4.271e-09
         ||dx||/||x|| error: 1.112e-06

    -- Y-Momentum --
    --- Solution converged ---

         iterations taken   : 8
         ||r ||/||b|| error: 2.204e-07
         ||dx||/||x|| error: 2.011e-06

    -- Z-Momentum --
    --- Solution converged ---

         iterations taken   : 8
         ||r ||/||b|| error: 1.106e-06
         ||dx||/||x|| error: 5.638e-06

    -- S-A Transport Model Equation --
    --- Solution converged ---

         iterations taken   : 10
         ||r ||/||b|| error: 6.583e-08
         ||dx||/||x|| error: 5.566e-06

    -- Energy Equation --
    --- Solution converged ---

         iterations taken   : 4
         ||r ||/||b|| error: 1.556e-08
         ||dx||/||x|| error: 6.384e-06

    -- Pressure Increment -
    --- Solution converged ---

         iterations taken   : 14
         ||r ||/||b|| error: 3.056e-06
         ||dx||/||x|| error: 5.964e-06
```

- No convergence output is written by default
- Convergence output for each equation solver can be individually turned on



- Number of iterations taken, final residual and final solution increment is written

SIMULIA

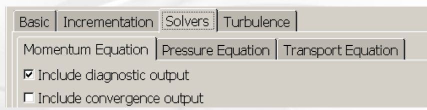
Equation Solver Output

- **Diagnostic output**

Log file

Momentum Equation

```
solver: ||b|| = 2.599e-06
 solver: ||x(0)|| = 0.0148
 solver: Iteration: 2 ||r|| = 1.834e-08 ||x(i+1)-x(i)|| = 0.0004975 ||x(i+1)|| = 0.01417
 solver: Iteration: 4 ||r|| = 1.036e-09 ||x(i+1)-x(i)|| = 2.863e-05 ||x(i+1)|| = 0.01415
 solver: Iteration: 6 ||r|| = 5.4e-11 ||x(i+1)-x(i)|| = 1.517e-06 ||x(i+1)|| = 0.01415
 solver: Iteration: 8 ||r|| = 2.874e-12 ||x(i+1)-x(i)|| = 7.977e-08 ||x(i+1)|| = 0.01415

  -- Z-Momentum --
--- Solution converged ---
       iterations taken  : 8
       ||r ||/||b|| error: 1.106e-06
       ||dx||/||x|| error: 5.638e-06
```



- No diagnostic output is written by default
- Diagnostic output for each equation solver can be individually turned on

Equation Solver Output

- **Diagnostic output (cont'd)**

Pressure Equation
Algebraic Multi Grid information

Log file

AMG coarser levels

```
Algebraic Multigrid Solver Information
======================================================
       |    Num.   |   Ave.   |    Max.    | Condition
 Level | Equations | nnzs/row | Eigenval.  | Num. Est.
------------------------------------------------------
     0 |    526338 |     18.0 |  2.63e+00 |  1.15e+04
     1 |     29241 |      8.9 |  1.82e+00 |  1.73e+04
     2 |      3249 |      8.8 |  1.85e+00 |  1.00e+04
     3 |       361 |      8.4 |        N/A |       N/A
======================================================
Operator Complexity: 1.031

solver: ||b|| = 0.34608
solver: ||x(0)|| = 0
solver: Iteration: 2 ||r|| = 0.38375 ||x(i+1)-x(i)|| = 24897 ||x(i+1)|| = 47117
solver: Iteration: 4 ||r|| = 0.1275 ||x(i+1)-x(i)|| = 4716.8 ||x(i+1)|| = 45701
solver: Iteration: 6 ||r|| = 0.029435 ||x(i+1)-x(i)|| = 625.4 ||x(i+1)|| = 42878
solver: Iteration: 8 ||r|| = 0.0067216 ||x(i+1)-x(i)|| = 192.43 ||x(i+1)|| = 43078
solver: Iteration: 10 ||r|| = 0.0015552 ||x(i+1)-x(i)|| = 26.183 ||x(i+1)|| = 43166
solver: Iteration: 12 ||r|| = 0.00052982 ||x(i+1)-x(i)|| = 24.867 ||x(i+1)|| = 43147
solver: Iteration: 14 ||r|| = 0.00053732 ||x(i+1)-x(i)|| = 39.872 ||x(i+1)|| = 43117
solver: Iteration: 16 ||r|| = 0.00023122 ||x(i+1)-x(i)|| = 21.882 ||x(i+1)|| = 43095
solver: Iteration: 18 ||r|| = 6.1186e-05 ||x(i+1)-x(i)|| = 2.8974 ||x(i+1)|| = 43090
solver: Iteration: 20 ||r|| = 1.8921e-05 ||x(i+1)-x(i)|| = 0.24363 ||x(i+1)|| = 43089
solver: Iteration: 22 ||r|| = 7.1782e-06 ||x(i+1)-x(i)|| = 0.083647 ||x(i+1)|| = 43089
solver: Iteration: 24 ||r|| = 4.004e-06 ||x(i+1)-x(i)|| = 0.039297 ||x(i+1)|| = 43089
solver: Iteration: 26 ||r|| = 9.4357e-07 ||x(i+1)-x(i)|| = 0.010813 ||x(i+1)|| = 43089


 -- Pressure Increment --

 --- Solution converged ---
      iterations taken  : 26
      ||r ||/||b|| error: 2.7264e-06
      ||dx||/||x|| error: 2.5095e-07
```

Notes

Notes

CFD Modeling Techniques – Part 2

Lecture 5

Overview

- Initial Conditions
- Boundary Conditions and Loads
- Output
- Deforming Mesh
- Monitoring Convergence

Initial Conditions

Initial Conditions

- **The solution to the transient Navier-Stokes equations requires initial conditions be specified**
  - Conditions on solution quantities at time $t = 0$

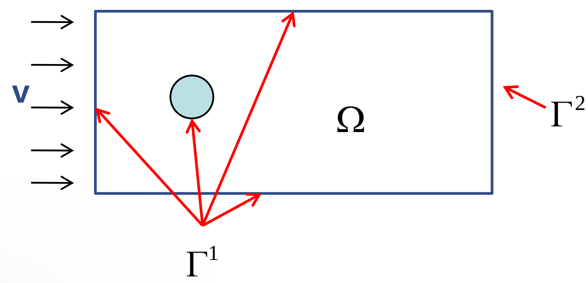
Initial Conditions			
Density	Velocity	Temperature	Turbulence variables
Trivial for constant density flows	Initial mass balance has to be achieved	Required when heat transfer is modeled	Required when turbulence model is chosen

Initial Conditions

- **Boundary conditions**

Velocity $\mathbf{v}(\mathbf{x},t)=\hat{\mathbf{v}}(\mathbf{x},t) \quad \text{on } \Gamma^1$

Traction $-p\mathbf{I}+2\mu\mathbf{S} \cdot \mathbf{n}=\hat{\mathbf{t}} \quad \text{on } \Gamma^2$



- **Initial condition:**

$$\mathbf{v}(\mathbf{x},0)=\mathbf{v}_0(\mathbf{x})$$

- **For a well-posed incompressible flow problem, prescribed initial conditions and boundary conditions must meet "solvability conditions"**
  - If these conditions are met, the solution of the pressure-Poisson equation and momentum equation is equivalent to the solution of the momentum equation along with a divergence constraint :

$$\mathbf{n}\cdot\mathbf{v}_0=\mathbf{n}\cdot\hat{\mathbf{v}} \quad on\ \Gamma^1$$

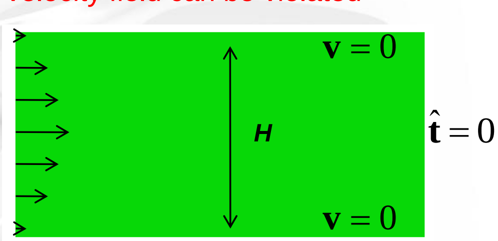
$$\nabla\cdot\mathbf{v}_0=0 \quad on\ \Omega$$

Initial Conditions

- **For enclosed flows (no traction boundaries), mass conservation must be met**

$$\int_\Gamma \mathbf{n}\cdot\mathbf{v}_0 d\Gamma=0$$

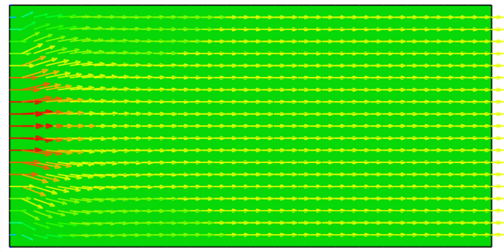
- **Abaqus/CFD tests the prescribed initial and boundary conditions to ensure that the solvability conditions are met**
  - A divergence-free velocity field is obtained from the initial velocity field
    - Pressure-Poisson's equation is solved
  - Based on built-in RMS divergence error tolerance
  - User-defined initial velocity field can be violated



Zero velocity initial conditions

Initial Conditions

```
INITIAL DIVERGENCE SUMMARY
==================================================

Initial divergence .................................... 0.0028868

  -- Initial Projection --
  --- Solution converged ---
       iterations taken  : 16
       ||r ||/||b|| error: 6.8918e-11
       ||dx||/||x|| error: 3.7721e-12

  -- Initial Projection --
  --- Solution converged ---
       iterations taken  : 16
       ||r ||/||b|| error: 6.7597e-10
       ||dx||/||x|| error: 4.177e-09

Projected divergence .................................... 0.0001076
Net volumetric flux balance ............................. -6.7353e-12

  -- Pressure Increment --
  --- Solution converged ---
       iterations taken  : 16
       ||r ||/||b|| error: 2.1492e-11
       ||dx||/||x|| error: 2.9106e-12
```



Velocity field: *t = 0*

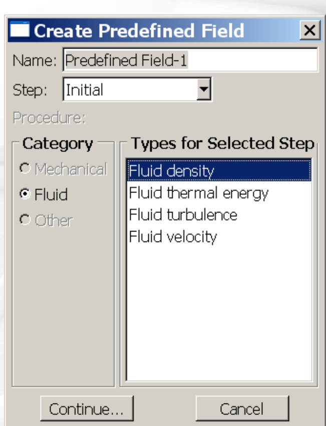
$\|\nabla \cdot \mathbf{v}\|_{rms}$ after div-free projection

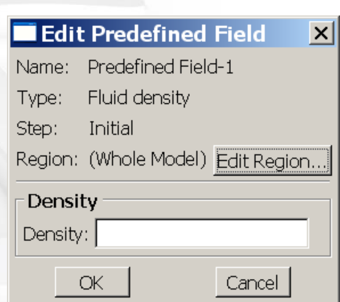
$\|\nabla \cdot \mathbf{v}\|_{rms}$ before projection

Initial Conditions

- **Density**
  - Trivial for constant density flows
    - Initial condition for density is not required for constant density incompressible flows
    - Fluid density from material specification is used

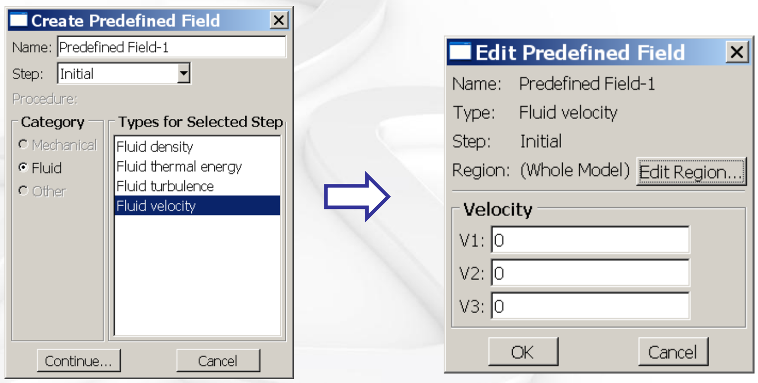
```
***WARNING: No initial density has been prescribed.  The default density from
            the material definition will be used.
```





Initial Conditions

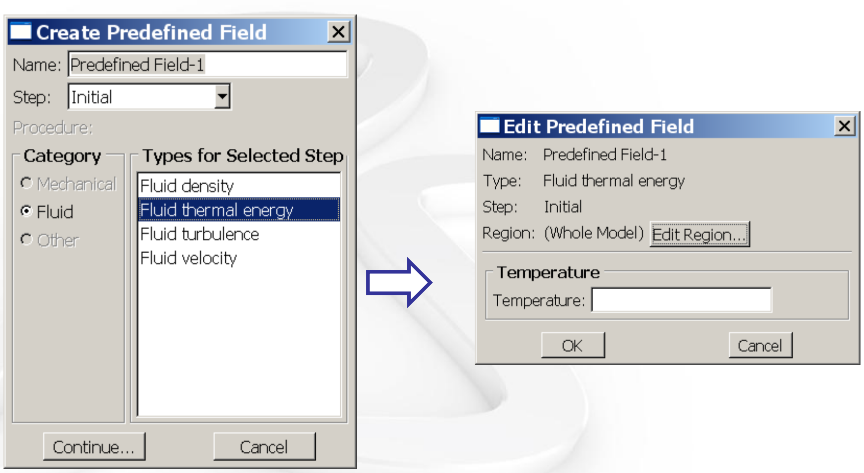
- **Velocity**
  - Initial condition on velocity can be specified
    - By default, the initial velocities are assumed to be zero
  - Initial conditions on velocity may be violated
    - The initial velocities are recomputed during the initialization phase to ensure a well-posed incompressible flow problem



Initial Conditions

- **Temperature**
  - Initial condition on temperature has to be specified if heat transfer is modeled
    - An error is issued if initial temperature is not specified

```
***ERROR: An initial temperature field is required when the energy equation
          is active.
```



Initial Conditions

- **Turbulence variables**
  - Initial condition on turbulence variables need to be specified
    - The variable depends on the choice of turbulence model
    - An error is issued if the initial values of turbulence variables are not specified

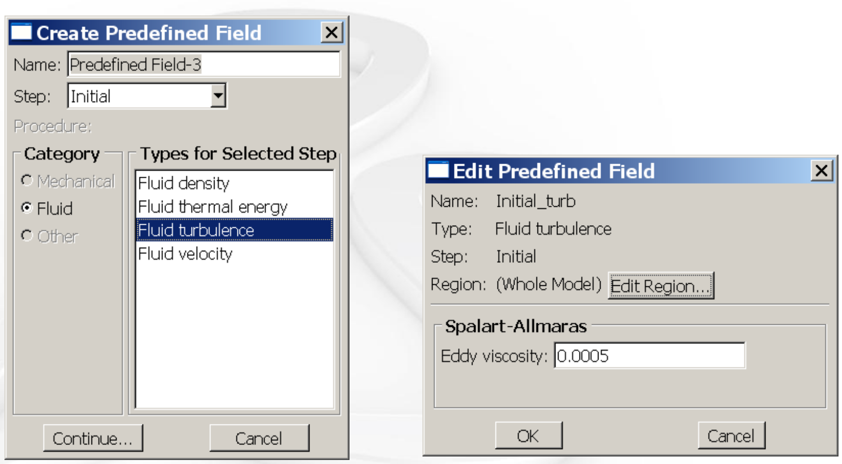
```
***ERROR: An initial turbulent viscosity is required for this turbulence
          model.
```

Turbulence variables required

Turbulence model	Variables
One-equation Spalart-Allmaras	Kinematic turbulent viscosity

Initial Conditions

- **Turbulence variables (cont'd)**
  - The inlet turbulence can be specified as an initial condition
    - The proper specification of inlet turbulence will be addressed when discussing boundary conditions



Boundary Conditions and Loads

SIMULIA

Boundary Conditions and Loads

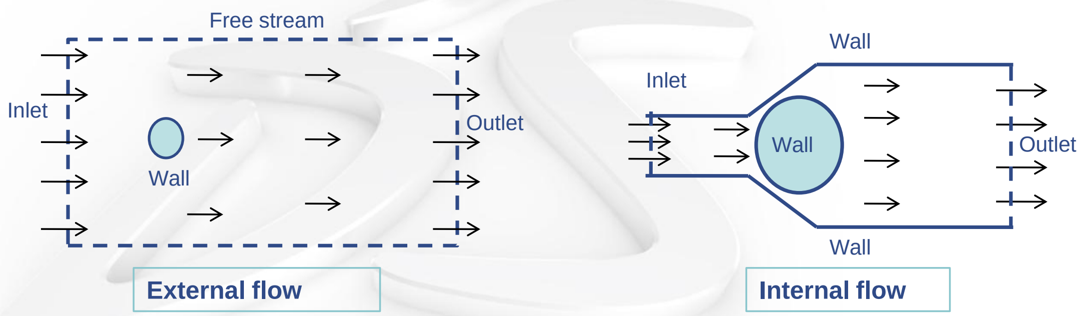
- **Correct specification of the flow and thermal and turbulence conditions on flow boundaries is necessary to accurately model the physics of the flow**
  - Boundary conditions define conditions at flow boundaries

Flow	Thermal	Turbulence	Mesh motion
Pressure	Temperature	Wall-normal distance	Mesh displacement
Velocity	Heat flux	Turbulence viscosity	

SIMULIA

Boundary Conditions and Loads

Where do I need boundary conditions?				
Flow inlet or outlet regions	**Physical wall (stationary or moving) region**	**Far-field region**	**Zones where flow is abstracted**	
Fluid enters or leaves the flow solution domain	Fluid is constrained to “stick” to an obstruction, “No-Slip, no-penetration”	Imaginary flow solution domain boundaries	Approximating three-dimensional flow as two-dimensional	Symmetry conditions



Boundary Conditions and Loads

- **Inlet**
  - Pressure
    - Specify when inlet pressure is known but velocity or mass flow rate is not known
  - Velocity
    - Specify when inlet velocity or mass flow rate is known
    - For incompressible flow, $\dot{m} = \rho \mathbf{A} \cdot \mathbf{v}$
  - If heat transfer is active
    - Specify inlet temperature
  - If turbulence model is active
    - Specify inlet turbulence
      - Variables depend on turbulence model chosen

Turbulence model	Specify
One-equation Spalart-Allmaras	Kinematic turbulence viscosity

Boundary Conditions and Loads

- **Inlet: Specifying inlet turbulence**
  - Spalart-Allmaras turbulence model
    - One equation RANS turbulence model
    - Transport equation for modified kinematic turbulent viscosity, $\tilde{\nu}$
    - Boundary condition for kinematic turbulent viscosity, ν_t

$$\nu_t = \tilde{\nu} f\left(\frac{\tilde{\nu}}{\nu}\right) \quad (m^2/s)$$

- Method 1:

$$\nu_t \approx (3-5)\,\nu$$

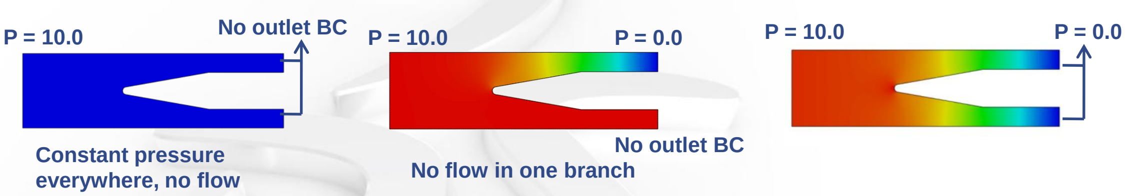
- Method 2:
  - Turbulence intensity and length scale are known

$$\nu_t \approx \sqrt{3/2}\, u_o I l$$

I Turbulence intensity
l Turbulence length scale
u_o Reference velocity (inflow velocity)

Boundary Conditions and Loads

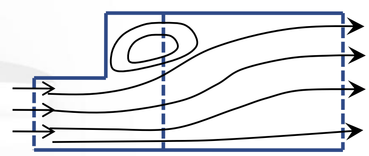
- **Outlet: Case 1**
  - Velocity or pressure is unknown at outlet (outflow boundary condition)
    - Velocity or pressure is part of the CFD solution
    - In this case, no BC needs to be specified
      - Homogeneous traction (natural) boundary condition: do-nothing BC
      - Do not use this approach if any of the inlets has a pressure boundary condition specified – Specify static pressure instead



- Can be used if velocity inlet BC is specified
  - But pressure has to be specified at one node (at least) to eliminate hydrostatic pressure mode

Boundary Conditions and Loads

- **Outlet: Case 2**
  - Static pressure known at outlet
    - Specify static pressure
    - Flow can reverse and actually enter the solution domain
    - Examples: External flows around structures, free surface flows, buoyancy driven flows, internal flows with outlets
  - Location of outlet boundary
    - Outlet BC of type “outflow” should only be located where solution gradients are small
    - Avoid regions of flow reversal
    - Ideally, the interior solution should be unaffected by the choice of location of the outlet

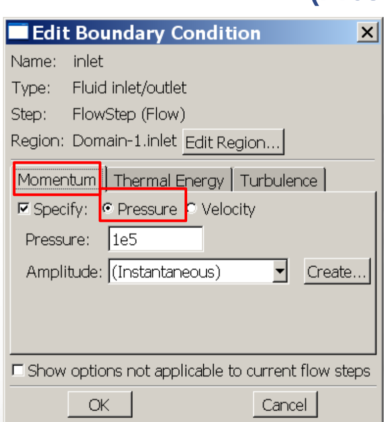


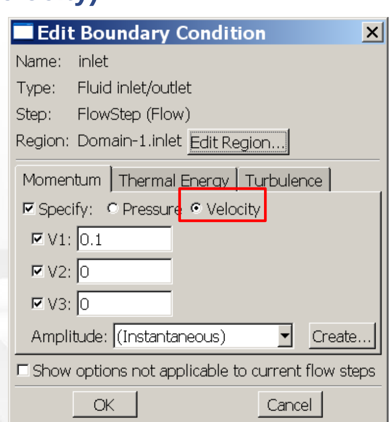
  - Static pressure BCs should not be specified in a re-circulating zone

Boundary Conditions and Loads

- **Specifying inlet/outlet BCs**

Inlet/Outlet Boundary Condition (Pressure/Velocity)



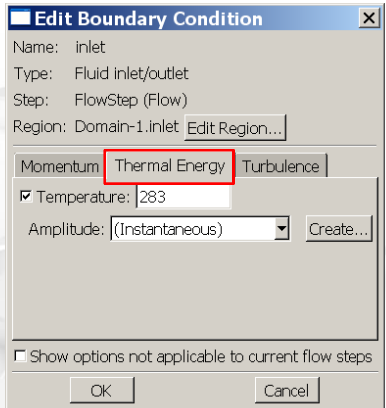


- Time dependence can be specified using amplitude curves

Boundary Conditions and Loads

- **Specifying inlet/outlet BCs (cont'd)**

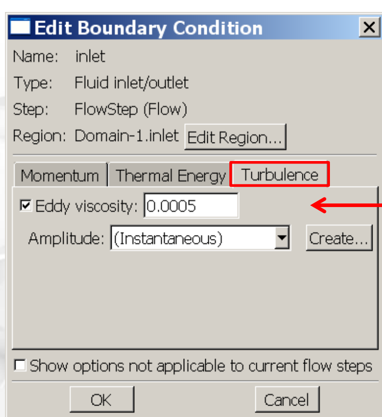
Inlet/Outlet Boundary Condition (Thermal)



Boundary Conditions and Loads

- **Specifying inlet/outlet BCs (cont'd)**

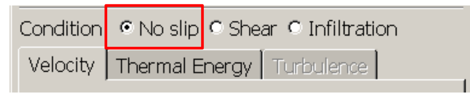
Inlet/Outlet Boundary Condition (Turbulence)

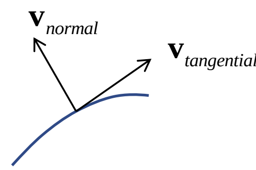


Specify inlet turbulence

Boundary Conditions and Loads

- **Wall: Velocity BC**
  - No-slip/no-penetration wall
    - Surface where the fluid adheres to the wall without penetrating it
    - Prescribed by setting all fluid velocity components equal to the wall velocity (zero if the wall is not moving)
    - No-slip condition is only relevant for viscous flows
      - Not physically relevant for inviscid flows; use only no-penetration condition for inviscid flows

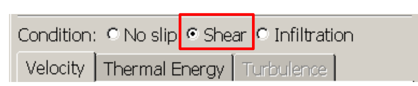


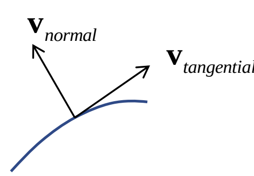


$$\mathbf{v}_{fluid} = \mathbf{v}_{surface}$$

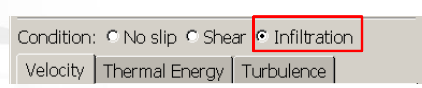
Boundary Conditions and Loads

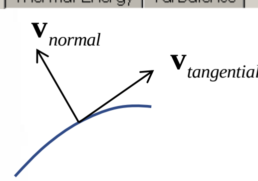
- **Wall: Velocity BC (cont'd)**
  - Slip wall
    - Surface where the fluid does not adhere to the wall but can not penetrate it
    - Prescribed by setting the wall-normal fluid velocity equal to the wall velocity (zero if the wall is not moving)
  - Infiltration wall
    - Permits the fluid to penetrate the surface while maintaining the no-slip condition
    - Prescribed by setting the wall-normal velocity equal to the velocity representing the infiltration velocity, while the wall-tangent fluid velocity is equal to the wall velocity (zero if the wall is not moving)





$$\mathbf{v}_{normal} = \mathbf{v}_{surface}$$



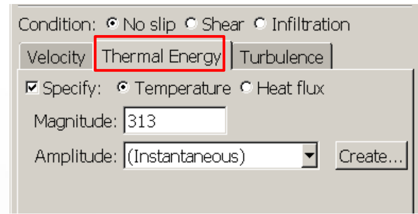


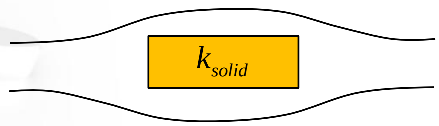
$$\mathbf{v}_{normal} = \mathbf{v}_{infiltration}$$

$$\mathbf{v}_{tangential} = \mathbf{v}_{surface}$$

Boundary Conditions and Loads

- **Wall: Thermal**
  - If heat transfer is active, you can specify:
    - Wall temperature
    - Wall heat flux
  - A wall is typically a part of a solid body
    - Examine the Biot number to determine if modeling heat transfer within the solid is necessary (conjugate heat transfer)
    - Bi << 1:
      - Heat conduction inside the solid body occurs much faster than the heat convection away from its surface; use temperature BC at solid walls
    - Bi >> 1:
      - Need to consider spatial variation of temperature within the solid; include effects of conduction in the solid





$$Biot\ Number\ (Bi) = {hL}/{k_{solid}}$$

Boundary Conditions and Loads

- **Wall: Turbulence**
  - Turbulence models require special boundary conditions at walls
    - Wall-normal distance
    - Boundary conditions on turbulence variable
    - Not required for infiltration walls
      - If fluid penetrates the surface, need to specify inlet turbulence instead

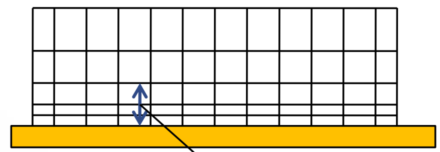
Turbulence specific BCs at walls

Turbulence model	Wall-normal distance	Turbulence variable
One-equation Spalart-Allmaras	Yes	Modified kinematic turbulent viscosity, $\tilde{\nu} = 0$

Boundary Conditions and Loads

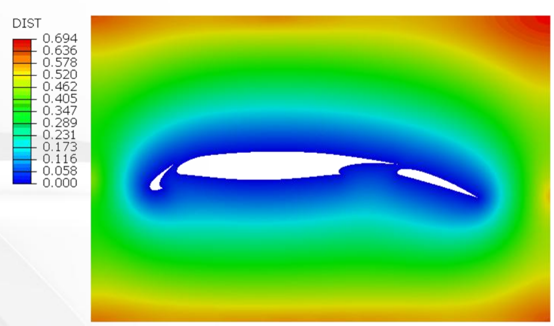
- **Wall: Turbulence (cont'd)**
  - Wall-normal distance
    - Distance from walls
    - Required to enable near-wall modeling
      - Using wall-functions with turbulence models
      - Built-in damping functions for low-Reynolds number turbulence models
    - Calculated by Abaqus/CFD
      - For flow problems, only need to calculate once at start-up
      - If the mesh deforms, the wall-normal distance is frequently calculated and updated; for example:
        - Fluid-structure interaction problems
        - Flow problems with prescribed boundary motion

Additional solution cost



Boundary Conditions and Loads

- **Wall: Turbulence (cont'd)**
  - Wall-normal distance (cont'd)
    - Poisson's equation for distance function is solved
    - Boundary conditions for distance function calculation
      - $d = 0$ at walls (no-slip and shear conditions)
      - Automatically set by Abaqus/CAE at surfaces with no-slip and shear wall conditions
      - Automatically set by Abaqus/CFD for FSI problems (where wall conditions need not be defined on FSI surface)



3 element aerofoil

- Visualize by requesting output variable DIST

Boundary Conditions and Loads

- **Wall: Turbulence (cont'd)**
  - Wall-normal distance (cont'd)

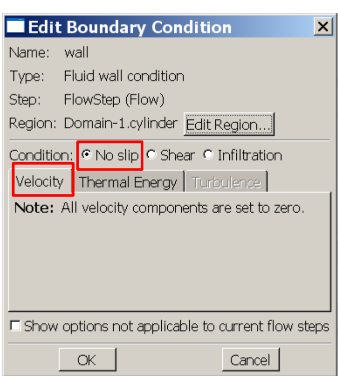
```
1   1.09e-03   1.09e-03  1.68e-01   1.78e+00  -7.89e-01   7.89e-01  -9.69e-04   4.71e-04   2.83e+02   2.89e+02

   -- Distance Function --
   --- Solution converged ---
        iterations taken   : 16
        ||r ||/||b|| error: 1.253e-06
        ||dx||/||x|| error: 3.052e-08

   -- X-Momentum --
   --- Solution converged ---
        iterations taken   : 6
        ||r ||/||b|| error: 1.713e-08
        ||dx||/||x|| error: 4.986e-06

   -- Y-Momentum --
   --- Solution converged ---
        iterations taken   : 6
        ||r ||/||b|| error: 6.386e-07
        ||dx||/||x|| error: 6.572e-06

   -- Z-Momentum --
   --- Solution converged ---
        iterations taken   : 8
        ||r ||/||b|| error: 1.597e-07
        ||dx||/||x|| error: 1.007e-06

   -- S-A Transport Model Equation --
   --- Solution converged ---
        iterations taken   : 10
        ||r ||/||b|| error: 8.463e-08
        ||dx||/||x|| error: 7.324e-06
```

- Turbulent flow problems with mesh deformation incur additional solution costs associated with frequent updating of wall-normal distances.

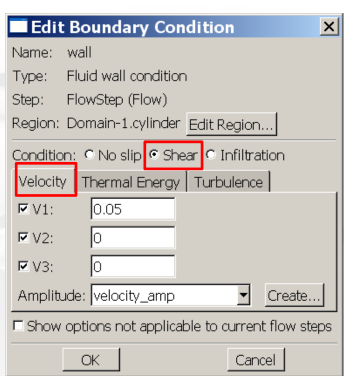
Boundary Conditions and Loads

- **Specifying wall BCs**

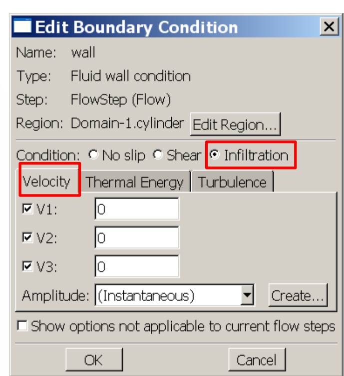
Wall Boundary Condition (Velocity)



No-slip/no-penetration wall condition



Slip wall condition

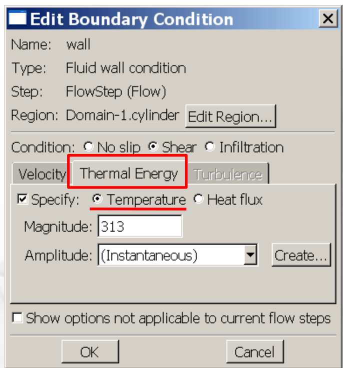


Infiltration wall condition

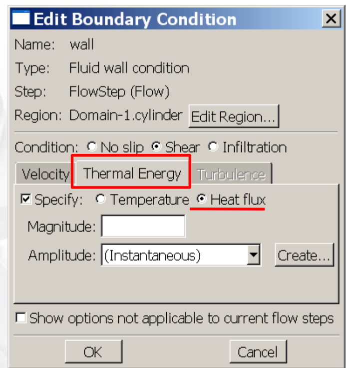
Boundary Conditions and Loads

- **Specifying wall BCs (cont'd)**

Wall Boundary Condition (Thermal)



Wall temperature



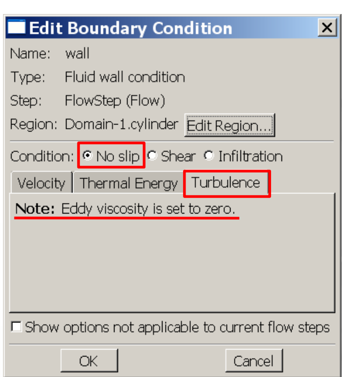
Wall heat flux

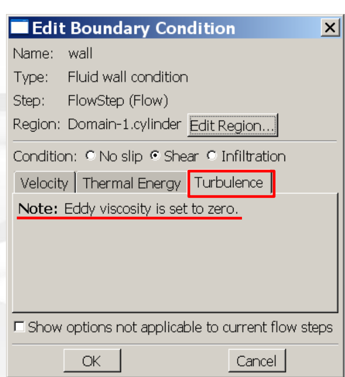
- The thermal energy tab is active only if heat transfer is selected on the incompressible flow step

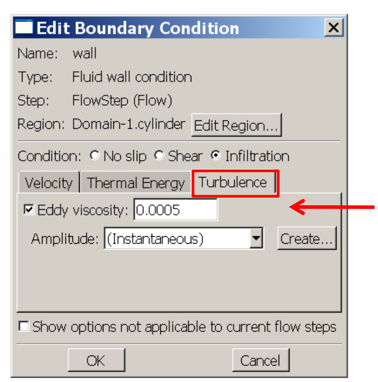
Boundary Conditions and Loads

- **Specifying wall BCs (cont'd)**

Wall Boundary Condition (Turbulence)





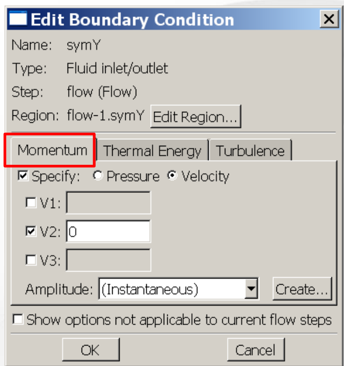


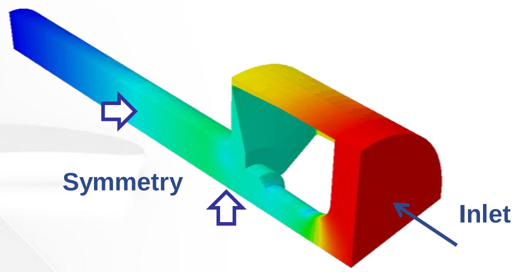
- Turbulence BCs automatically set for no-slip and shear wall conditions

- Must specify inlet turbulence for infiltrating fluid

Boundary Conditions and Loads

- **Symmetry**
  - Modeling abstraction to reduce the computational model size
    - Normal component of the velocity is zero
    - Gradient of other quantities along the normal direction is zero

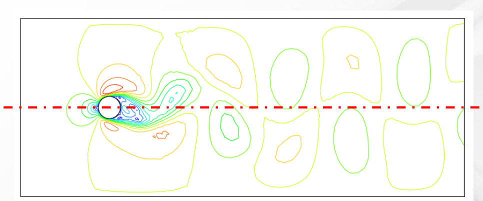




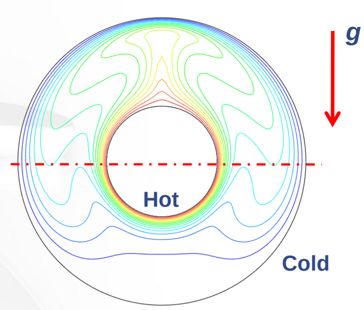
- Use inlet/outlet BC for specifying velocity condition

Boundary Conditions and Loads

- **Tips**
  - Geometric symmetry need not imply symmetric flow patterns
  - Flow asymmetry can be introduced due to
    - Flow conditions (Example: Flow Reynolds number)
    - Physics of the flow (Example: Instabilities, Bifurcations etc.)
    - Body forces



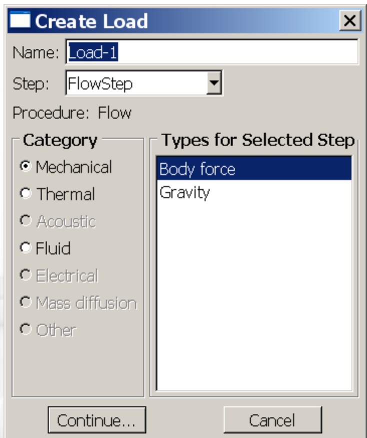
Velocity:
Vortex shedding at *Re = 100*

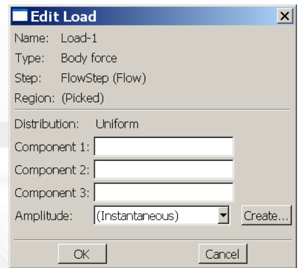


Temperature:
Convection between concentric cylinders at *Ra = 48000*

Boundary Conditions and Loads

- **Body forces**
  - Specified as loads
    - E.g., body force due to rotation

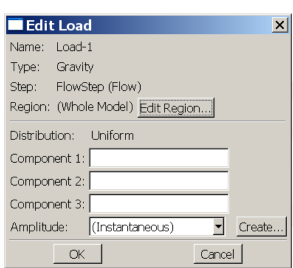


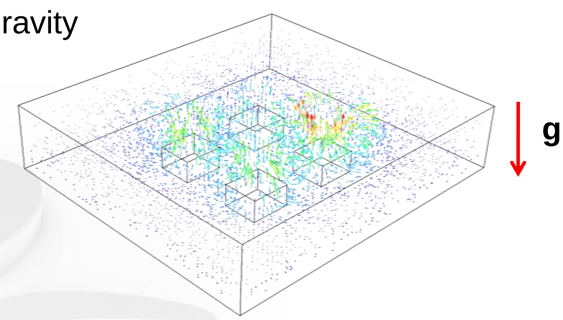


- Specify body force *per unit volume*

Boundary Conditions and Loads

- **Gravity**
  - Natural convection using Boussinesq body forces
  - Specify acceleration due to gravity



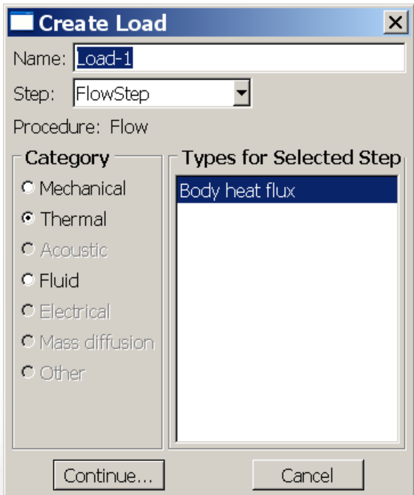


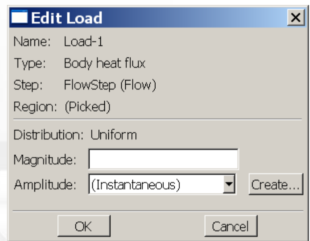
- **When modeling natural convection, the expansion coefficient of the fluid must be specified to ensure thermal-momentum coupling**
  - A warning is issued if the expansion coefficient is not specified

```
Warning: Buoyancy driven flow will not occur in this analysis because the thermal
expansion coefficient of the fluid has not been specified.  Buoyancy forces will
be zero as a result.
```

Boundary Conditions and Loads

- **Volumetric heat sources**
  - Specify volumetric heating sources in the fluid

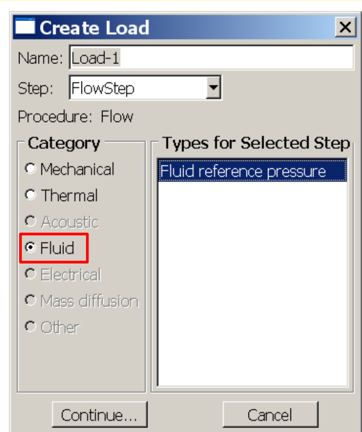


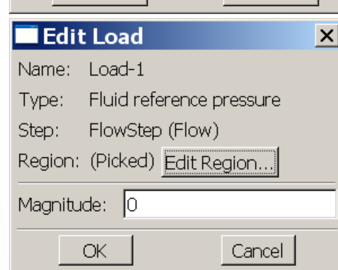


- Specify body heat flux *per unit volume*

Boundary Conditions and Loads

- **Reference pressure**
  - In incompressible flows, the pressure is only known to within an arbitrary additive constant (the hydrostatic pressure)
    - Specifying pressure at an outflow boundary sets the hydrostatic pressure level
    - If no pressure BC is prescribed, it is necessary to set the hydrostatic pressure level at one node (at least) in the mesh
      - Ensures non-singularity of the pressure equation
  - If pressure boundary conditions are prescribed in addition to the reference pressure level, the reference pressure simply adjusts the output pressures according to the specified pressure level
  - Disconnected regions: Each requires its own hydrostatic pressure level to be set





Output

Output

Field output	History output
• Available at nodes • Honors the applied boundary conditions	• Available at element centers • May seem to violate the applied boundary conditions since these quantities are from element center

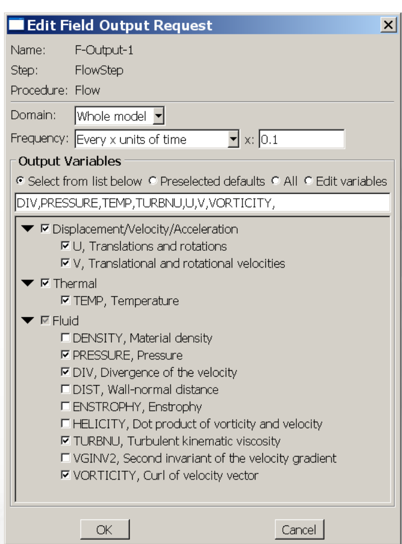
- Output is available at
  - Every *n* increments (default is 1)
  - Evenly spaced time intervals (default is 20)
  - Every *x* units of time
- Output is available at approximate times
  - Analysis does not cut back to output at exact times
- Preselected default output is available
- Output variables that are not relevant to an analysis are ignored by Abaqus/CFD

Output

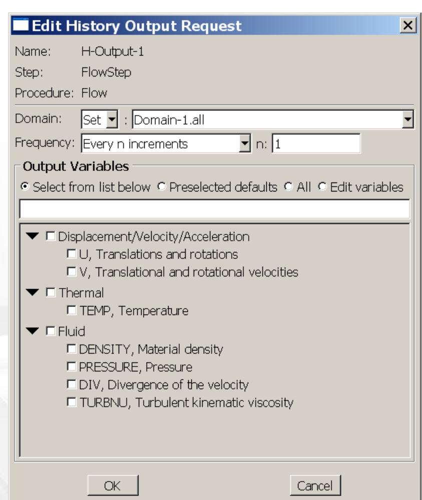
	Field Output	History Output
Flow	Pressure Velocity Density Helicity Vorticity Enstrophy	Pressure Velocity Density
Thermal	Temperature	Temperature
Divergence of Velocity	Yes	Yes
Turbulence	Kinematic turbulence viscosity Wall-normal distance	Kinematic turbulence viscosity
Mesh displacement	Yes	No

Output

Field output



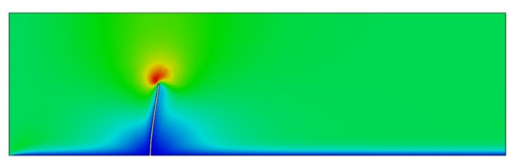
History output



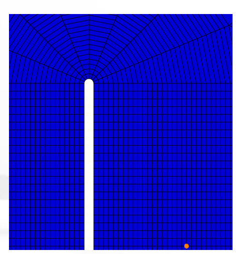
Deforming Mesh

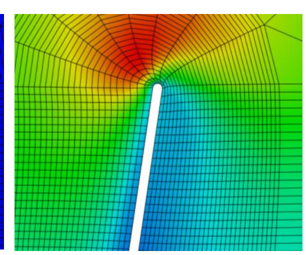
Deforming Mesh

- Mesh deformation is required for problems involving moving boundaries
  - Prescribed boundary motion
  - Fluid-structure interface motion due to structural deformation



- Arbitrary Lagrangian-Eulerian (ALE) method
  - Hyperfoam material model
  - Material parameters are automatically determined

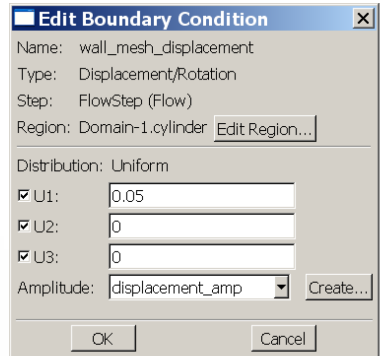




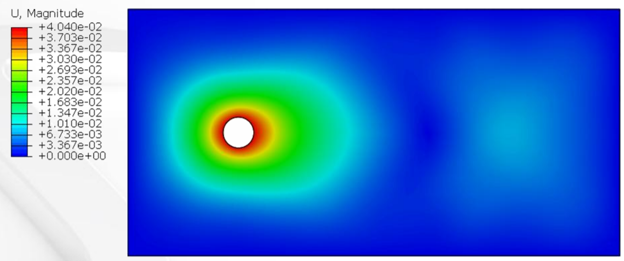
- Preserves boundary layer mesh
- Automatically activated for problems that involve moving boundaries
  - No user control available

Deforming Mesh

- Boundary conditions need to be specified for mesh motions
  - Displacement condition on mesh nodes
  - Mesh needs to be appropriately constrained to prevent rigid-body motion
  - For a symmetry face, appropriate symmetry conditions on mesh displacement need to be applied



- For boundary motions caused by structural motion/deformation (modeled as part of a structural analysis), no mesh displacement boundary conditions are needed
  - Dictated by the FSI coupling



- Visualize by requesting mesh displacement output variable, U

Monitoring Convergence

Monitoring Convergence

- For incompressible flows, continuity equation has to be satisfied
  - Measured by divergence (RMS)

$$\left\|\nabla \cdot \mathbf{v}\right\|_{RMS} = \sqrt{\frac{\sum_N \left(\nabla \cdot \mathbf{v}\right)^2}{N}}$$ Divergence (RMS)

$$\varepsilon_{volume} = \left\|\nabla \cdot \mathbf{v}\right\|_{RMS} \Delta t$$ Volume error

Status (`.sta`) file

```
CFD Incompressible Flow
 Step   Inc    DT                Time             RMS Div.          KE
 ------ ------ ---------- -------------- -------------- --------------
      1      0  0.01000     0.000000       4.87625e-07     0.129588
      1      1  0.01025     0.0102500      6.03596e-07     0.129617
      1      2  0.01051     0.0207562      6.43673e-07     0.129641
```

- Smaller values indicate better mass balance
  - Should be < 1e–3

Notes

Notes

Getting Started with FSI using Abaqus/CFD

Lecture 6

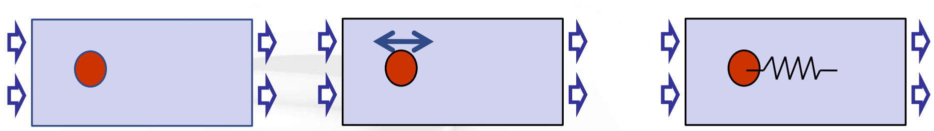
Overview

- Setting up FSI Analyses
- Case Study 3: Flow around a Spring-loaded Rigid Circular Cylinder
- Conjugate Heat Transfer Analyses

Setting up FSI Analyses

Setting up FSI Analyses

- **Case study introduction**
  1. Flow around a rigid circular cylinder
  2. Flow around an oscillating rigid circular cylinder
  3. Flow around a spring-loaded rigid circular cylinder

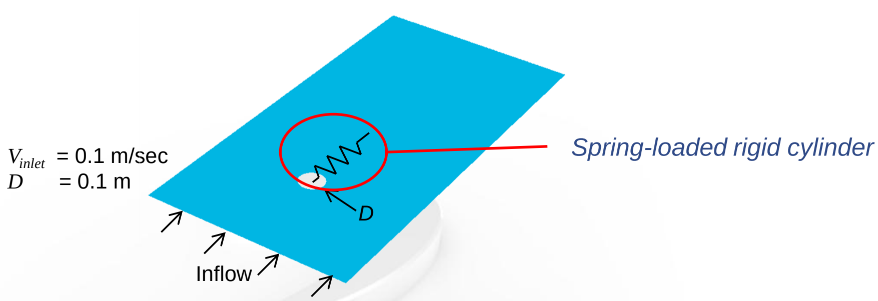


Problem description	Flow around a rigid circular cylinder	Flow around an oscillating rigid circular cylinder	Flow around a spring-loaded rigid circular cylinder
Flow domain	Around the cylinder	Around the cylinder but domain *changes* due to cylinder's oscillation	Around the cylinder but domain *changes* due to cylinder's oscillation
How do I model it?	1) Model fluid flow 2) Mesh is fixed	1) Model fluid flow 2) Allow mesh at cylinder surface to accommodate displacements (ALE)	1) Model fluid flow 2) Allow mesh at cylinder surface to accommodate displacements (ALE) 3) Model the cylinder and the spring in structural solver (co-simulation)
Cylinder motion	None	Modeled in Abaqus/CFD as a boundary condition	Determined by structural analysis (two separate models)

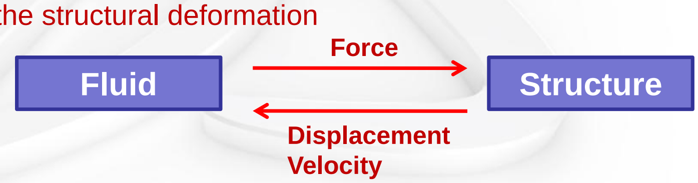
Case Study 3: Flow around a Spring-loaded Rigid Circular Cylinder

Introduction

- Consider the case of flow around a spring-loaded rigid circular cylinder
  - Flow at Reynolds number = 100

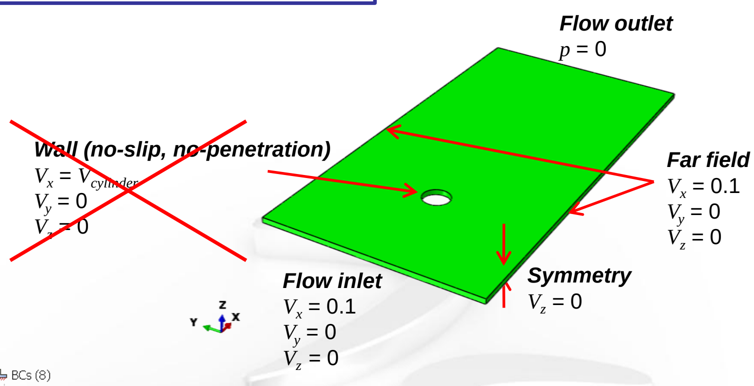


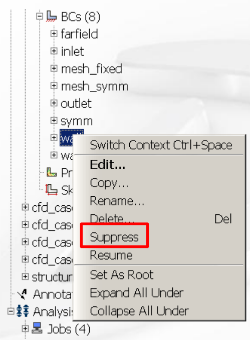
- Boundary motion due to structural deformation
  - The structural deformation and fluid velocities are governed by the coupled physics
  - Boundary conditions on the mesh displacements and fluid velocities are dictated by the structural deformation



Defining the CFD Model

1. Define boundary conditions





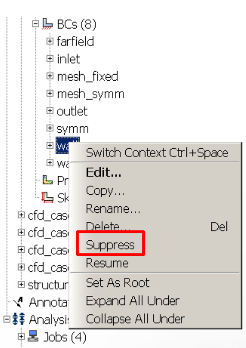
- ❖ *Cylinder surface still requires no-slip, no-penetration boundary condition but this boundary condition is now dictated by the FSI coupling*
- ❖ *Define fluid-structure interaction instead*

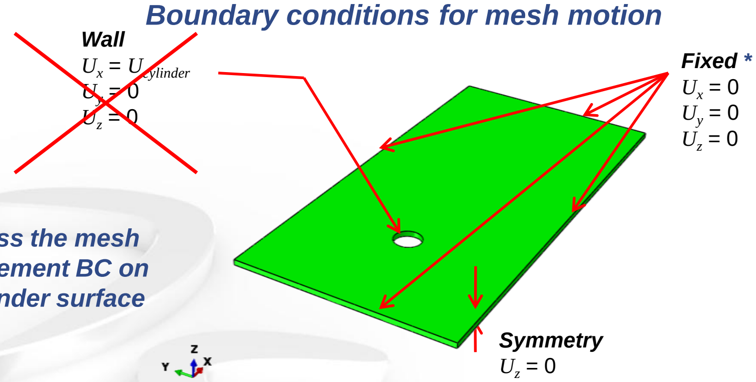
Suppress the wall BC on the cylinder surface

Defining the CFD Model

1. Define boundary conditions (cont'd)

Boundary conditions for mesh motion





Suppress the mesh displacement BC on the cylinder surface

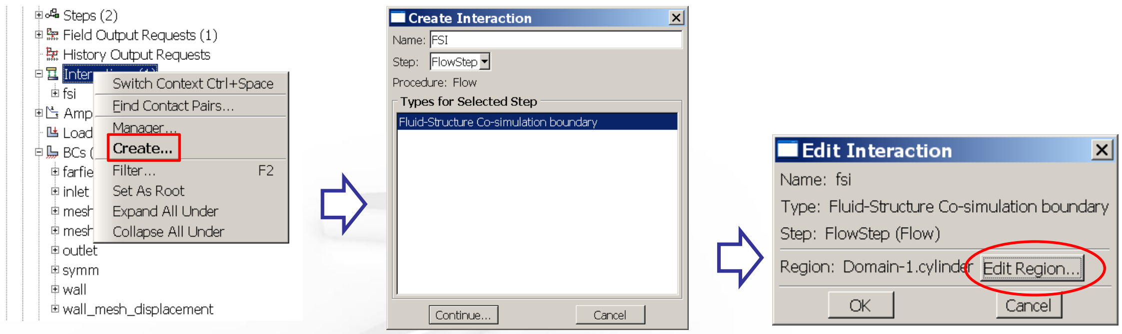
- ❖ *Cylinder surface still requires mesh displacement boundary condition but this boundary condition is now dictated by FSI coupling*
- ❖ *Define fluid-structure interaction instead*

---

* As an alternative, the far-field and outlet boundary conditions on the mesh can be left unspecified leaving the mesh to slip (tow-tank condition)

Defining the CFD Model

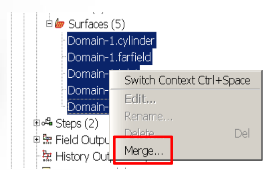
2. Define fluid-structure interaction



Define fluid-interaction surface

- The interaction surface needs to be defined
  - Only one surface per co-simulation definition*
- Only one FSI co-simulation definition per analysis step

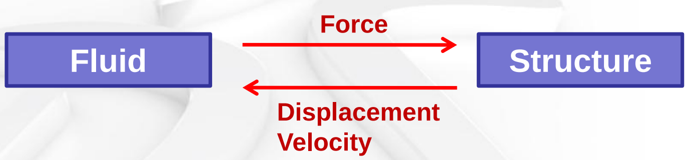
* If necessary, multiple surfaces can be merged into a single surface



Defining the CFD Model

2. Define fluid-structure interaction (cont'd)

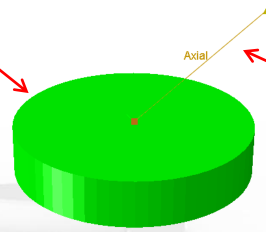
		Abaqus/Standard + Abaqus/CFD		Abaqus/Explicit + Abaqus/CFD	
	Option	Abaqus/ Standard	Abaqus/ CFD	Abaqus/ Explicit	Abaqus/ CFD
Fluid-structure interaction	Export	Displacement Velocity	Forces	Displacement Velocity	Forces
	Import	Forces	Displacement Velocity	Forces	Displacement Velocity



❖ The displacements and velocities that are being imported into Abaqus/CFD from Abaqus/Standard or Abaqus/Explicit serve as the necessary boundary conditions at the FSI interface

Defining the Structural Model

Rigid cylinder
• The cylinder is constrained to move in the axial flow direction



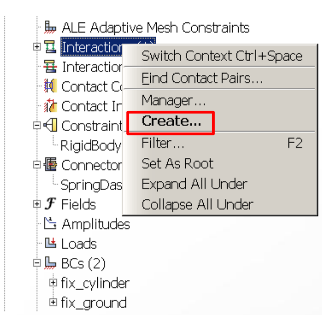
Fixed at ground
(**U** = 0)

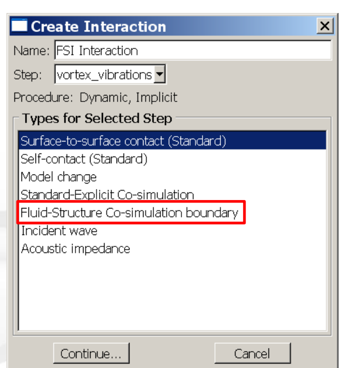
Axial Connector definition
• Defines a spring with linear stiffness, *K* = 1 N/m

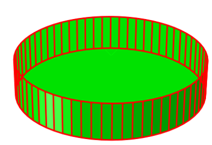
- The FSI interface in the structural and CFD models should be co-located

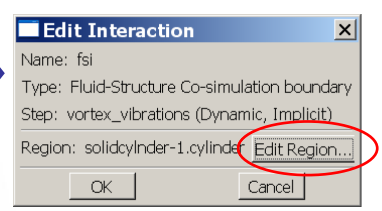
Defining the Structural Model

- **Define fluid-structure interaction**







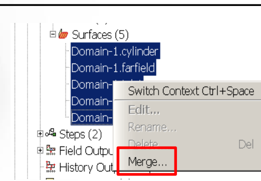


Define fluid-interaction surface

- The interaction surface needs to be defined
  - Only one surface per co-simulation definition*
- Only one FSI co-simulation definition per analysis step can be defined

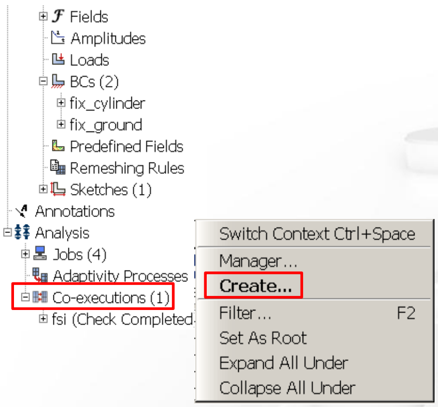
---

* If necessary, multiple surfaces can be merged into a single surface

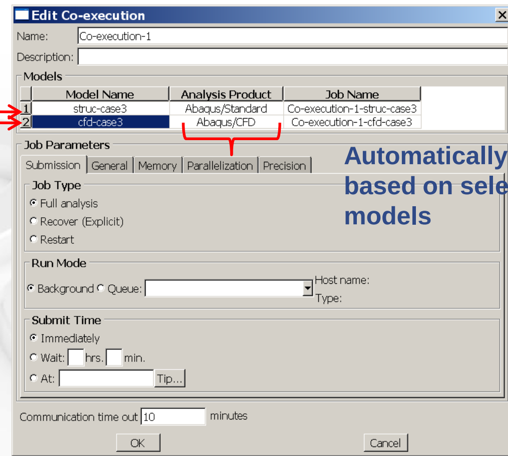


FSI Analysis Execution

- Coupled fluid-structure interaction jobs can be set up and run interactively using the co-execution framework in Abaqus/CAE



Creating co-execution jobs



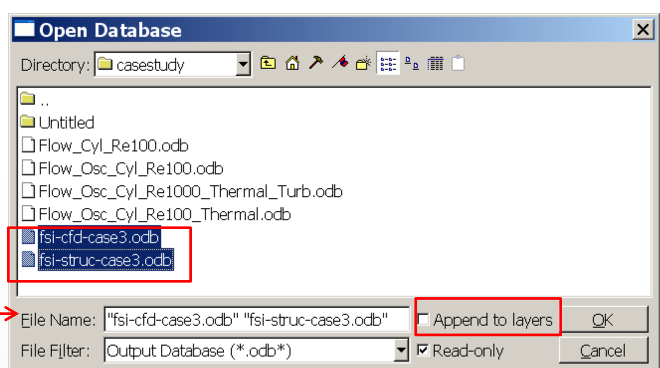
Automatically chosen based on selected models

Define structural job
Define CFD job

SIMULIA

Postprocessing FSI Analyses

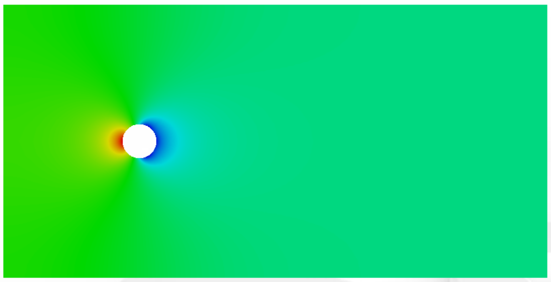
- Open the structural and CFD output databases simultaneously



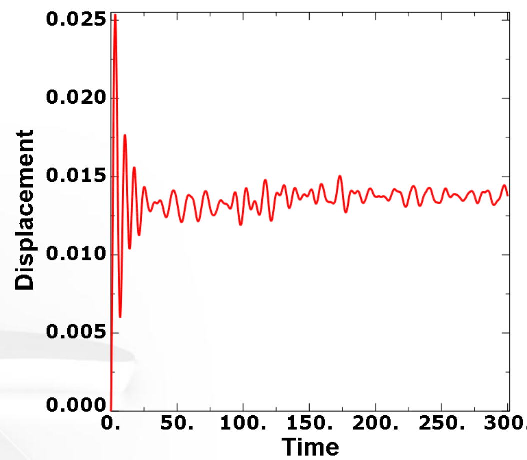
- Can also overlay viewports from multiple output databases

SIMULIA

Postprocessing FSI Analyses



Pressure contours



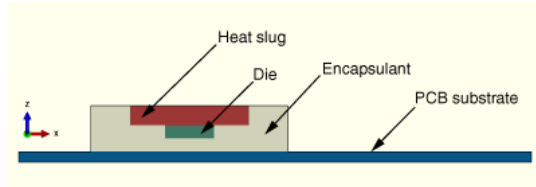
Cylinder displacement

Conjugate Heat Transfer

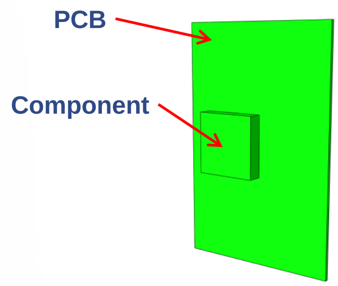
Introduction

- **Model heat transfer within a solid region that interacts with the surrounding fluid**

- **We will show a simple example to demonstrate the basic concepts involved**
  - Transient conjugate heat transfer between a printed circuit board (PCB)-mounted electronic component and ambient air
  - Specified power dissipation within the component

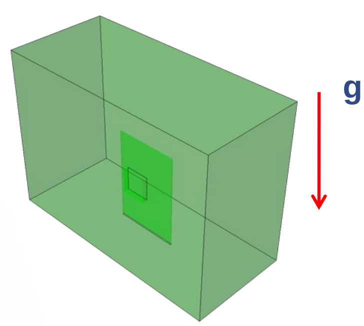


- **Heat transfer mechanism involved**
  - Heat transfer within the component and the PCB due to conduction – **Modeled in Abaqus/Standard**
  - Heated surface of the PCB/Component induces a temperature-dependent density differential in the surrounding air
    - Buoyancy-driven natural convection is set up
    - **Modeled in Abaqus/CFD**



Defining the CFD Model

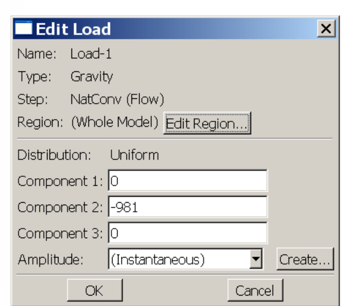
- **CFD mesh is built around the PCB/Component**
- **Material properties for air**
  - Density: 1.127 Kg/m^3
  - Viscosity: 1.983x10^{-5} kg/m/s
  - Thermal conductivity: 2.71x10^{-2} W/m/K
  - Specific heat (C_p): 1006.4 J/Kg/K
  - Thermal expansion coefficient: 3.43x10^{-3} /K



- **Thermal expansion property for air has been specified to enable coupling between momentum and energy equations – Natural convection**

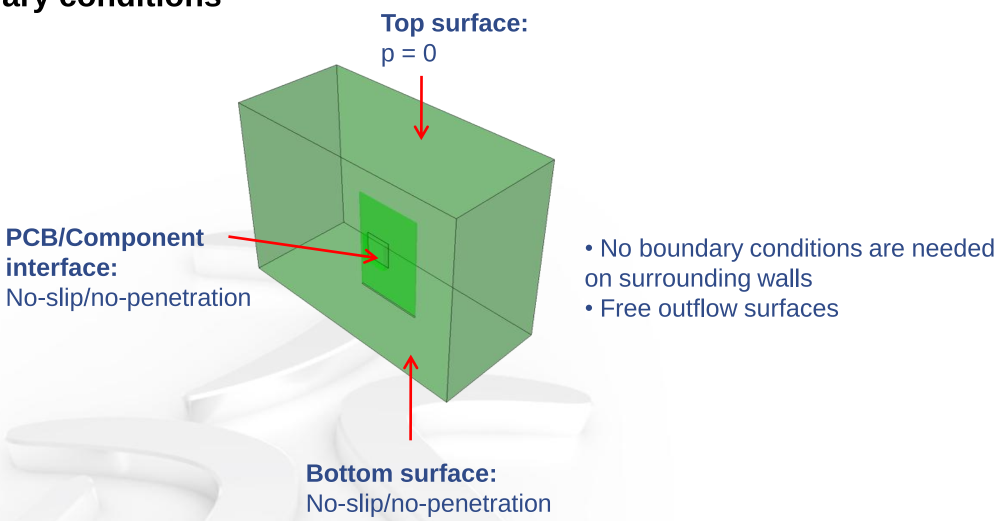
- **Define thermal initial condition**
  - Initial temperature of air = 293 K

- **Define gravity load**



Defining the CFD Model

- **Boundary conditions**



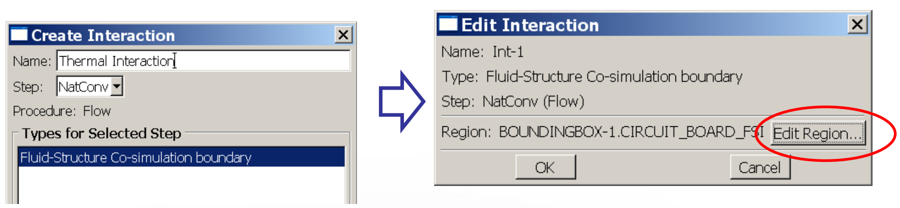
❖ **PCB/Component surfaces require thermal boundary conditions**
  - ❖ **Should not be specified in CFD model**
  - ❖ **Dictated by thermal coupling with the solid**
  - ❖ **Create co-simulation interaction instead**

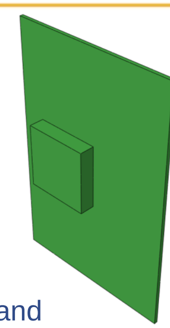
Fluid → Heat flux → **Structure**; **Structure** → Temperature → **Fluid**

SIMULIA

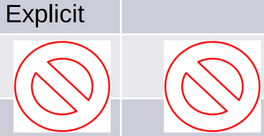
Defining the CFD Model

- **Define thermal interaction**





- The co-simulation definition enables thermal coupling between the fluid and the structure

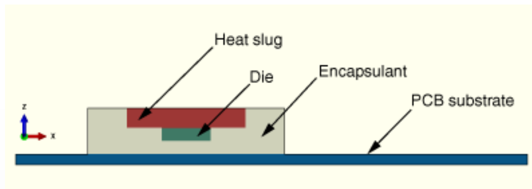
		Abaqus/Standard + Abaqus/CFD		Abaqus/Explicit + Abaqus/CFD	
Coupling type	Option	Abaqus/ Standard	Abaqus/CFD	Abaqus/ Explicit	Abaqus/CFD
Conjugate heat transfer	Export	Temperature	Heat flux		
	Import	Heat flux	Temperature		

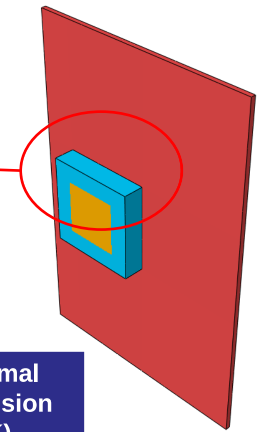
Fluid → Heat flux → **Structure**; **Structure** → Temperature → **Fluid**

SIMULIA

Defining the Structural Model

- **Printed circuit board (PCB)-mounted electronic component**
  - Heat transfer step in Abaqus/Standard

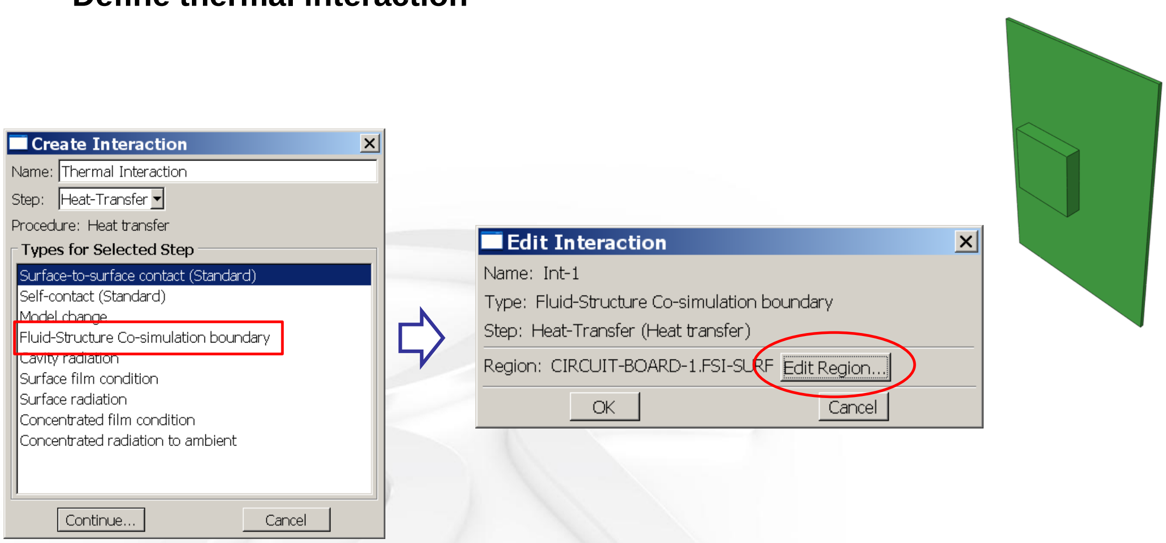




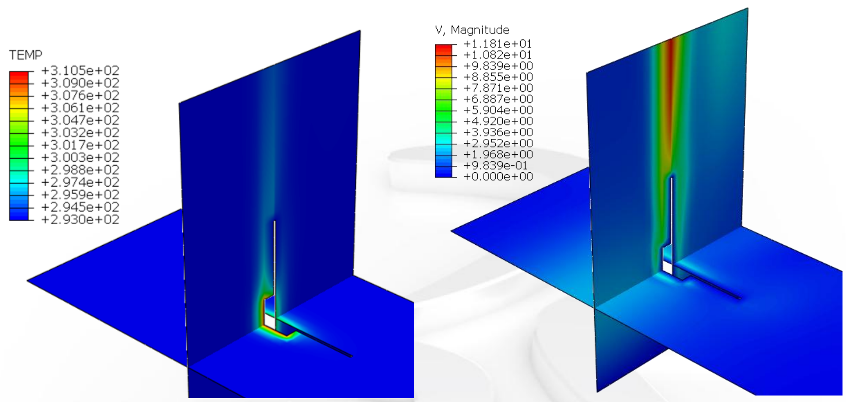
	Density (Kg/m^3)	Specific heat (J/Kg/K)	Thermal conductivity (W/m/K)	Thermal expansion (/K)
PCB substrate	8950	1300	19.25	1.6×10^{-5}
Encapsulant	1820	882	0.63	1.9×10^{-5}
Die	2330	712	130.1	3.3×10^{-6}
Heat slug	8940	385	398	3.3×10^{-6}

Defining the Structural Model

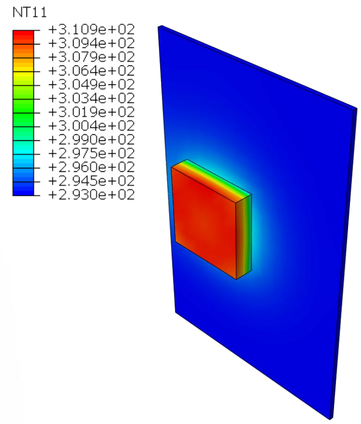
- **Define thermal interaction**



Results



Surrounding air:
Temperature and velocity contours



PCB/Component:
Temperature contours

Notes

Notes

FSI Modeling Techniques

Lecture 7

Overview

- FSI Analysis Workflow
- FSI Analysis Attributes
- Conjugate Heat Transfer

FSI Analysis Workflow

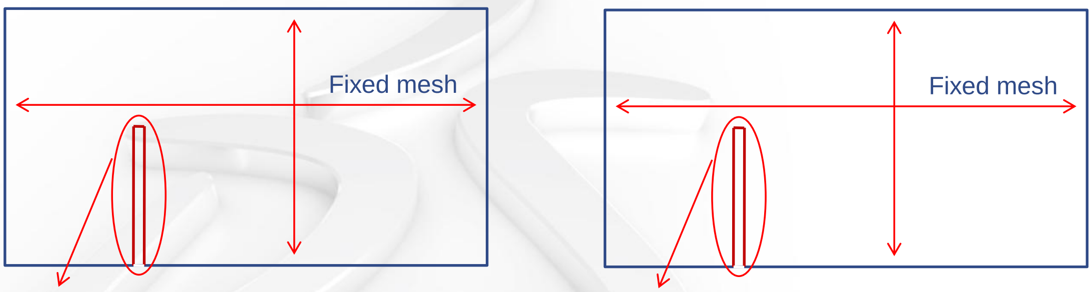
FSI Analysis Workflow

- **Develop the structural model (Abaqus/Standard or Abaqus/Explicit)**
  - Identify the fluid-structure interfaces
    - Co-locate the interface boundary between the fluid and the structural domains
  - Verify the structural model using "assumed" pressure/heat flux loads at the interface
    - Apply pressure/heat flux load magnitudes that are reasonable and similar to the expected fluid loads
- **Develop the CFD model (Abaqus/CFD)**
  - Define the fluid-structure interface wall boundary
    - Co-locate the interface boundary between the fluid and the structural domains
  - Verify your CFD-only analysis by prescribing temperatures at the interface wall

FSI Analysis Workflow

- **Develop the CFD model (cont'd)**
  - Verify your CFD-only analysis by moving the interface wall
    - Modeling mesh motion in Abaqus/CFD requires correct specification of boundary conditions on nodes
    - Ensure that boundary conditions on the mesh motion required at non-FSI interfaces are correctly defined

Flow around a baffle



Prescribed "test" displacement

Define FSI interaction (BC dictated by structural solver)

CFD-only **FSI**

FSI Analysis Workflow

- **Interconnect the structural and CFD models for the co-simulation**
  - Delete the "assumed loads"
  - Define the fluid-structure interaction and the exchange variables

- **Run the FSI analysis**
  - Create co-execution jobs

- **Postprocess the structural and CFD solution**

It is unlikely that the coupled analysis will be successful if the individual structural and CFD analyses are incorrectly set up!

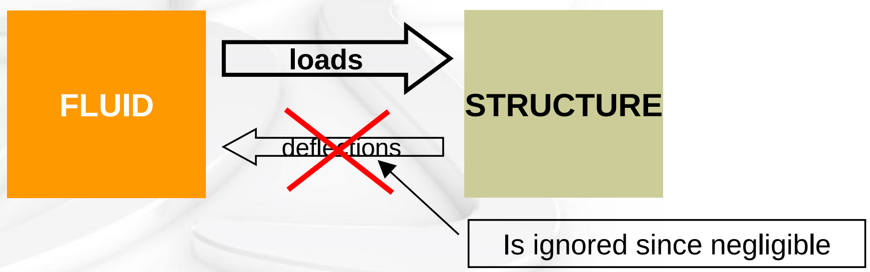
FSI Analysis Attributes

FSI Analysis Attributes

- **Coupling type**: Uni-directional; Bi-directional
- **Coupling schemes**: Algorithm
- **Rendezvousing**: Coupling step size; Time incrementation
- **Coupling strength**: Weak coupling; Strong coupling

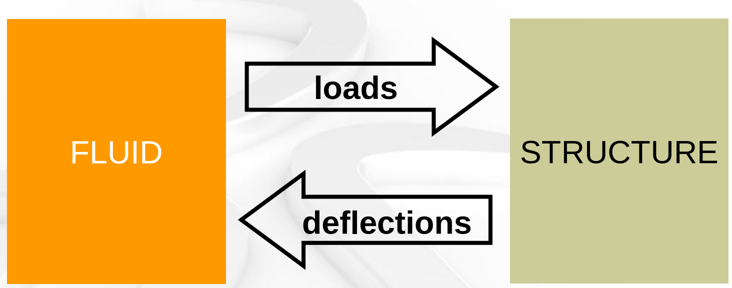
FSI Analysis Attributes

- **Coupling type**
  - Unidirectional coupled analysis
    - Coupling strength in one direction may be so small as to be negligible
      - Common with mechanical structural response influence on fluid
      - Enables a sequential "one-way" analysis: fluid, then structure
    - For transient analysis, need to perform one-way coupling at designated time level



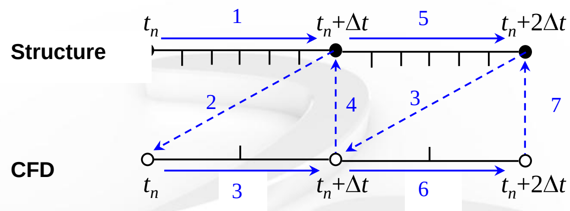
FSI Analysis Attributes

- **Coupling type**
  - Bidirectional coupled analysis
    - The fluid and structural fields affect each other
      - The solution needs to be computed in a coupled manner



FSI Analysis Attributes

- **Coupling scheme**
  - Gauss-Seidel (serial) coupling scheme
  - One solver waits while the other solver proceeds



  - Abaqus/Standard or Explicit leads the simulation
  - Abaqus/CFD lags
  - Automatically set – No user control

FSI Analysis Attributes

- **Rendezvousing**
  - Coupling step size is determined automatically
  - Two methods to determine the coupling step size
    - Min/Min
      - Minimum coupling step size based on the suggested coupling step size of structural and CFD models
    - Import/Export
      - Structural model imports the time step size from CFD model
  - The coupling step size is always reached exactly in both the structural and CFD analyses

FSI Analysis Attributes

- **Rendezvousing**
  - Time incrementation strategy
    - Two methods available
      - Subcycling
        - Model takes one or more increments to reach the next coupling time
      - Lockstep
        - Model takes only one increment to reach the next coupling time
    - The CFD model always moves ahead to the next coupling time in a lockstep fashion
    - The structural model (Abaqus/Standard or Abaqus/Explicit) will either subcycle or lockstep

FSI Analysis Attributes

		Abaqus/Standard + Abaqus/CFD		Abaqus/Explicit + Abaqus/CFD	
Coupling	**Option**	**Abaqus/ Standard**	**Abaqus/ CFD**	**Abaqus/ Explicit**	**Abaqus/ CFD**
Fluid-structure interaction	Step size	MIN	MIN	IMPORT	EXPORT
	Time Incrementation	Subcycle	Lockstep	Subcycle	Lockstep
Conjugate heat transfer	Step size	MIN	MIN		
	Time Incrementation	Subcycle	Lockstep		

FSI Analysis Attributes

- **Coupling strength**
  - The FSI technology is based on a sequentially staggered methodology
  - The native FSI capability in Abaqus addresses weak to moderately coupled FSI problems
    - For problems where "added-mass" effects are important, this approach may lead to numerical instabilities
      - Occurs when fluid density is close to the density of the structure
  - Added mass effect
    - The fluid acts as an extra mass on the structural degrees of freedom at the coupling interface
      - Ignoring added mass effect can cause numerical instability
      - Limit density ratios $\rho_s / \rho_f >> 1$

FSI Analysis Attributes

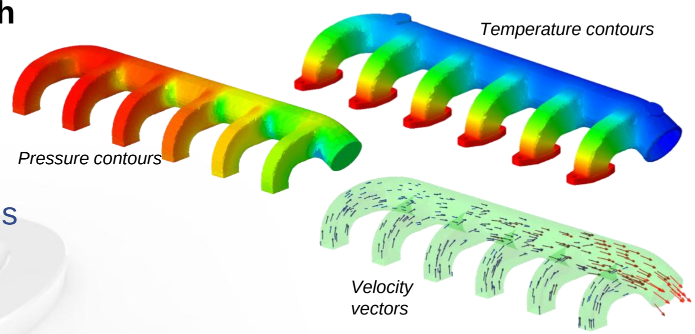
- **"Symptoms" of instability**
  - The bigger the density ratio, the worse the instability gets
  - Rapidly moving interface (high accelerations)
  - With decreasing Δt, the instability occurs earlier
    - Uncharacteristic behavior for explicit coupling
  - Increased fluid viscosity increases the instability while increased structural stiffness offers a decreasing effect

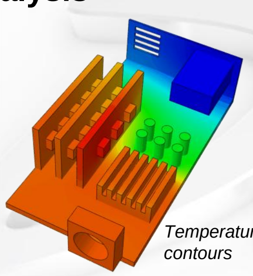
Conjugate Heat Transfer

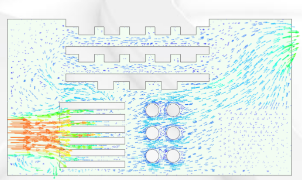
Conjugate Heat Transfer

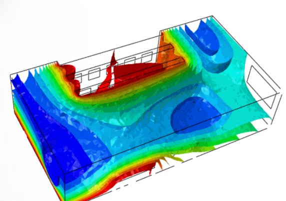
- **Model heat transfer within a solid region that interacts with a surrounding fluid**
  - Examples:
    - Engine manifold
    - Electronic circuit boards

- **Can be followed by a sequential thermal-stress analysis**



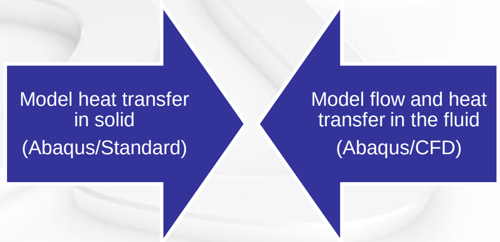

Temperature contours


Velocity vectors on intermediate plane


Temperature isosurfaces

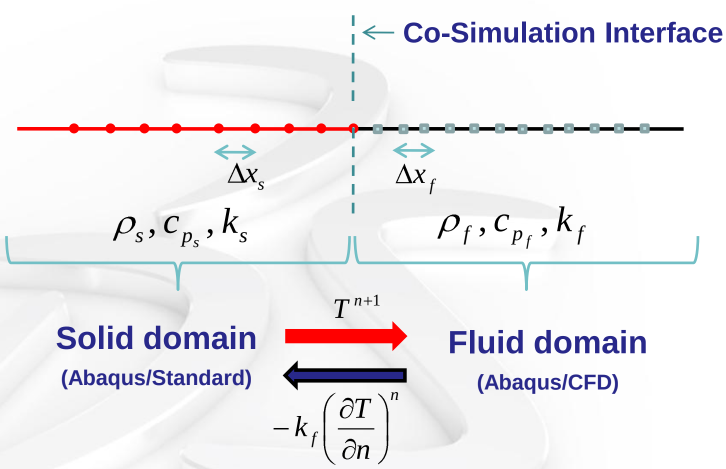
Conjugate Heat Transfer

- **Two approaches**
  - Model heat transfer in the solid completely within the CFD code
    - Not available in Abaqus/CFD
  - Model heat transfer in the solid using the structural solver and perform co-simulation with the CFD code
    - Available method
    - Dissimilar meshes can be used at the interface
    - Coupling utilizes the optimal time increment based on the time increment of both the structural solver and the CFD solver



Conjugate Heat Transfer

- **Stability of conjugate heat transfer analyses**
  - Conjugate heat transfer analyses are conditionally stable
    - The stability envelop is very large and is unlikely to be encountered in practical applications



Conjugate Heat Transfer

- The stability envelope

$$\Delta t_{crit} = \frac{\Delta x_f^2}{r\alpha_f},$$

where

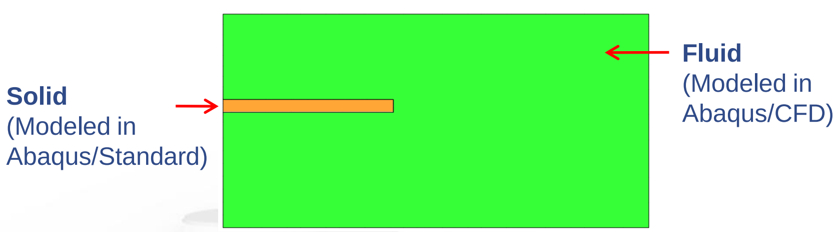
$$\alpha_f = \frac{k_f}{\rho_f c_{p_f}}, r = \frac{c_{p_f}\Delta x_f}{c_{p_s}\Delta x_s}$$

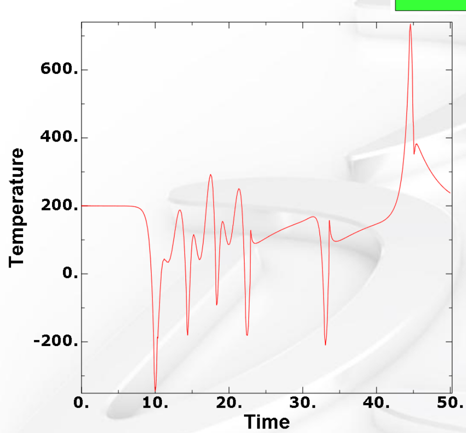
Materials	Δx_s	Δx_f	r	$\Delta t_{crit} = (\Delta x_f)^2/r\alpha_f$
Water-Steel	1×10^{-3}	1×10^{-6}	1.1×10^{-3}	6×10^{-3}
Liquid metal-Steel	1×10^{-3}	1×10^{-6}	6.4×10^{-4}	9×10^{-4}

- **Worst case scenario example**
  - Liquid metal/Steel
  - Fine turbulent boundary layer fluid mesh
  - Fine mesh in solid domain

Conjugate Heat Transfer

- **Examples of unstable analysis (triggered due to incorrect units)**





Unstable temperature for Δt = 0.2 > 0.02

Materials	r	Δt_{crit}
Water-Steel	0.2	~17
Incorrect units (**Units for C_p were given in J/g K instead of J/Kg K**)	0.2	0.02

Conjugate Heat Transfer

- **Tips**
  - For conjugate heat transfer analyses, the stability envelope is much wider
  - To gauge stability, monitor the interface temperature
  - If the analysis goes unstable:
    - Check the units of the material properties
    - Ensure that the physical properties are realistic

Notes

Notes

Postprocessing CFD/FSI Analyses

Lecture 8

Overview

- Abaqus/CAE Tips
- Isosurfaces
- View Cuts
- Vector Plots

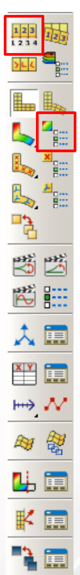
Abaqus/CAE Tips

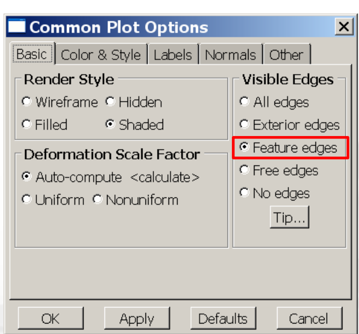
Abaqus/CAE Tips

- **Useful settings for postprocessing CFD output databases (ODBs)**

1

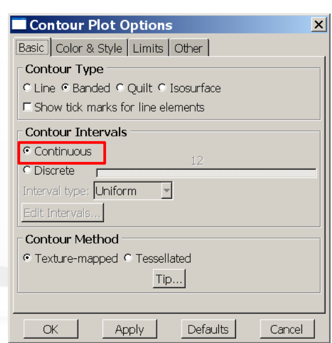
2





1

Turn off mesh display



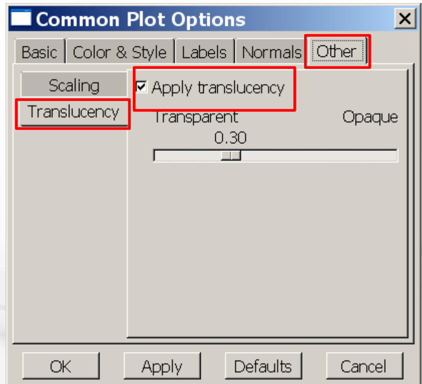
2

Use continuous contour intervals

Abaqus/CAE Tips

- **Useful settings (cont'd)**

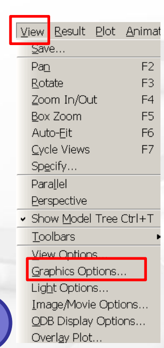
3



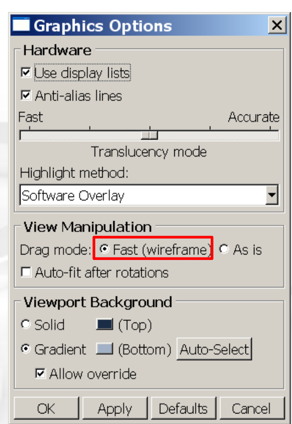
Activate translucency

Abaqus/CAE Tips

- **Useful settings (cont'd)**



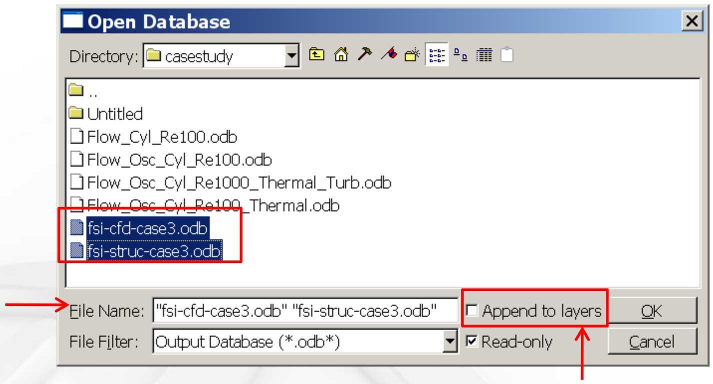
4



Fast view manipulation
(for large CFD models)

Abaqus/CAE Tips

- **Other tips**
  - Open multiple ODBs simultaneously
    - Structural and CFD models

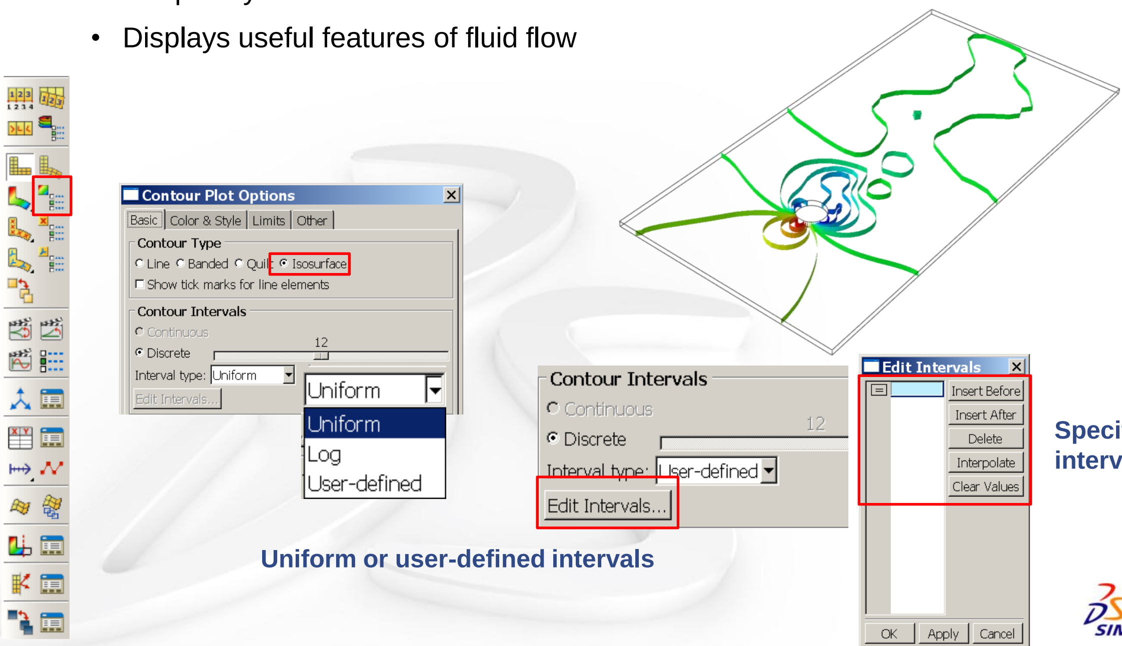


Can automatically overlay views from multiple output databases

Isosurfaces

Isosurfaces

- **An isosurface is a surface that represents points of constant value (e.g., pressure, temperature, velocity, etc.) within a volume**
  - Frequently used in CFD visualization
  - Displays useful features of fluid flow



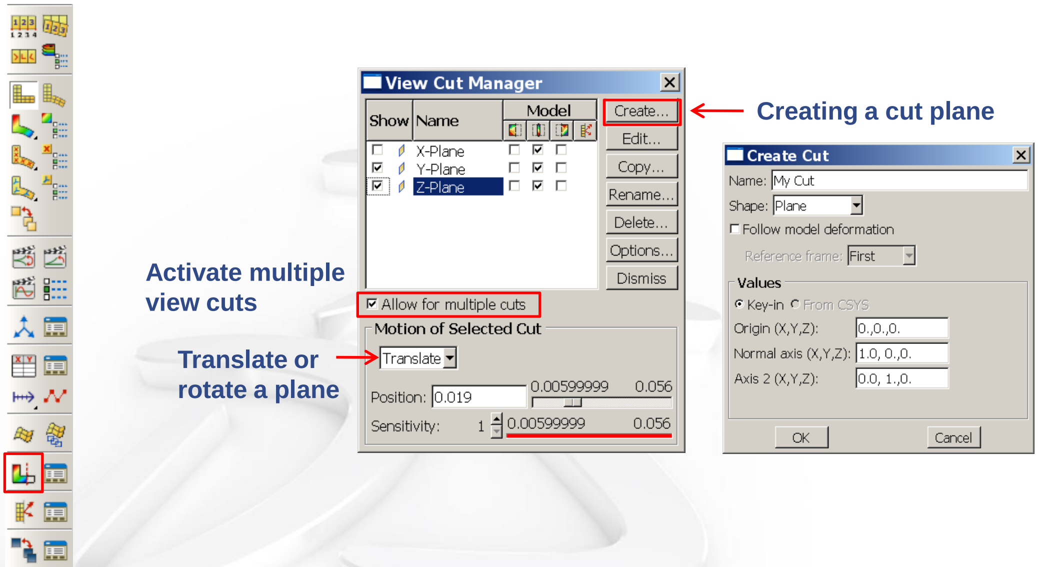
Uniform or user-defined intervals

Specify intervals

View Cuts

View Cuts

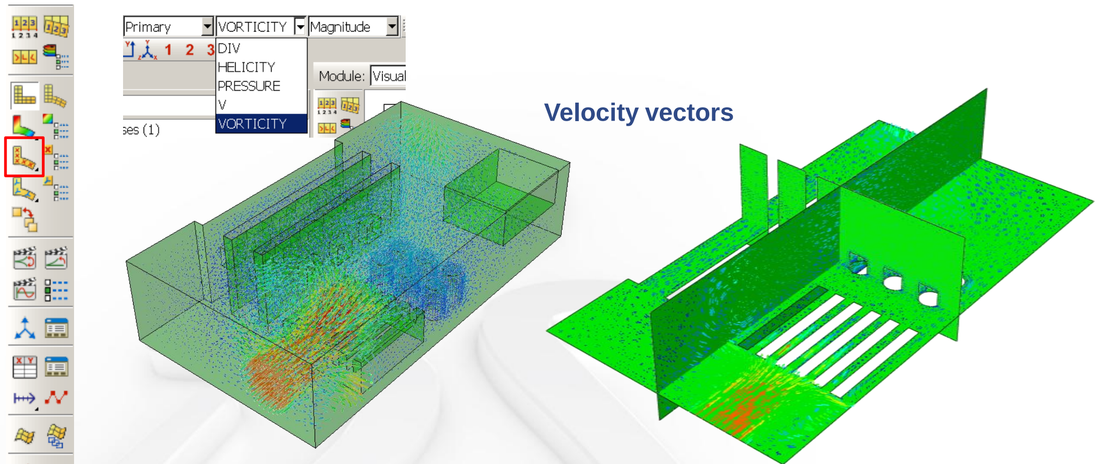
- **View cuts can be used to visualize the interior of the flow domain**
  - Multiple view cuts can be used simultaneously



Vector Plots

Vector Plots

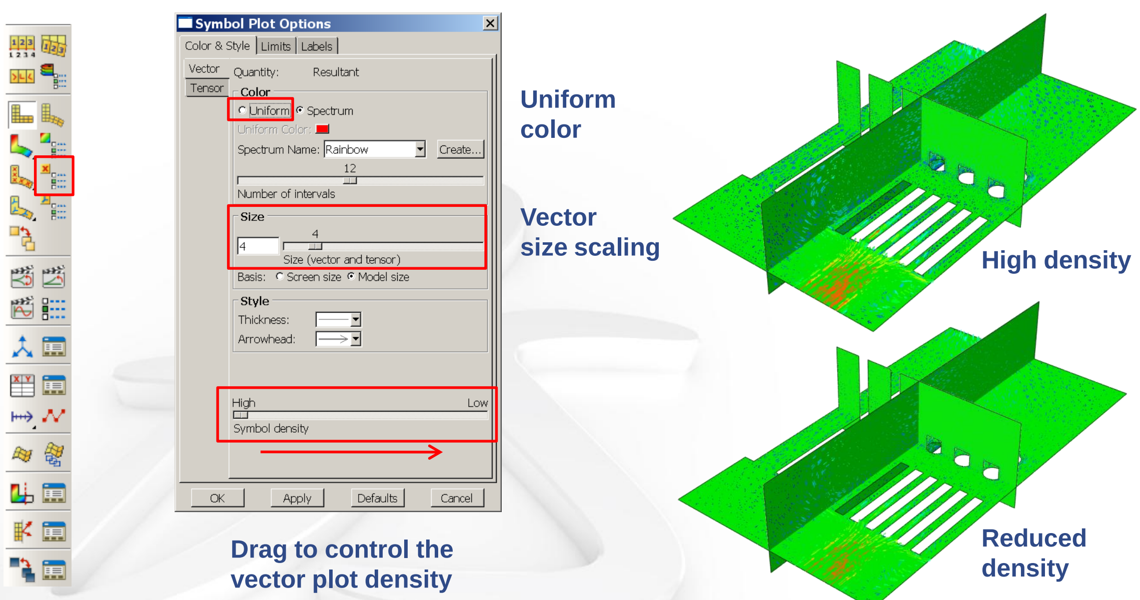
- **Vector plots are often used in CFD visualization**
  - Can be used in conjunction with view cuts



- Velocity vectors, by default, are colored by the primary variable

Vector Plots

- **Vector density on the vector plot can be controlled**



Notes

Notes

Workshop Preliminaries

Setting up the workshop directories and files

If you are taking a public seminar, the steps in the following section have already been done for you: skip to **Basic Operating System Commands, (p. WP.2)**. If everyone in your group is familiar with the operating system, skip directly to the workshops.

The workshop files are included on the Abaqus release CD. If you have problems finding the files or setting up the directories, ask your systems manager for help.

Note for systems managers: If you are setting up these directories and files for someone else, please make sure that there are appropriate privileges on the directories and files so that the user can write to the files and create new files in the directories.

Workshop file setup

(Note: UNIX is case-sensitive. Therefore, lowercase and uppercase letters must be typed as they are shown or listed.)

1. Find out where the Abaqus release is installed by typing

   UNIX and Windows NT: **abq***xxx* **whereami**

   where **abq***xxx* is the name of the Abaqus execution procedure on your system. It can be defined to have a different name. For example, the command for the 6.10–1 release might be aliased to **abq6101**.

   This command will give the full path to the directory where Abaqus is installed, referred to here as *abaqus_dir*.

2. Extract all the workshop files from the course tar file by typing

   UNIX: **abq***xxx* **perl** *abaqus_dir***/samples/course_setup.pl**

   Windows NT: **abq***xxx* **perl** *abaqus_dir***\samples\course_setup.pl**

   Note that if you have Perl and the compilers already installed on your machine, you may simply type:

   UNIX: *abaqus_dir***/samples/course_setup.pl**

   Windows NT: *abaqus_dir***\samples\course_setup.pl**

3. The script will install the files into the current working directory. You will be asked to verify this and to choose which files you wish to install. Choose "**y**" for the appropriate lecture series when prompted. Once you have selected the lecture series, type "**q**" to skip the remaining lectures and to proceed with the installation of the chosen workshops.

Basic operating system commands

(You can skip this section and go directly to the workshops if everyone in your group is familiar with the operating system.)

Note: The following commands are limited to those necessary for doing the workshop exercises.

Working with directories

1. Start in the current working directory. List the directory contents by typing

   UNIX: **`ls`**

   Windows NT: **`dir`**

   Both subdirectories and files will be listed. On some systems the file type (directory, executable, etc.) will be indicated by a symbol.
2. Change directories to a workshop subdirectory by typing

   Both UNIX and Windows NT: **`cd`** *dir_name*
3. To list with a long format showing sizes, dates, and file, type

   UNIX: **`ls -l`**

   Windows NT: **`dir`**
4. Return to your home directory:

   UNIX: **`cd`**

   Windows NT: **`cd`** *home-dir*

   List the directory contents to verify that you are back in your home directory.
5. Change to the workshop subdirectory again.
6. The **`*`** is a wildcard character and can be used to do a partial listing. For example, list only Abaqus input files by typing

   UNIX: **`ls *.inp`**

   Windows NT: **`dir *.inp`**

Working with files

Use one of these files, *filename*.**`inp`**, to perform the following tasks:

1. Copy *filename*.**`inp`** to a file with the name **`newcopy.inp`** by typing

   UNIX: **`cp`** *filename*.**`inp newcopy.inp`**

   Windows NT: **`copy`** *filename*.**`inp newcopy.inp`**
2. Rename (or move) this new file to **`newname.inp`** by typing

   UNIX: **`mv newcopy.inp newname.inp`**

   Windows NT: **`rename newcopy.inp newname.inp`**

   (Be careful when using **`cp`** and **`mv`** since UNIX will overwrite existing files without warning.)

3. Delete this file by typing

   UNIX: **`rm newname.inp`**

   Windows NT: **`erase newname.inp`**

4. View the contents of the files *filename*.**`inp`** by typing

   UNIX: **`more`** *filename*.**`inp`**

   Windows NT: **`type`** *filename*.**`inp | more`**

   This step will scroll through the file one page at a time.

Now you are ready to start the workshops.

Notes

Notes

Workshop 1

Unsteady flow across a circular cylinder

Introduction

The phenomenon of vortex shedding is important in engineering applications such as heat exchangers, nuclear reactor fuel rod assemblies, suspension bridge and other numerous applications. For flow passing over a stationary cylinder, experimental observations and numerical predictions have shown that a vortex sheet in the wake of the cylinder is formed which induces unsteady lift and drag forces on the cylinder. The unsteadiness in the fluid forces can induce vibrations on structures which need to be considered during their design.

In this workshop, we analyze the unsteady flow across a circular cylinder at a Reynolds number of 100. This classical problem forms the basis of many engineering problems where both the vortex generation as well as vortex-induced vibrations need to be considered. Three different cases are presented in this workshop:

Case 1: Unsteady flow across a stationary circular cylinder.

Case 2: Unsteady flow across an oscillating circular cylinder where the cylinder oscillation is prescribed.

Case 3: Unsteady flow across a spring-loaded rigid cylinder where the cylinder oscillates due to its interaction with the fluid.

The first two cases can be completely modeled within Abaqus/CFD. Modeling unsteady flow over an oscillating cylinder requires invoking the arbitrary Lagrangian-Eulerian (ALE) methodology within Abaqus/CFD where the mesh is deformed to accommodate the boundary displacements. The third case is an example of fluid-structure interaction (FSI) where oscillations of the spring-loaded cylinder occur due to its interaction with the fluid. Modeling this phenomenon requires modeling the spring-loaded cylinder in Abaqus/Standard or Abaqus/Explicit and modeling the fluid flow in Abaqus/CFD and coupling the distinct physical fields through FSI.

The CFD models are set up such that the flow Reynolds number based on the cylinder's diameter ($\rho Vd/\mu$) is equal to 100.

Fluid model

The Abaqus/CFD model representing the fluid domain is shown in Figure W1–1. The fluid model used in the three cases is the same. The diameter of the cylinder is 0.1 m. The computational model dimensions have been chosen such that the inlet, outlet and far-field boundaries are far enough from cylinder's surface to avoid any boundary effects. The inlet is placed 4 diameters away from cylinder's center while the outlet is 12 diameters away. The far-field boundaries are each placed 4 diameters away from the cylinder center.

We will model the 2D flow across the cylinder. Since Abaqus/CFD only offers a 3D solver, we will model the through-thickness direction with one-element and impose appropriate symmetry boundary conditions on the faces in the through-thickness direction to recover 2D behavior.

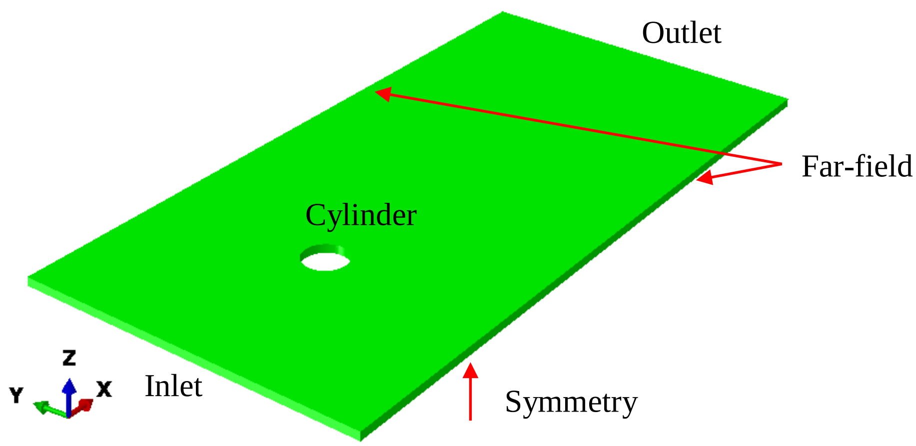


Figure W1–1 CFD model

The model consists of 3564 hexahedral fluid elements (FC3D8) with one-element in the through-thickness direction. The CFD mesh used in the current workshop is very coarse and a finer near-wall mesh would be required to resolve flow gradients near the surfaces. However, the mesh used in the cases presented here is adequate to show the development of the vortex sheet in the wake of the cylinder.

The fluid is modeled as an incompressible Newtonian fluid. The properties of the fluid are chosen to achieve a flow Reynolds number of 100 based on cylinder's diameter and the inlet velocity. The fluid density is chosen to be 1000 kg/m^3 and the viscosity is 0.1 Pa·sec. The fluid is assumed to be quiescent and hence, the initial velocity is zero everywhere.

The Abaqus/CFD procedure invokes a transient incompressible laminar flow analysis. Automatic time incrementation based on a fixed Courant-Freidrichs-Lewy (CFL) condition is used.

Preliminaries

1. Enter the working directory for this workshop:

    `../cfd/cylinder`

2. Run the script **`ws_cfd_cylinder.py`** using the following command:

    `abaqus cae startup=ws_cfd_cylinder.py`

    The above command creates an Abaqus/CAE database named **`cylinder.cae`** in the current directory. The database contains two separate models. The model **stationary** defines the fluid domain while the model **solid** defines the spring-loaded rigid circular cylinder.

Case 1

For unsteady flow across a stationary circular cylinder, the following boundary conditions are applied to the fluid.

Boundary conditions on the fluid

1. Inlet: An inlet velocity of 0.1 m/sec is assumed.
2. Outlet: An outlet boundary condition is specified with the fluid pressure set to zero.
3. Cylinder surface: A no-slip/no-penetration wall boundary condition is applied at the cylinder surface. All velocity components are set equal to zero.
4. Far-field: The far-field velocity is assumed to be equal to the inlet velocity (i.e., the *x*-component of velocity is set equal to 0.1 m/sec). This is a reasonable choice if the far-field boundaries are far away from the cylinder surface. Alternatively, a traction-free condition can be enforced (i.e., no BCs prescribed).
5. Symmetry: The velocity normal to the symmetry planes (V_z) is assumed to be zero to constrain the out-of-plane flow.

The boundary conditions for Case 1 are depicted in Figure W1–2.

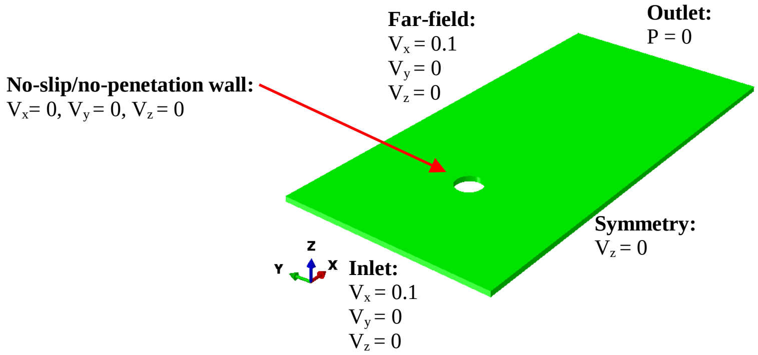


Figure W1–2 Boundary conditions for the CFD model for Case 1

Case 1 is run for a total simulation time of 120 s which is sufficient to allow for the development of the vortex sheet.

Completing the CFD model

In this section, you will complete the CFD model.

1. In the Model Tree, expand the container under the model **stationary**.
2. Define the material properties.
    a. In the Model Tree, double-click **Materials** and create a new material named **`fluid`**.
    b. From the **General** menu of the material editor, select **Density** and enter a value of `1000` kg/m^3.
    c. From the **Mechanical** menu of the material editor, select **Viscosity** and enter a value of `0.1` Pa·sec.
3. Define and assign the CFD section.
    a. In the Model Tree, double-click **Sections** and create a new section named **`fluid`**. Note that a homogeneous fluid section is the only choice available for CFD models. Click **Continue**.
    b. In the **Edit Section** dialog box that appears, select **fluid** as the **Material** and click **OK**.
    c. Assign the CFD section.
        a. In the Model Tree, expand the **Parts** container. Expand the container for the part named **`domain`**.
        b. Double-click **Section Assignments**.
        c. In the prompt area, click **Sets**. In the **Region Selection** dialog box, choose **`all`** and toggle on **Highlight selections in viewport** to identify the region. Click **Continue**.

4. Define an incompressible laminar flow analysis step.
    a. In the Model Tree, double-click **Steps**.
    b. In the **Create Step** dialog box accept the default procedure type **Flow** and click **Continue**.
    c. In the step editor, under the **Basic** tab, do the following:
        i. In the **Description** field, enter `Flow around a cylinder`.
        ii. In the **Time period** field, enter `120` sec.
    d. In the step editor, under the **Incrementation** tab, accept all default settings. An initial time increment of `0.01` sec and the automatic time incrementation strategy based on a fixed CFL number of `0.45` is used.
    e. In the step editor, under the **Solvers** tab, accept the default settings on the **Momentum Equation**, **Pressure Equation** and **Transport Equation** tabs.
    f. In the step editor, under the **Turbulence** tab, accept the default setting of **None** under **Turbulence Model**.
5. Define output requests.
    a. In the Model Tree, expand the **Field Output Requests** container.

       Note that a default field output request named **F-Output-1** was automatically created at the time the step was created.
    b. Double-click **F-Output-1**. Note that the output is requested at `20` evenly-spaced time intervals. Change the number of time intervals to `100`.
    c. Expand the output variable identifier containers and toggle on the following output variables: **V**, **PRESSURE**, **DIV**, **HELICITY**, and **VORTICITY**.
6. Define boundary conditions.
    a. Define a no-slip/no-penetration boundary condition at the cylinder surface.
        i. In the Model Tree, double-click **BCs**.
        ii. Name the boundary condition `noSlip` and select **Step-1** as the step.
        iii. Select **Fluid** as the category and **Fluid wall condition** as the type.
        iv. Select **domain-1.cylinder** as the surface to which the boundary condition will be applied.
        v. Select **No slip** as the condition.
    b. Define an inlet boundary condition at the inlet surface.
        i. In the Model Tree, double-click **BCs**.
        ii. Name the boundary condition `inlet` and select **Step-1** as the step.

   iii. Select **Fluid** as the category and **Fluid inlet/outlet** as the type.
   iv. Select **domain-1.inlet** as the surface to which the boundary condition will be applied.
   v. In the **Momentum** tab, toggle on **Specify** and then select **Velocity**.
   vi. Set the *x*-velocity **V1** to `0.1`. Set the *y*- and *z*-velocity components **V2** and **V3** to `0`.

c. Define an outlet boundary condition at the outlet surface.
   i. In the Model Tree, double-click **BCs**.
   ii. Name the boundary condition **`outlet`** and select **Step-1** as the step.
   iii. Select **Fluid** as the category and **Fluid inlet/outlet** as the type.
   iv. Select **domain-1.outlet** as the surface to which the boundary condition will be applied.
   v. In the **Momentum** tab, toggle on **Specify** and then select **Pressure**.
   vi. Set the pressure to `0`.

d. Define velocity boundary conditions at the far-field surfaces.
   i. In the Model Tree, double-click **BCs**.
   ii. Name the boundary condition **`far field`** and select **Step-1** as the step.
   iii. Select **Fluid** as the category and **Fluid inlet/outlet** as the type.
   iv. Select **domain-1.farfield** as the surface to which the boundary condition will be applied.
   v. In the **Momentum** tab, toggle on **Specify** and then select **Velocity**.
   vi. Set the *x*-velocity **V1** to `0.1`. Set the *y*- and *z*-velocity components **V2** and **V3** to `0`.

e. Define velocity boundary conditions at the symmetry planes.
   i. In the Model Tree, double-click **BCs**.
   ii. Name the boundary condition **`symm`** and select **Step-1** as the step.
   iii. Select **Fluid** as the category and **Fluid inlet/outlet** as the type.
   iv. Select **domain-1.symm** as the surface to which the boundary condition will be applied.
   v. In the **Momentum** tab, toggle on **Specify** and then select **Velocity**.
   vi. Set the *z*-velocity **V3** to `0`. Leave the *x*- and *y*-velocity components unspecified.

This completes the CFD model set up for Case 1.

Creating a CFD analysis job

1. In the Model Tree, expand the **Analysis** container.
2. Double-click **Jobs**.
3. In the **Create Job** dialog box, select the model **stationary** and name the job **`stationary-flow`**.

Running the CFD analysis

Now that the model set up is complete, run the CFD analysis job. The job can be launched from within Abaqus/CAE as follows: Click mouse-button 3 on the CFD analysis job name and select **Submit** from the menu that appears.

Monitoring the CFD analysis

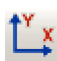
While the job is running, you can monitor its progress.

1. Click mouse-button 3 on the CFD analysis job name and select **Monitor** from the menu that appears.
2. The job monitor appears. Note that time incrementation information, divergence (RMS) and kinetic energy is updated every time increment.

Viewing the results

Once the job completes, do the following.

1. Click mouse-button 3 on the CFD analysis job name and select **Results** from the menu that appears.
   The output database file **`stationary-flow.odb`** opens in the Visualization module.
2. Create pressure and velocity contour plots and a pressure line plot.

   a. Click  to set the view.

   b. In the toolbox, click  to create a contour plot (alternatively select **Plot→Contours→On Undeformed Shape**).

   c. From the main menu bar, select **Result → Field Output**. In the **Field Output** dialog box, select **PRESSURE** as the output variable and click **OK**.

      Tip: You may also select the variable from the **Field Output** toolbar.

   d. In the toolbox, click  to open the **Common Plot Options** dialog box. Toggle on **Feature edges** for the visible edges and click **OK**. This turns off the mesh feature lines in the model.

   e. In the toolbox, click  to open the **Contour Plot Options** dialog box. Toggle on **Continous** under **Contour Intervals** and click **Apply**. This creates a smooth pressure contour plot, as shown in Figure W1–3 (left).

f. In the **Contour Plot Options** dialog box, toggle on **Line** under **Contour Type**. Move the **Discrete** slider bar under **Contour Intervals** to **24** and click **Apply**. This creates a line plot as shown in Figure W1–4.

g. Using the **Field Output** toolbar, select **V** as the output variable to plot. Reset the contour plot options to their default settings with the exception of the **Continous** contour intervals. A contour plot of the velocity appears as shown in Figure W1–3 (right).

h. In the toolbox, click  to create a time history animation. Select an output variable (velocity, pressure, etc.) to animate the results.

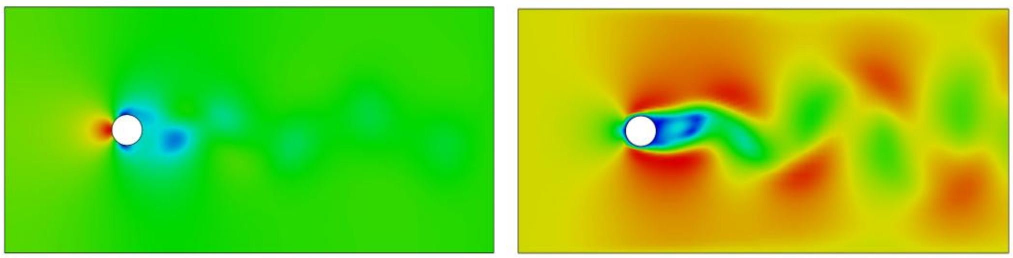


Figure W1–3 Pressure and velocity contour plot for Case 1 at time = 120 sec

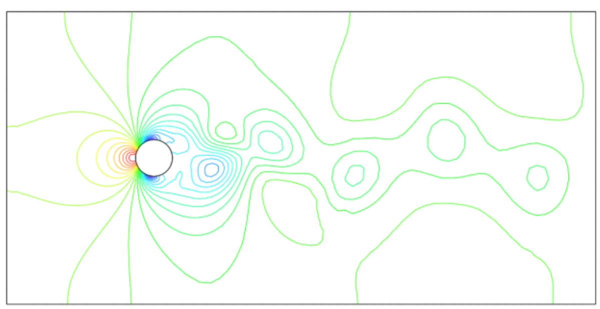


Figure W1–4 Pressure line plot for Case 1 at time = 120 sec

3. Create a pressure isosurface plot.

a. Using the **Field Output** toolbar, select **PRESSURE** as the output variable to plot.

c. In the **Contour Plot Options** dialog box, toggle on **Isosurface** under **Contour Type**. Set the position of the **Discrete** slider bar to **24** and click **OK**. Adjust the view to more clearly see the isosurfaces, as shown in Figure W1–5.

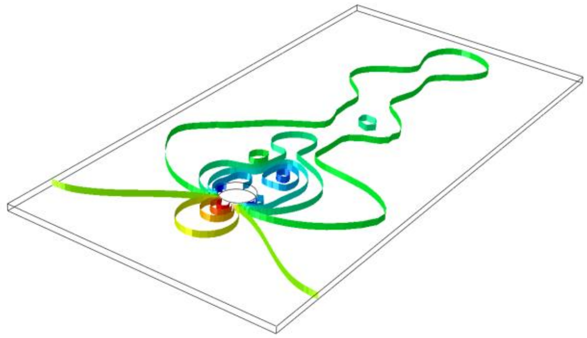


Figure W1–5 Pressure isosurfaces for Case 1 at t = 120 sec

Case 2

Unsteady flow across an oscillating circular cylinder requires prescribing the cylinder's oscillation as a function of time. Additionally, since ALE and mesh deformation will be activated to accommodate the displacement of the cylinder, appropriate boundary conditions are required for the mesh deformation solution. The following modifications to the boundary conditions are required.

Boundary conditions on the fluid

1. Cylinder surface: A no-slip/no-penetration wall boundary condition requires that the fluid velocity at the wall remain equal to the cylinder's velocity. The following time-dependent velocity is prescribed using an amplitude definition:

$$V_x = \frac{2\pi A_o}{T}\cos\left(\frac{2\pi t}{T}\right)$$

Boundary conditions on the mesh

1. Cylinder surface: Since the velocity of the cylinder has been prescribed, the displacement of the cylinder is also known. The mesh displacement at the cylinder surface is hence known and will have to be specified. The following time-dependent mesh displacement is prescribed using an amplitude definition:

$$U_x = A_o \sin\left(\frac{2\pi t}{T}\right)$$

   It should be noted that the mesh displacement at the cylinder surface is not independent but is *kinematically* related to the cylinder's velocity (and hence the fluid velocity).

2. Inlet, Outlet and Far-field: The mesh is fixed by prescribing zero-valued mesh displacement boundary conditions ($U_x = U_y = U_z = 0$).

3. Symmetry: The mesh motion normal to the symmetry planes is constrained by prescribing $U_z = 0$.

The boundary conditions for Case 2 are depicted in Figure W1–6.

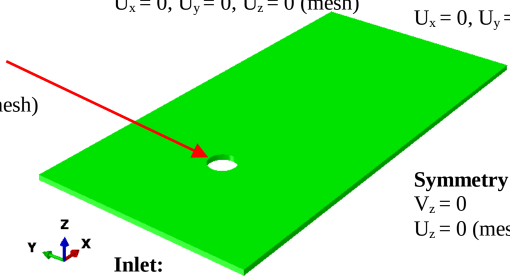


Figure W1–6 Boundary conditions for the CFD model for Case 2

Case 2 is run for a total of 4 s which represents two cycles of prescribed oscillation.

Completing the CFD model

In this section, you will complete the CFD model.

1. In the Model Tree, click mouse button 3 on the model named **stationary**. From the menu that appears, select **Copy Model**. Name the new model **`oscillating`**.

2. In the Model Tree, expand the container under the model **oscillating**.

3. Modify the CFD analysis step.

   a. Expand the **Steps** container and double-click **Step-1**.

   b. In the **Basic** tabbed page of the step editor:

      i. Modify the description to read **`Flow around an oscillating cylinder`**.

      ii. Set the time period of the step to **`4`** sec.

2. Modify the output requests.

   a. In the Model Tree, expand the **Field Output Requests** container.

   b. Double-click **F-Output-1**. Ensure that the mesh displacement output **U** is turned on. This variable is required to contour the mesh displacements.

   c. Select **Every x units of time** as the output frequency option and enter **`0.1`** sec as the value of **x**.

3. Create amplitude curves to define the time-dependent displacement and velocity.
   a. In the Model Tree, double-click **Amplitudes**.
   b. Name the amplitude curve **`disp`**.
   c. Select **Periodic** as the type.
   d. Accept **Step time** as the time span.
   e. Enter **`pi`** for the **Circular frequency**, **`0`** for the **Starting time** and **`0`** for the **Initial amplitude**. Enter **`0`** for **A** and **`1`** for **B**.
   f. Repeat the previous steps to define a periodic amplitude curve named **`vel`**. Enter **`pi`** for the **Circular frequency**, **`0`** for the **Starting time** and **`0`** for the **Initial amplitude**. Enter **`pi`** for **A** and **`0`** for **B**
4. Modify the fluid boundary conditions.
   a. Modify the no-slip/no-penetration boundary condition at cylinder surface.
      i. In the Model Tree, expand the **BCs** container.
      ii. Click mouse button 3 on the boundary condition named **`noSlip`** and select **Delete** from the menu that appears.

         This deletes the boundary condition on cylinder surface.
      iii. In the Model Tree, double-click **BCs** to create a new boundary condition named **`wall`**.
      iv. Select **Fluid** as the category and **Fluid wall condition** as the type.
      v. Select **domain-1.cylinder** as the surface to which the boundary condition will be applied.
      vi. Select **Shear** as the condition.
      vii. Set the *x*-velocity **V1** to **`0.05`**. Set the *y*- and *z*-velocity components **V2** and **V3** to **`0`**.
      viii. Select **vel** as the amplitude curve.
5. Create boundary conditions for the mesh motion.
   a. Define the fixed mesh condition at the inlet, outlet, and far-field boundaries.
      i. In the Model Tree, double-click **BCs** to create a new boundary condition named **`mesh-fixed`**.
      ii. Select **Mechanical** as the category and **Displacement/Rotation** as the type.
      iii. Select **domain-1.fixed** as the set to which the boundary condition will be applied.
      iv. Set **U1**, **U2**, and **U3** to **`0`**.
   b. Define the symmetry condition to constrain the motion of the mesh in the through-thickness direction.

i. In the Model Tree, double-click **BCs** to create a new boundary condition named **`mesh-symm`**.

ii. Select **Mechanical** as the category and **Displacement/Rotation** as the type.

iii. Select **domain-1.symm** as the set to which the boundary condition will be applied.

iv. Set **U3** to `0`.

c. Specify the mesh displacement boundary condition on the cylinder surface.

i. In the Model Tree, double-click **BCs** to create a new boundary condition named **`mesh-wall`**.

ii. Select **Mechanical** as the category and **Displacement/Rotation** as the type.

iii. Select **domain-1.cylinder** as the set to which the boundary condition will be applied.

iv. Set **U1** to `0.05` and **U2** and **U3** to `0`.

v. Select **disp** as the amplitude curve.

This completes the CFD model set up for Case 2.

Creating a CFD analysis job

1. In the Model Tree, expand the **Analysis** container.
2. Double-click **Jobs**.
3. In the **Create Job** dialog box, select the model **stationary** and name the job **`oscillating-flow`**.

Running the CFD analysis

Now that the model set up is complete, run the CFD analysis job. The job can be launched from within Abaqus/CAE as follows: Click mouse-button 3 on the CFD analysis job name and select **Submit** from the menu that appears.

Monitoring the CFD analysis

While the job is running, you can monitor its progress.

1. Click mouse-button 3 on the CFD analysis job name and select **Monitor** from the menu that appears.
2. The job monitor appears. Note that time incrementation information, divergence (RMS) and kinetic energy is updated every time increment.

Viewing the results

Once the job completes, do the following.

1. Click mouse-button 3 on the CFD analysis job name and select **Results** from the menu that appears.

The output database file **`oscillating-flow.odb`** opens in the Visualization module.

2. Repeat the steps described earlier to create contour plots for pressure and velocity (shown in Figure W1–7).
3. Plot the mesh displacements at various times (as shown in Figure W1–8).

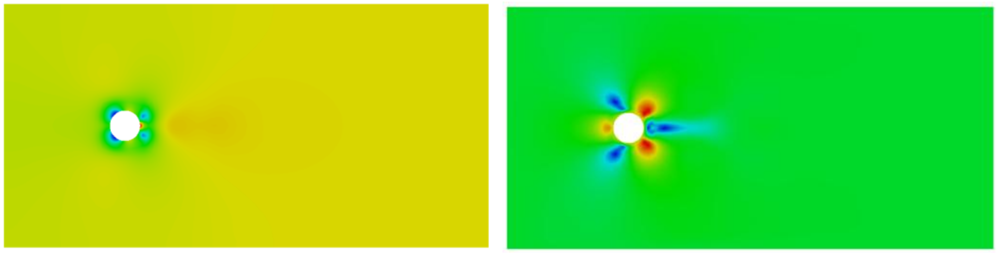


Figure W1–7 Pressure and velocity contour plot for Case 2 at time = 4 sec

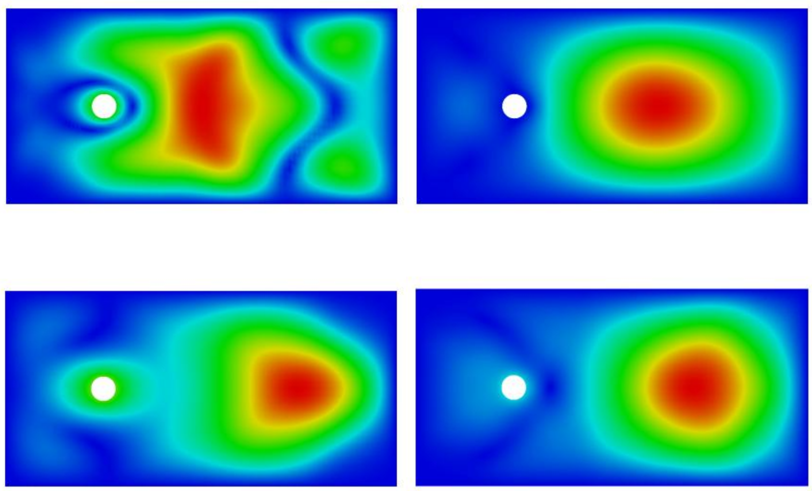


Figure W1–8 Mesh displacement contour plots for Case 2 at time = approximately 1, 2, 3, and 4 seconds

S T O P!

Continue with the remainder of this workshop after the completion of Lecture 6.

Case 3

Unsteady flow across a spring-loaded circular cylinder requires co-simulation with Abaqus/Standard or Abaqus/Explicit. In this case the spring-loaded cylinder will be modeled with Abaqus/Standard. The boundary conditions required on the fluid and the mesh for the CFD model remain the same as in Case 2 except that the following boundary conditions need to be suppressed:

Boundary conditions on the fluid

1. Cylinder surface: The no-slip/no-penetration wall boundary condition on the cylinder surface is suppressed. The fluid velocity will be dictated by the coupled solution.

Boundary conditions on the mesh

1. Cylinder surface: The mesh displacement BC at the cylinder surface is suppressed. This displacement will be dictated by the coupled solution.

The boundary conditions for Case 3 are depicted in Figure W1–9.

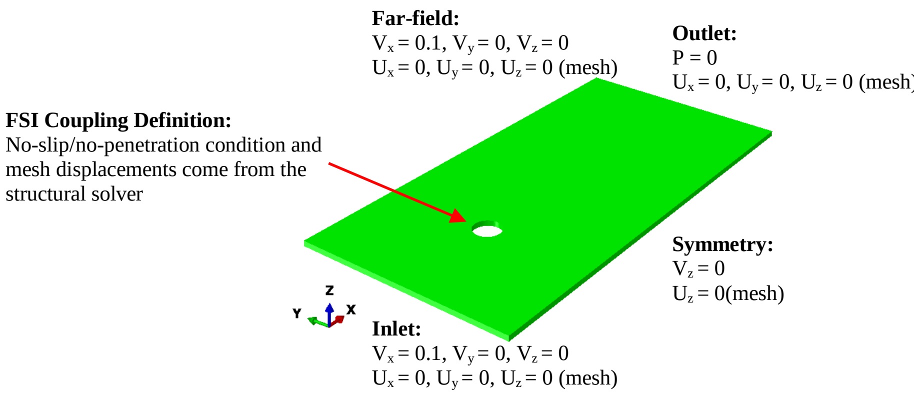


Figure W1–9 Boundary conditions for the CFD model for Case 3

The CFD model includes a surface definition representing the region of the fluid which interacts with the cylinder surface. It will be used to define the co-simulation interaction with the Abaqus/Standard model.

Structural model

The Abaqus/Standard model of the spring-loaded cylinder is shown in Figure W1–10.

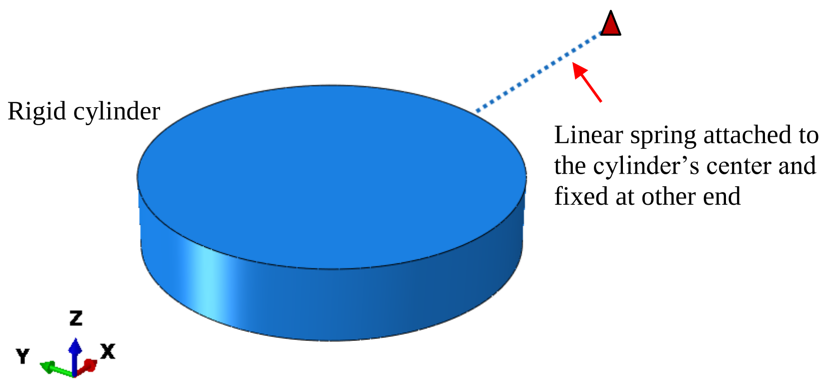


Figure W1–10 Structural model for spring-loaded cylinder

The structural model in Abaqus/Standard is comprised of first-order hexahedral stress/displacement elements (C3D8R). A total of 57 elements are used to define the cylinder. A density of 8000 kg/m^3, Young's modulus of 200 GPa, and Poisson's ratio of 0.3 are used to define the cylinder's material properties.

A rigid-body constraint has been applied to model the cylinder as a rigid body. The cylinder is connected to a linear spring. The spring is modeled in Abaqus/Standard as a connector with axial behavior. A spring stiffness of 1 N/m has been specified. The spring stiffness is chosen to illustrate the coupled physics and allow for appreciable displacement of the cylinder (~20% of the cylinder's diameter).

The cylinder's surface interacts with the surrounding fluid and hence it is used to define the co-simulation interaction with the Abaqus/CFD model.

The Abaqus/Standard procedure invokes an implicit dynamic analysis step. An initial time increment of 0.01 s is used; however, the time increment can change depending on whether the structural or CFD model is dictating the time increment size. The built-in time incrementation strategy is used where the co-simulation coupling time is chosen as the minimum of the time increments dictated by the structural and CFD models. The total simulation time is chosen to be 25 s.

Completing the CFD model

In this section, you will complete the CFD model.

1. In the Model Tree, click mouse button 3 on the model named **oscillating**. From the menu that appears, select **Copy Model**. Name the new model **`fluid`**.
2. In the Model Tree, expand the container under the model **fluid**.
3. Modify the CFD analysis step.
   a. Expand the **Steps** container and double-click **Step-1**.
   b. In the **Basic** tabbed page of the step editor:
      i. Modify the description to read **`Flow around a spring-loaded cylinder`**.
      ii. Set the time period of the step to **`25`** sec.
4. Modify the output requests.
   a. In the Model Tree, expand the **Field Output Requests** container.
   b. Double-click **F-Output-1**.
   c. Set the output frequency to **`1`** sec.
5. Modify the fluid boundary conditions.
   a. In the Model Tree, expand the **BCs** container.
   b. Click mouse button 3 on the boundary condition named **`wall`** and select **Delete** from the menu that appears.

      This deletes the fluid velocity boundary condition on the cylinder surface.
6. Modify the mesh boundary conditions.
   a. In the **BCs** container, click mouse button 3 on the boundary condition named **`mesh-wall`** and select **Delete** from the menu that appears.

      This deletes the mesh displacement boundary condition on the cylinder surface.
7. Define the FSI interaction.
   a. In the Model Tree, double-click **Interactions**.
   b. Name the interaction **`fsi`**.
   c. Select **Step-1** as the step in which it will be defined and accept **Fluid-Structure Co-simulation boundary** as the type.
   d. Select **domain-1.cylinder** as the surface to which the interaction will be applied.

Completing the structural model

In this section, you will complete the structural model. The model is complete with the exception of the FSI interaction.

1. In the Model Tree, expand the container under the model **solid**.
2. Define the FSI interaction.
    a. In the Model Tree, double-click **Interactions**.
    b. Name the interaction `fsi`.
    c. Select **vortex_vibrations** as the step in which it will be defined and **Fluid-Structure Co-simulation boundary** as the type.
    d. Select **solidCylinder-1.cylinder** as the surface to which the interaction will be applied.

Creating a co-execution analysis

In order to perform the fluid-structure interaction analysis, the Abaqus/Standard and Abaqus/CFD jobs need to execute together. A co-simulation is performed where the two solvers exchange information at each co-simulation target time. The co-simulation target time is automatically chosen as the minimum of the time increments required by the structural and CFD solvers. In order to facilitate the co-simulation of the two analyses, the co-execution job procedure is used. A co-execution job creates two analysis jobs and runs them simultaneously. It also automatically provides the driver options needed for communication between the two analysis jobs.

1. In the Model Tree, expand the **Analysis** container.
2. Double-click **Co-executions** and create a co-execution named `fsi_cylinder`.
3. In the **Edit Co-execution** dialog box:
    a. Select **solid** as the first model. Change the job name to `fsi-solid`.
    b. Select **fluid** as the second model. Change the default job name to `fsi-fluid`.
    c. Click **OK**.

In the Model Tree, expand the **Co-executions** container and then expand the **fsi_cylinder** container. Expand the **Jobs** container under **fsi_cylinder**.
Note that two analyses jobs have been created – one representing the Abaqus/Standard structural model and the other representing the Abaqus/CFD model.

Running the co-simulation analysis

Launch the co-execution job from within Abaqus/CAE.

1. Click mouse button 3 on the co-execution job **fsi_cylinder**.
2. From the menu that appears, select **Submit**.
   This launches the co-execution job. Both the Abaqus/Standard and Abaqus/CFD jobs will be launched.

Monitoring the co-execution analysis

While the co-execution is running, you can monitor its progress.

1. Click mouse-button 3 on the CFD analysis job name and select **Monitor** from the menu that appears.
2. The job monitor appears. Note that time incrementation information, divergence (RMS) and kinetic energy is updated every time increment.

Viewing the results

Once the co-execution completes, do the following:

1. Click mouse button 3 on the co-execution named **fsi_cylinder**.
2. From the menu that appears, select **Results**.
   The output database files **`fsi-solid.odb`** and **`fsi-fluid.odb`** are opened simultaneously in the Visualization module and are overlaid in the viewport.
3. Toggle off the overlay plot option  in the toolbox.

Make the output database file **`fsi-solid.odb`** current.

1. From the main menu bar, select **Result→History Output**.
2. In the **History Output** dialog box, select **Spatial displacement: U1 PI: rootAssembly Node 1 in NSET REFPOINT** and click **Plot**. This creates a history plot of the cylinder's displacement (as shown in Figure W1–11).

Make the output database file **`fsi-fluid.odb`** current.

1. Create a contour plot of vorticity at time = 25 sec, as shown in Figure W1–12.
2. Also contour the pressure, velocity, mesh displacements, etc.

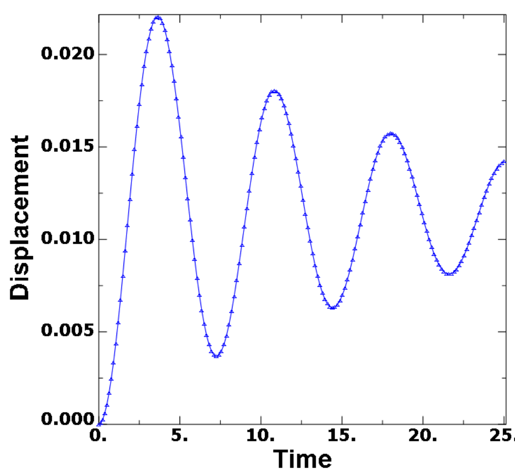


Figure W1–11 Cylinder's displacement for Case 3

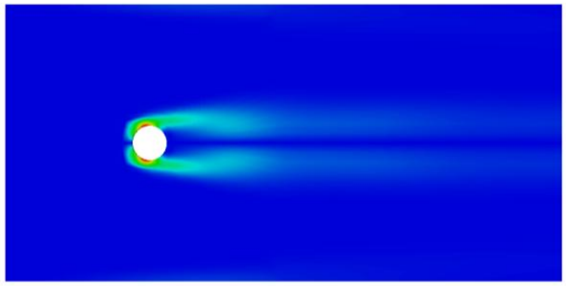


Figure W1–12 Vorticity plot for Case 3 at time = 25 sec

Note: A script that creates the complete model described in these instructions is available for your convenience. Run this script if you encounter difficulties following the instructions outlined here or if you wish to check your work. The script is named

`ws_cfd_cylinder_answer.py`

and is available using the Abaqus fetch utility.

Notes

Notes

Workshop 2

Heat transfer analysis of a component-mounted electronic circuit board

Introduction

This workshop considers the transient conjugate heat transfer between a single printed circuit board (PCB)–mounted electronic component and the ambient air. The component is subjected to passive power dissipation that results in the transfer of heat both within the component and the PCB due to conduction. Furthermore, the heated surfaces of the component and the PCB induce a temperature-dependent density differential in the surrounding air, thereby setting up a buoyancy-driven natural convection process external to the surface. Heat is thus transferred from the component and PCB surfaces to the ambient air through this convection process. Understanding the resulting conduction-convection conjugate heat transfer phenomenon allows more accurate damage estimation and life predictions for electronic components.

Heat transfer within the PCB and the electronic component is modeled in Abaqus/Standard using the heat transfer analysis procedure. The buoyancy-driven natural convection process in the surrounding air is modeled using Abaqus/CFD. Two models representing the solid and the fluid domains are created for the respective solvers in Abaqus/CAE. The models are defined in a CGS system of units.

Structural model

The Abaqus/Standard model of the PCB and the electronic component is shown in Figure W2–1. The PCB dimensions are 7.8 × 11.6 × 0.16 cm. The mounted electronic component consists of a 3 × 3 × 0.7 cm encapsulant that encapsulates a heat slug of size 1.8 × 1.8 × 0.3 cm mounted atop a die of dimensions 0.75 × 0.75 × 0.2 cm. The cross-sectional view of the assembled package is shown in Figure W2–2.

Various constituents of the PCB-component package and their material properties are listed in Table W2–1. The density, specific heat, and thermal conductivity properties have been predefined.

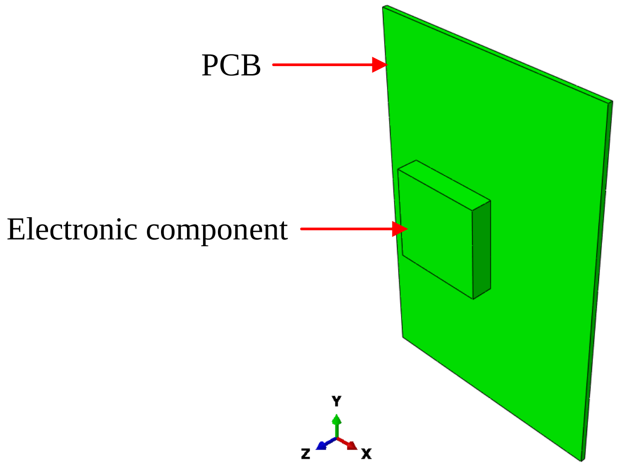


Figure W2–1 Component-mounted electronic circuit board

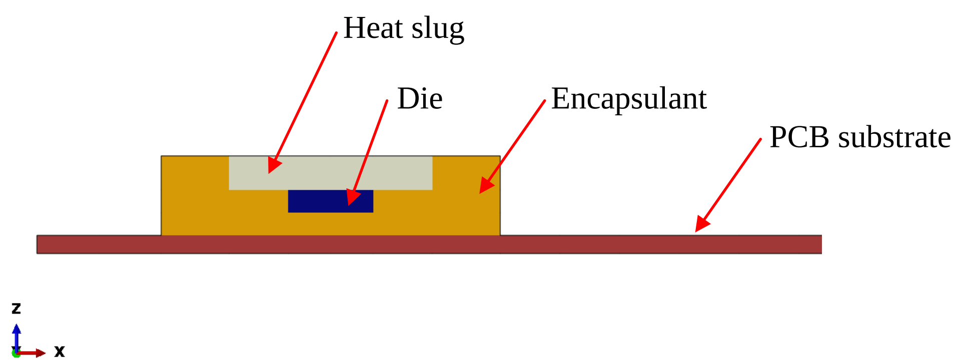


Figure W2–2 Cross-sectional view of the electronic package indicating the different material constituents of the electronic component and the PCB

	Density (g/cm^3)	Specific heat (erg/g/K)	Thermal conductivity (erg/cm/sec/K)	Thermal expansion (1/K)
PCB substrate	8.95	1×10^7	1.925×10^6	1.6×10^{-5}
Encapsulant	1.82	8.82×10^6	6.3×10^4	1.9×10^{-5}
Die	2.33	7.12×10^6	1.3×10^7	3.3×10^{-6}
Heat slug	8.94	3.85×10^4	3.98×10^5	3.3×10^{-6}

Table W2–1 Material properties of various constituents of the PCB-component package

The heat transfer model in Abaqus/Standard is comprised of first-order hexahedral thermal-diffusion elements (DC3D8). Different section properties are assigned with appropriate material choices for elements representing the various constituents of the PCB-component package. A total of 468 elements are used to define the heat transfer model.

The initial temperature of the assembled electronic package is 293 K.

The electronic component of the assembled package is subjected to passive power dissipation corresponding to a specified body heat flux of 5×10^6 erg/sec/cm^3.

The model contains a surface named **CIRCUIT-BOARD-1.FSI-SURF** comprising the exterior surface of the PCB and electronic component. The exterior surface interacts with the surrounding air and hence will be used to define the co-simulation interaction definition with the Abaqus/CFD model.

The Abaqus/Standard procedure invokes a transient heat transfer analysis. An initial time increment is 0.01 s is used; however, the time increment can change during the course of the analysis depending on the structural and fluid response. The built-in time incrementation strategy is used for co-simulation; the co-simulation coupling time is chosen as the minimum of the time increments dictated by the structural and CFD solvers. The total simulation time is chosen to be 15 s. While a longer analysis time is required to reach steady-state conditions, it is sufficient to illustrate the main concepts of the workshop.

Fluid model

The Abaqus/CFD model representing the surrounding air is shown in Figure W2–3.

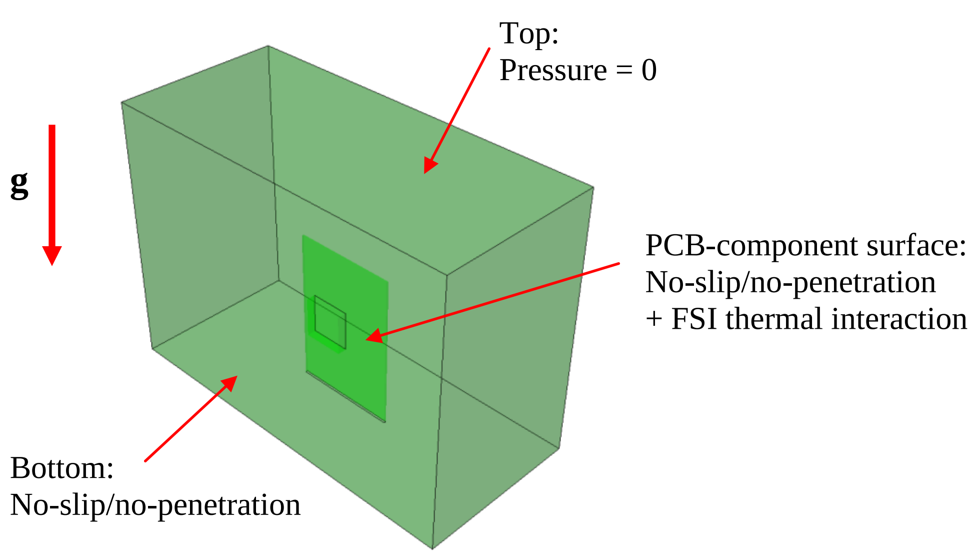


Figure W2–3 CFD model

The size of the CFD computational domain that encloses the PCB-component assembly is chosen to be 27.8 × 20 × 12.56 cm. With these dimensions the far-field boundaries are far enough away to not affect the flow behavior close to the PCB-component assembly.

Air is modeled as an incompressible Newtonian fluid. The properties of air are listed in Table W2–2. The thermal expansion coefficient for air is defined to enable thermal-momentum coupling due to buoyancy-driven natural convection flow. The heated PCB-component surface induces a temperature-dependent density differential in the surrounding air. This is modeled within the incompressible flow solver using a Boussinesq approximation. The source term in Navier-Stokes equation due to changes in density and gravity is linearized and the density-differential is assumed to be proportional to the temperature-differential and gravity.

	Density (g/cm^3)	**Specific heat (erg/g/K)**	**Thermal conductivity (erg/cm/sec/K)**	**Thermal expansion (/K)**	**Viscosity (Poise)**
Air	1.127×10^{-3}	1.0064×10^{7}	2710	3.43×10^{-3}	1.983×10^{-4}

Table W2–2 Properties of air

The CFD model consists of 58892 hexahedral fluid elements (FC3D8). This mesh is relatively coarse (a finer near-wall mesh would be required to resolve flow gradients near the surfaces).

The initial temperature of the air is 293 K. The fluid is assumed to be quiescent and hence, the initial velocity is zero everywhere.

The CFD boundary conditions are depicted in Figure W2–3. The following boundary conditions are applied:

1. The bottom surface of the fluid domain is assumed to be a rigid floor, and an adiabatic wall condition is assumed there. An adiabatic condition is the default do-nothing boundary condition for the fluid model so it need not be specified explicitly. Also, a no-slip/no-penetration wall boundary condition is applied at the bottom surface.

2. A no-slip/no-penetration wall boundary condition is applied at the surface representing the skin of the PCB-component assembly.

3. At the top surface, an outlet boundary condition is specified with the fluid pressure set to zero.

4. For all the other boundaries, free stream conditions are assumed for both the fluid velocity and temperature. This is the default do-nothing boundary condition for the fluid model so it need not be specified explicitly.

A gravity load is applied in the negative *y*-direction (see Figure W2–3) to activate buoyancy-driven natural convection heat transfer.

The CFD model also consists of a surface named **BOUNDINGBOX-1.CIRCUIT_BOARD_FSI** which represents the skin of the PCB-component surface. This surface interacts with the PCB-component assembly and hence will be used to define the interaction with the Abaqus/Standard model.

The Abaqus/CFD procedure invokes a transient incompressible laminar flow analysis coupled with the energy equation. Automatic time incrementation based on a fixed Courant-Freidrichs-Lewy (CFL) condition is used.

Preliminaries

1. Enter the working directory for this workshop:

   `../cfd/cboard`

2. Run the script **`ws_cfd_cboard.py`** using the following command:

   `abaqus cae startup=ws_cfd_cboard.py`

   The above command creates an Abaqus/CAE database named **`cboard.cae`** in the current directory. The database contains two separate models. The model **fluid** defines the fluid domain while the model **thermal** defines the PCB-component assembly.

Completing the CFD model

In this section, you will complete the CFD model by defining the following:

- Material properties
- Section properties
- Incompressible laminar flow analysis step with the energy equation activated
- Output requests
- Boundary conditions
- Gravity load
- Initial conditions
- FSI definition

1. In the Model Tree, expand the container under the model **fluid**.
2. Define the material properties.
    a. In the Model Tree, double-click **Materials** and create a new material named **`air`**.
    b. From the **General** menu of the material editor, select **Density** and enter a value of **`1.127e-3`** g/cm^3.
    c. From the **Thermal** menu of the material editor, select **Conductivity** and enter a value of **`2710`** g/cm^3.
    d. From the **Thermal** menu of the material editor, select **Specific Heat**. Choose **Constant Pressure** as the type and enter a value of **`1.0064e7`** g/cm^3.
    e. From the **Mechanical** menu of the material editor, select **Expansion** and enter a value of **`3.43e-3`** g/cm^3.
    f. From the **Mechanical** menu of the material editor, select **Viscosity** and enter a value of **`1.983e-4`** g/cm^3.
3. Define and assign the CFD section.
    a. In the Model Tree, double-click **Sections** and create a new section named **`air`**. Note that a homogeneous fluid section is the only choice available for CFD models. Click **Continue**.
    b. In the **Edit Section** dialog box that appears, select **air** as the **Material** and click **OK**.
    c. Assign the CFD section.
        a. In the Model Tree, expand the **Parts** container. Expand the container for the part named **`BOUNDINGBOX`**.
        b. Double-click **Section Assignments**.
        c. In the prompt area, click **Sets**. In the **Region Selection** dialog box choose **`ALL`**. Click **Continue**.
4. Define an incompressible laminar flow analysis step.
    a. In the Model Tree, double-click **Steps**.
    b. In the **Create Step** dialog box, name the step **`NatConv`** and accept the default procedure type **Flow**. Click **Continue**.
    c. In the step editor, under the **Basic** tab, do the following:
        i. In the **Description** field, enter **`Buoyancy Driven Natural Convection`**.
        ii. In the **Time period** field, enter **`15`** sec.
        iii. Toggle on **Temperature** for the **Energy equation**.

d. In the step editor, under the **Incrementation** tab, accept all default settings. An initial time increment of `0.01` sec and the automatic time incrementation strategy based on a fixed CFL number of `0.45` is used.

e. In the step editor, under the **Solvers** tab, accept the default settings on the **Momentum Equation**, **Pressure Equation** and **Transport Equation** tabs.

f. In the step editor, under the **Turbulence** tab, accept the default setting of **None** under **Turbulence Model**.

5. Define output requests.

   a. In the Model Tree, expand the **Field Output Requests** container.

      Note that a default field output request named **F-Output-1** was automatically created at the time the step was created.

   b. Double-click **F-Output-1**. Note that the output is requested at `20` evenly-spaced time intervals. Accept the default values.

6. Define boundary conditions.

   a. Define a no-slip/no-penetration boundary condition at the bottom surface of the fluid domain.

      i. In the Model Tree, double-click **BCs**.

      ii. Name the boundary condition **`no-slip-bot`** and select **NatConv** as the step.

      iii. Select **Fluid** as the category and **Fluid wall condition** as the type.

      iv. Select **BOUNDINGBOX-1.BOTTOM** as the surface to which the boundary condition will be applied.

         Tip: In the **Region Selection** dialog box, toggle on **Highlight selections in viewport** to identify the region.

      v. Select **No slip** as the condition.

   b. Define a no-slip/no-penetration wall boundary condition at the surface representing the skin of the PCB-component assembly.

      i. In the Model Tree, double-click **BCs**.

      ii. Name the boundary condition **`no-slip-fsi`** and select **NatConv** as the step.

      iii. Select **Fluid** as the category and **Fluid wall condition** as the type.

      iv. Select **BOUNDINGBOX-1.CIRCUIT_BOARD_FSI** as the surface to which the boundary condition will be applied.

      v. Select **No slip** as the condition.

c. Define an outlet boundary condition at the top surface of the CFD computational domain.
   i. In the Model Tree, double-click **BCs**.
   ii. Name the boundary condition `outlet` and select **NatConv** as the step.
   iii. Select **Fluid** as the category and **Fluid inlet/outlet** as the type.
   iv. Select **BOUNDINGBOX-1.TOP** as the surface to which the boundary condition will be applied.
   v. In the **Momentum** tab, toggle on **Specify** and then toggle on **Pressure**.
   vi. Set the pressure to `0`.

7. Define the gravity load.
   a. In the Model Tree, double-click **Loads**.
   b. Name the load `gravity` and select **NatConv** as the step.
   c. Select **Mechanical** as the category and **Gravity** as the type.
   d. Accept the default region to which the load will be applied (**Whole Model**).
   e. Enter a value of `–981` cm/s^2 for **component 2**.

      This specifies acceleration due to gravity in the global *y*-direction.

8. Define initial conditions.
   a. In the Model Tree, double-click **Predefined Fields**.
   b. Name the field `initial temperature` and select **Initial** as the step.
   c. Select **Fluid** as the category and **Fluid thermal energy** as the type.
   d. In the predefined field editor, accept the default region to which the initial temperature field will be applied (**Whole Model**).
   e. Enter a value of `293` K as the initial fluid temperature.

   The default initial condition for velocity is zero so we need not define it explicitly.

9. Define the FSI interaction.
   a. In the Model Tree, double-click **Interactions**.
   b. Name the interaction `fsi`.
   c. Select **NatConv** as the step in which it will be defined and accept **Fluid-Structure Co-simulation boundary** as the type.
   d. Select **BOUNDINGBOX-1.CIRCUIT_BOARD_FSI** as the surface to which the interaction will be applied.

Completing the structural model

In this section, you will complete the structural model. The model is largely complete with the exception of the FSI interaction.

1. In the Model Tree, expand the container under the model **thermal**.
2. Define the FSI interaction.
    a. In the Model Tree, double-click **Interactions**.
    b. Name the interaction `fsi`.
    c. Select **heat transfer** as the step in which it will be defined and **Fluid-Structure Co-simulation boundary** as the type.
    d. Select **CIRCUIT-BOARD-1.FSI_SURF** as the surface to which the interaction will be applied.
3. Modify the step time.
    a. In the Model Tree, expand the **Steps** container.
    b. Double-click **heat transfer**.
    c. Set the time period for the step to `15` sec. This makes it consistent with the CFD model.
4. Modify the output requests.
    a. In the Model Tree, expand the **Field Output Requests** container.

    Note that a default field output request named **F-Output-1** was automatically created at the time the step was created.

    b. Double-click **F-Output-1**.
    c. Select **Evenly spaced time intervals** as the frequency option and enter `20` as the interval size. Accept the default output requests.

The output is requested at approximately the same time intervals in the structural and CFD models. While the structural model writes the output at exact time points, the CFD model writes output at approximate time points.

Creating a co-execution analysis

In order to perform the conjugate heat transfer analysis, the Abaqus/Standard and Abaqus/CFD jobs need to execute together. A co-simulation is performed where the two solvers exchange information at each co-simulation target time. The co-simulation target time is automatically chosen as the minimum of the time increments required by the structural and CFD solvers. In order to facilitate the co-simulation of the two analyses, the co-execution job procedure is used. A co-execution job creates two analysis jobs and runs them simultaneously. It also automatically provides the driver options needed for communication between the two analysis jobs

1. In the Model Tree, expand the **Analysis** container.
2. Double-click **Co-executions** and create a co-execution named **`fsi_cboard`**.
3. In the **Edit Co-execution** dialog box:
    a. Select **thermal** as the first model.
    b. Select **fluid** as the second model.
    c. Click **OK**.

In the Model Tree, expand the **Co-executions** container and then expand the **fsi_cboard** container. Expand the **Jobs** container under **fsi_cboard**.

Note that two analyses jobs have been created – one representing the Abaqus/Standard structural model and the other representing the Abaqus/CFD model.

Running the co-simulation analysis

Launch the co-execution job from within Abaqus/CAE.

1. Click mouse button 3 on the co-execution job **fsi_cboard**.
2. From the menu that appears, select **Submit**.
   This launches the co-execution job. Both the Abaqus/Standard and Abaqus/CFD jobs will be launched.

Monitoring the co-execution analysis

While the co-execution is running, you can monitor its progress.

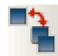
1. Click mouse-button 3 on the CFD analysis job name and select **Monitor** from the menu that appears.
2. The job monitor appears. Note that time incrementation information, divergence (RMS) and kinetic energy is updated every time increment.

S T O P!

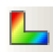
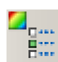
Continue with the remainder of this workshop after the completion of the next lecture.

Viewing the results

Once the co-execution completes, do the following:

1. Click mouse button 3 on the co-execution named **fsi_cboard**.
2. From the menu that appears, select **Results**.
   The output database files `fsi_cboard-thermal.odb` and `fsi_cboard-fluid.odb` are opened simultaneously in the Visualization module and are overlaid in the viewport.
3. Toggle off the overlay plot option  in the toolbox.

Make the output database file `fsi_cboard-fluid.odb` current.

1. Create a temperature contour plot for the surrounding air on cut-planes perpendicular to the *x*- and *y*-axes and cutting the component through the center.
   a. In the toolbox, click  to create a contour plot (alternatively select **Plot→Contours→On Undeformed Shape**).
   b. From the main menu bar, select **Result → Field Output**. In the **Field Output** dialog box, select **TEMP** as the output variable and click **OK**.
      Tip: You may also select the variable from the **Field Output** toolbar.
   c. In the toolbox, click  to open the **Common Plot Options** dialog box. Toggle on **Feature edges** for the visible edges and click **OK**. This turns off the mesh feature lines in the model.
   d. In the toolbox, click  to open the **Contour Plot Options** dialog box. Toggle on **Continous** under **Contour Intervals** and click **OK**. This creates a smooth temperature contour plot.
   e. Create two cut-planes.
      i. For the main menu bar, select **Tools→View Cut→Manager**.
      ii. In the **View Cut Manager**, toggle on **Allow for multiple cuts**.
          This will enable multiple view-cuts.
      iii. Select the **X-Plane** and enter `43.18` as the **Position**.
      iv. Select the **Y-Plane** and enter `5.8` as the **Position**.
      v. Toggle off the *above-cut* options (under the column labeled ) next to **X-Plane** and **Y-Plane**. This only leaves only the *on-cut* option enabled.
   f. The temperature contour on the two cut-planes appears as shown in Figure W2–4.

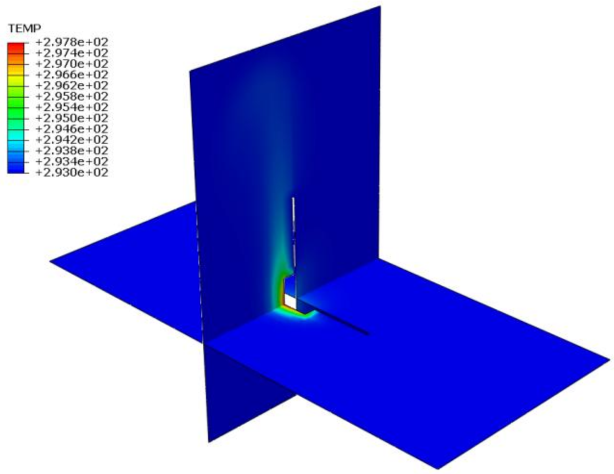


Figure W2–4 Temperature contour plot on two cut-planes: surrounding air

2. Create a pressure contour plot for the surrounding air on the two cut-planes.
    a. Using the **Field Output** toolbar, select **PRESSURE** as the output variable to plot.

        The pressure contour plot on the activate cut-planes appears as shown in Figure W2–5.

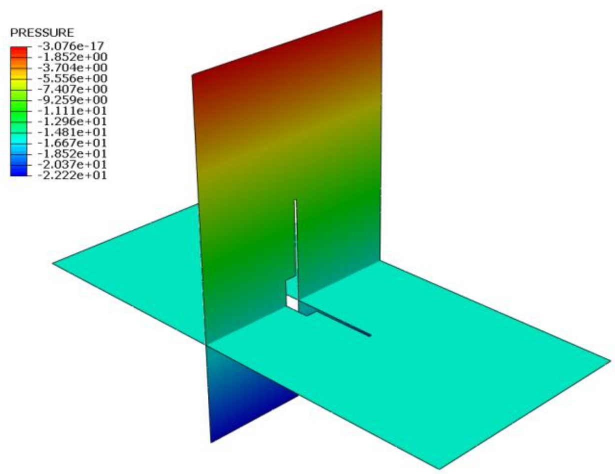


Figure W2–5 Pressure contour plot on two cut-planes: surrounding air

3. Create a velocity vector plot for the surrounding air on the two cut-planes.
    a. Using the **Field Output** toolbar, select **V** as the output variable to plot.

        This creates a velocity contour plot on the cut-planes.

    b. In the toolbox, click  to create a vector plot (alternatively select **Plot→Symbols→On Undeformed Shape**).   The plot appears as shown in Figure W2–6.

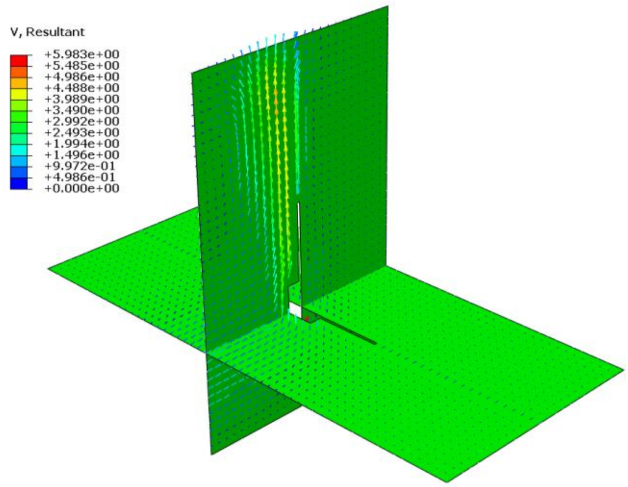


Figure W2–6 Velocity vector plot on two cut-planes: surrounding air

Make the output database file **`fsi_cboard-thermal.odb`** current.

1. Create a temperature contour plot for the PCB-component assembly, as shown in Figure W2–7. Also, activate the two cut-planes used earlier (perpendicular to the *x*- and *y*-axes and cutting the component through the center).

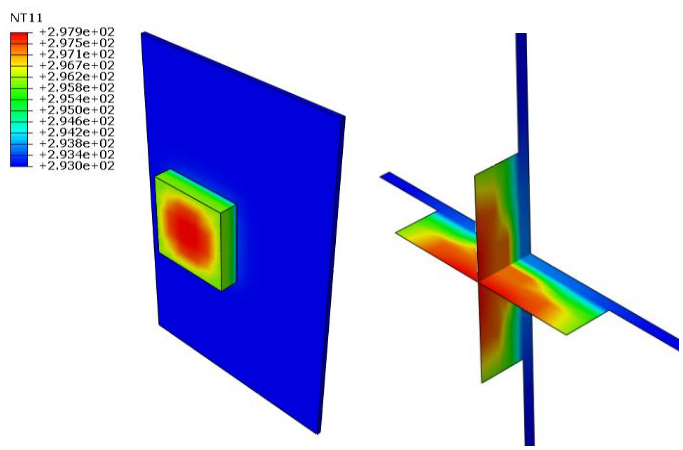


Figure W2–7 Temperature contours: PCB-component assembly

Note: A script that creates the complete model described in these instructions is available for your convenience. Run this script if you encounter difficulties following the instructions outlined here or if you wish to check your work. The script is named

`ws_cfd_cboard_answer.py`

and is available using the Abaqus fetch utility.

Notes

Notes