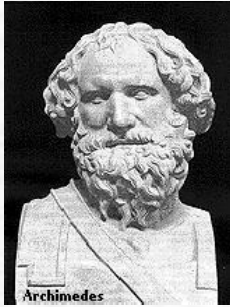# Introduction to Computational Fluid Dynamics (CFD)

Zhaoyuan Wang, Maysam Mousaviraad, Shanti Bhushan, Tao Xing, Timur K. Dogan, and Fred Stern

IIHR—Hydroscience & Engineering
C. Maxwell Stanley Hydraulics Laboratory
The University of Iowa

ME:5160 Intermediate Mechanics of Fluids
http://css.engineering.uiowa.edu/~me_160/

Aug. 30, 2024

Faces of Fluid Mechanics

Archimedes
(C. 287-212 BC)

Newton
(1642-1727)

Leibniz
(1646-1716)

Bernoulli
(1667-1748)
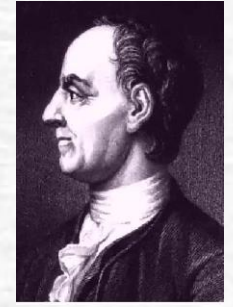
Euler
(1707-1783)

Navier
(1785-1836)

Stokes
(1819-1903)

Reynolds
(1842-1912)

Prandtl
(1875-1953)

Taylor
(1886-1975)

Kolmogorov
(1903-1987)

Outline

1. What, why and where of CFD?
2. Modeling
3. Numerical methods
4. Types of CFD codes
5. ANSYS CFD Interface
6. CFD Process
7. Example of CFD Process
8. ME:5160 CFD Labs

What is CFD?

- CFD is the simulation of fluids engineering systems using modeling (mathematical physical problem formulation) and numerical methods (discretization methods, solvers, numerical parameters, and grid generations, etc.)

- Historically only Analytical Fluid Dynamics (AFD) and Experimental Fluid Dynamics (EFD).

- CFD made possible by the advent of digital computer and advancing with improvements of computer resources (500 flops, 1947$\rightarrow$20 teraflops (10^{12}), 2003 $\rightarrow$1.3 pentaflops (10^{15}), 2009 $\rightarrow$200 petaflops, Oak Ridge National Laboratory,2019 $\rightarrow$513 petaflops, Japan, 2020 $\rightarrow$1,102 petaflops, USA, 2022).
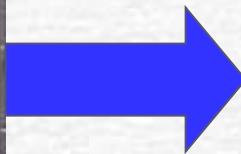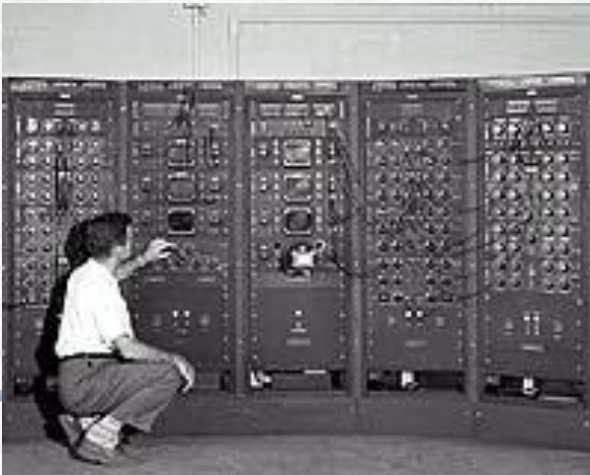
Supercomputer FRONTIER, DOE/SC/Oak Ridge National Laboratory, Peak Compute: 1,679.82 (PFlop/s), TOP500: 1st, June 2022-2024

Why use CFD?

- Analysis and Design
  1. Simulation-based design instead of "build & test"
     - More cost effective and more rapid than EFD
     - CFD provides high-fidelity database for diagnosing flow field
     - Provides guidance for planning experiments
  2. Simulation of physical fluid phenomena that are difficult for experiments
     - Full scale simulations (e.g., ships and airplanes)
     - Environmental effects (wind, weather, etc.)
     - Hazards (e.g., explosions, radiation, pollution)
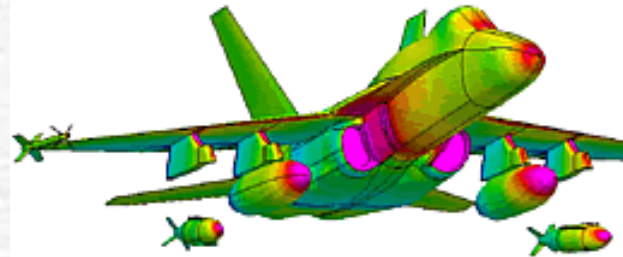   - Physics (e.g., planetary boundary layer, stellar evolution)

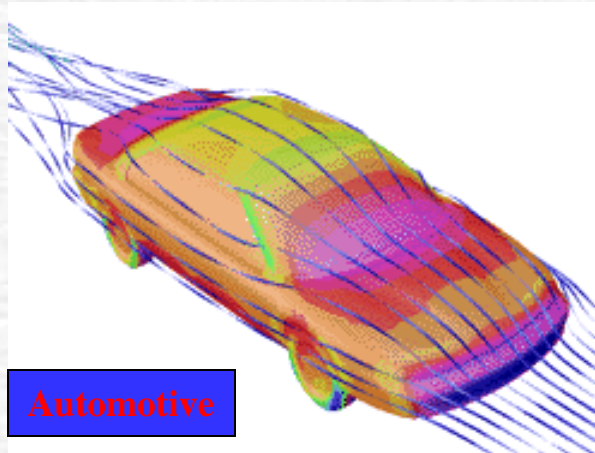- Knowledge and exploration of flow physics

Where is CFD used?

- Where is CFD used?
  - ***Aerospace***
  - ***Automotive***
  - ***Biomedical***
  - Chemical Processing
  - HVAC
  - Hydraulics
  - Marine
  - Oil & Gas
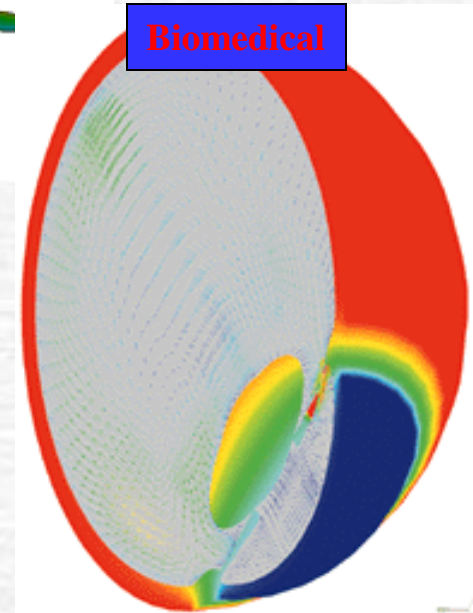 - Power Generation
  - Sports
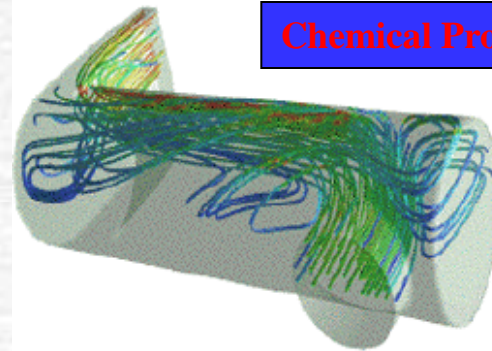
Aerospace

F18 Store Separation

Biomedical

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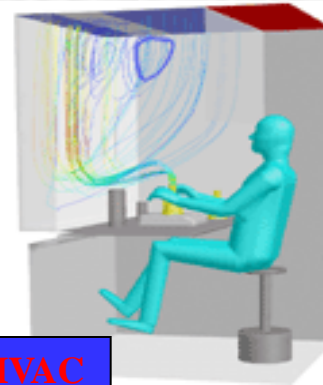
Automotive

6

Where is CFD used?

- Where is CFD used?
  - Aerospacee
  - Automotive
  - Biomedical
  - **Chemical Processing**
  - **HVAC**
  - **Hydraulics**
  - Marine
  - Oil & Gas
  - Power Generation
  - Sports

Chemical Processing

Polymerization reactor vessel - prediction of flow separation and residence time effects.
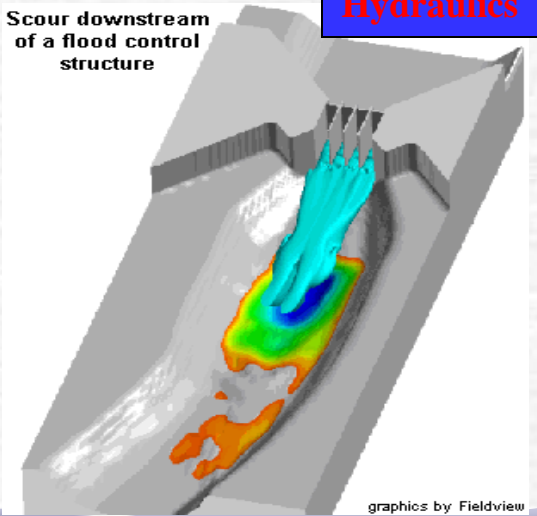

HVAC

Streamlines for workstation ventilation

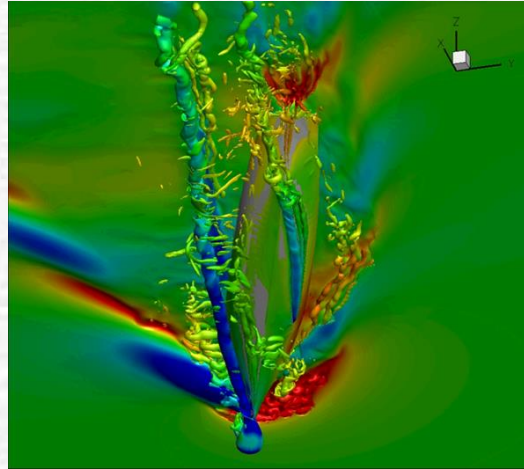Scour downstream of a flood control structure

Hydraulics

graphics by Fieldview

Where is CFD used?

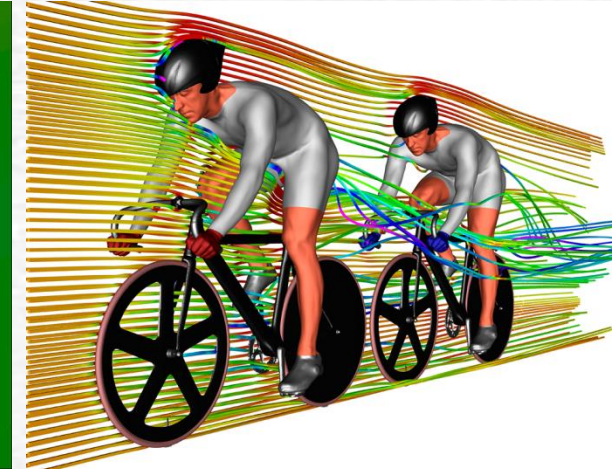- Where is CFD used?
  - Aerospace
  - Automotive
  - Biomedical
  - Chemical Processing
  - HVAC
  - Hydraulics
  - *Marine*
  - *Oil & Gas*
  - *Power Generation*
  - *Sports*

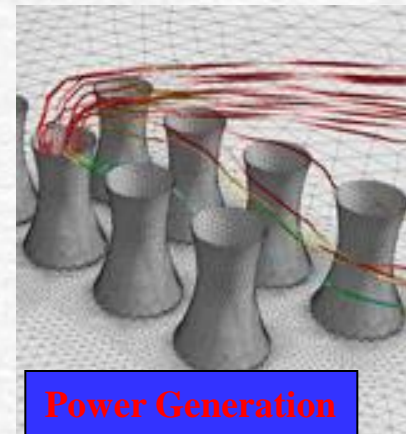**Marine**

Sports

http://gallery.ensight.com/keyword/sport/1/701168838_KVnHn#!i=701168838&k=KVnHn

Oil & Gas

Flow of lubricating mud over drill bit

Power Generation

Flow around cooling towers

8

Modeling

- **Modeling is** the mathematical physics problem formulation in terms of a continuous initial boundary value problem (IBVP)
- IBVP is in the form of Partial Differential Equations (PDEs) with appropriate boundary conditions and initial conditions.
- **Modeling includes:**
    1. Geometry and domain
    2. Coordinates
    3. Governing equations
    4. Flow conditions
    5. Initial and boundary conditions
    6. Selection of models for different applications

Modeling (geometry and domain)

- **Simple geometries** can be easily created by few geometric parameters (e.g. circular pipe)

- **Complex geometries** must be created by the partial differential equations or importing the database of the geometry(e.g. airfoil) into commercial software

- **Domain**: size and shape

- Typical approaches

  - Geometry approximation

  - CAD/CAE integration: use of industry standards such as Parasolid, ACIS, STEP, or IGES, etc.

  - The three coordinates: Cartesian system (x,y,z), cylindrical system (r, θ, z), and spherical system(r, θ, Φ) should be appropriately chosen for a better resolution of the geometry (e.g. cylindrical for circular pipe).

Modeling (coordinates)

Cartesian
(x,y,z)

Cylindrical
(r,θ,z)

Spherical
(r,θ,ϕ)

General Curvilinear Coordinates

$\hat{u}_1 \times \hat{u}_2 = \hat{u}_3$

General orthogonal Coordinates

11
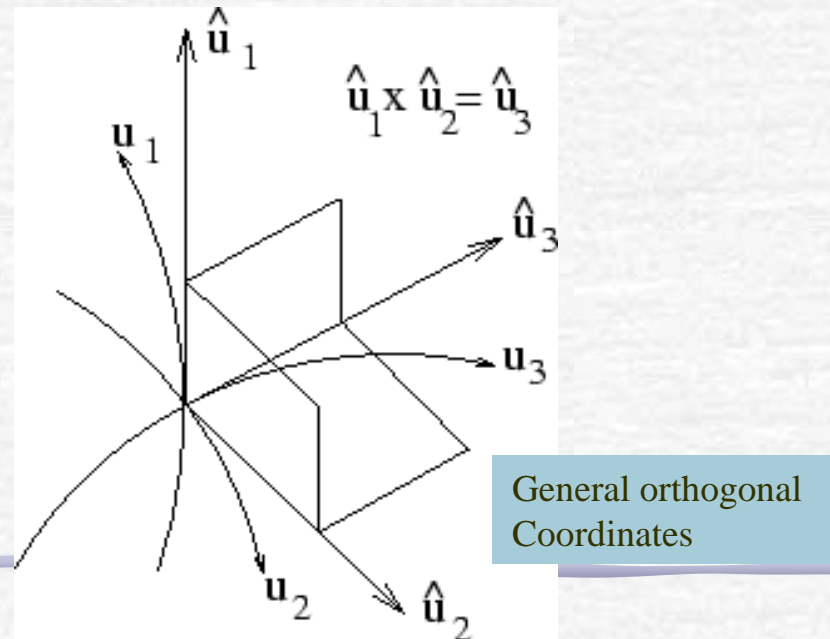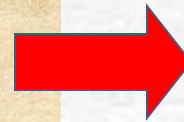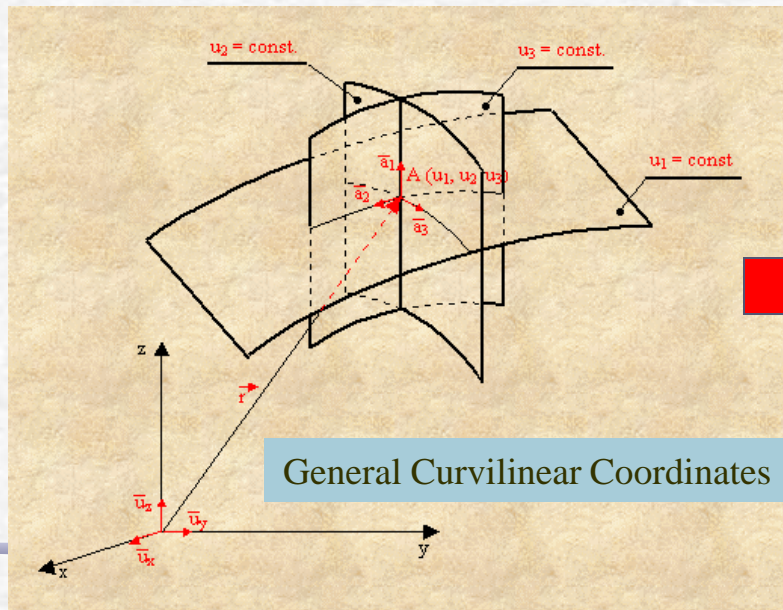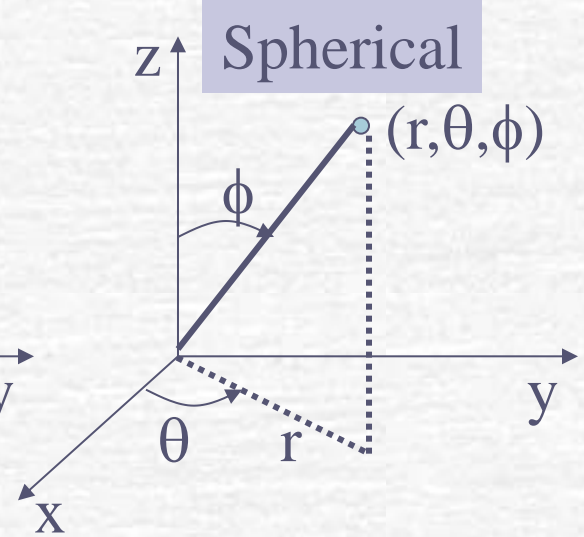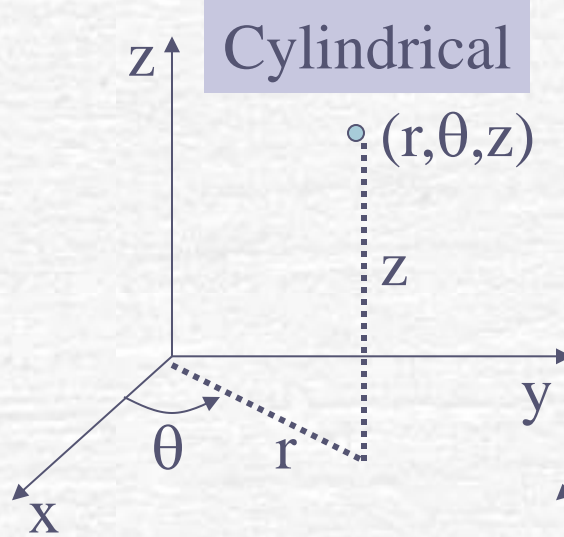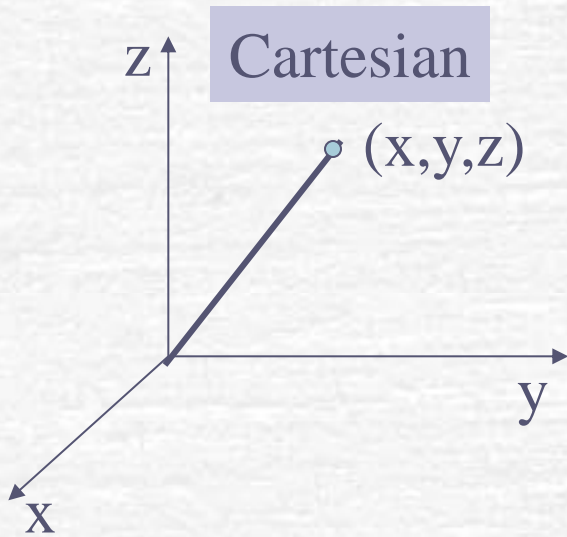
Modeling (governing equations)

- Navier-Stokes equations (3D in Cartesian coordinates)

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} + \rho w \frac{\partial u}{\partial z} = -\frac{\partial \hat{p}}{\partial x} + \mu \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right]$$

$$\rho \frac{\partial v}{\partial t} + \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} + \rho w \frac{\partial v}{\partial z} = -\frac{\partial \hat{p}}{\partial y} + \mu \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right]$$

$$\rho \frac{\partial w}{\partial t} + \rho u \frac{\partial w}{\partial x} + \rho v \frac{\partial w}{\partial y} + \rho w \frac{\partial w}{\partial z} = -\frac{\partial \hat{p}}{\partial z} + \mu \left[\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right]$$

Local acceleration   Convection   Piezometric pressure gradient   Viscous terms

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z} = 0$$ Continuity equation

$$p = \rho R T$$ Equation of state

$$R \frac{D^2 R}{Dt^2} + \frac{3}{2} \left(\frac{DR}{Dt} \right)^2 = \frac{p_v - p}{\rho_L}$$ Rayleigh Equation

12

Modeling (flow conditions)

• Based on the physics of the fluids phenomena, CFD can be distinguished into different categories using different criteria

- Viscous vs. inviscid (Re: Reynolds No.)

- External flow or internal flow (wall bounded or not)

- Turbulent vs. laminar (Re)

- Incompressible vs. compressible (Ma: Mach No.)

- Single- vs. multi-phase (Ca: Cavitation No.)

- Thermal/density effects (Pr: Prandtl No., γ, Gr: Grashof No., Ec)

- Free-surface flow (Fr) and surface tension (We)

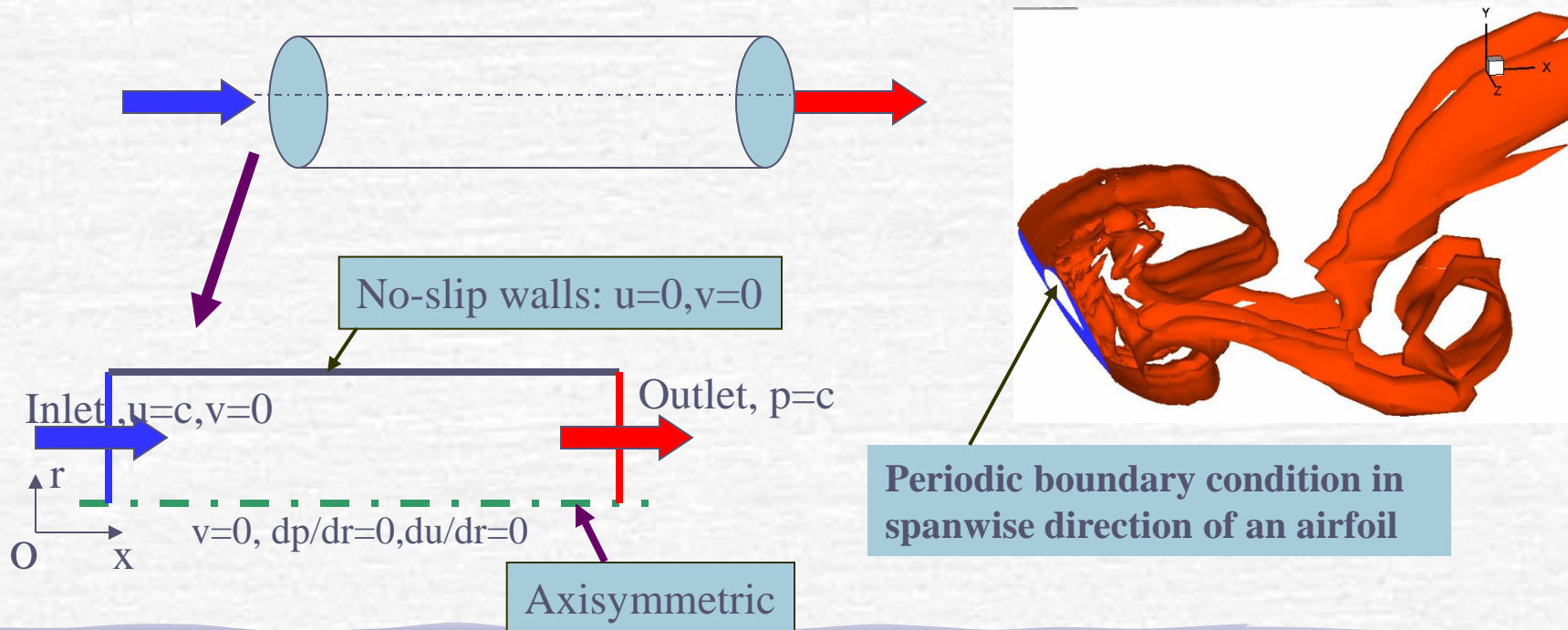- Chemical reactions and combustion (Pe, Da)

- etc…

Modeling (initial conditions)

- Initial conditions (ICS, steady/unsteady flows)
  - ICs should not affect final results and only affect convergence path, i.e. number of iterations (steady) or time steps (unsteady) need to reach converged solutions.
  - More reasonable guess can speed up the convergence
  - For complicated unsteady flow problems, CFD codes are usually run in the steady mode for a few iterations for getting a better initial conditions

Modeling(boundary conditions)

- **Boundary conditions:** No-slip or slip-free on walls, periodic, inlet (velocity inlet, mass flow rate, constant pressure, etc.), outlet (constant pressure, velocity convective, numerical beach, zero-gradient), and non-reflecting (for compressible flows, such as acoustics), etc.

No-slip walls: $u=0, v=0$

Inlet, $u=c, v=0$

Outlet, $p=c$

r

O x

$v=0$, $dp/dr=0$, $du/dr=0$

Axisymmetric

Periodic boundary condition in spanwise direction of an airfoil

Modeling (selection of models)

- CFD codes typically designed for solving certain fluid phenomenon by applying different models

  - Viscous vs. inviscid (Re)

  - Turbulent vs. laminar (Re, Turbulent models)

  - Incompressible vs. compressible (Ma, equation of state)

  - Single- vs. multi-phase (Ca, cavitation model, two-fluid model)

  - Thermal/density effects and energy equation
    (Pr, γ, Gr, Ec, conservation of energy)

  - Free-surface flow (Fr, level-set & surface tracking model) and surface tension (We, bubble dynamic model)

  - Chemical reactions and combustion (Chemical reaction model)

 - etc...

Modeling (Turbulence and free surface models)

- Turbulent flows at high Re usually involve both large and small scale vortical structures and very thin turbulent boundary layer (BL) near the wall

- **Turbulent models**:

  - DNS: most accurately solve NS equations, but too expensive for turbulent flows

  - RANS: predict mean flow structures, efficient inside BL but excessive diffusion in the separated region.

  - LES: accurate in separation region and unaffordable for resolving BL

  - DES: RANS inside BL, LES in separated regions.

- **Free-surface models**:

  - Surface-tracking method: mesh moving to capture free surface, limited to small and medium wave slopes

  - Single/two phase level-set and volume-of-fluid (VOF) methods: mesh fixed and level-set/VOF functions used to capture the gas/liquid interface, capable of studying steep or breaking waves.

Modeling (examples)
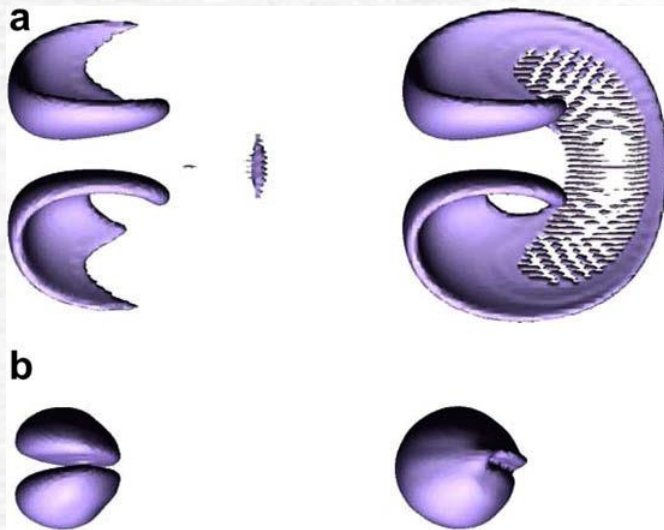
Deformation of a sphere.(a)maximum stretching; (b) recovered shape. Left: LS; right: VOF.
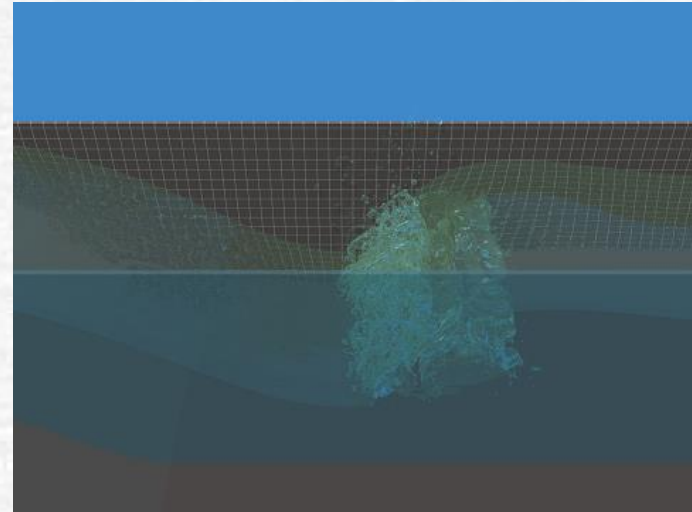
Wave breaking in bump flow simulation: 2.2 billion grid points

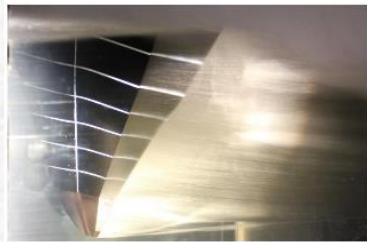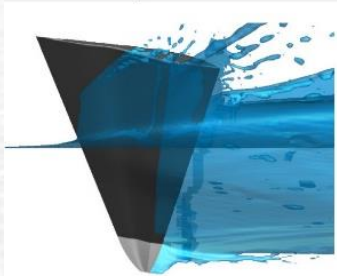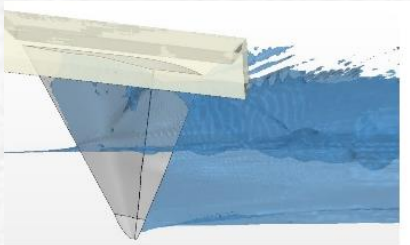Movie

Wedge flow simulation, 1 billion grid points

Movie

Stokes wave breaking: 3.2-12 Billion Grid Points

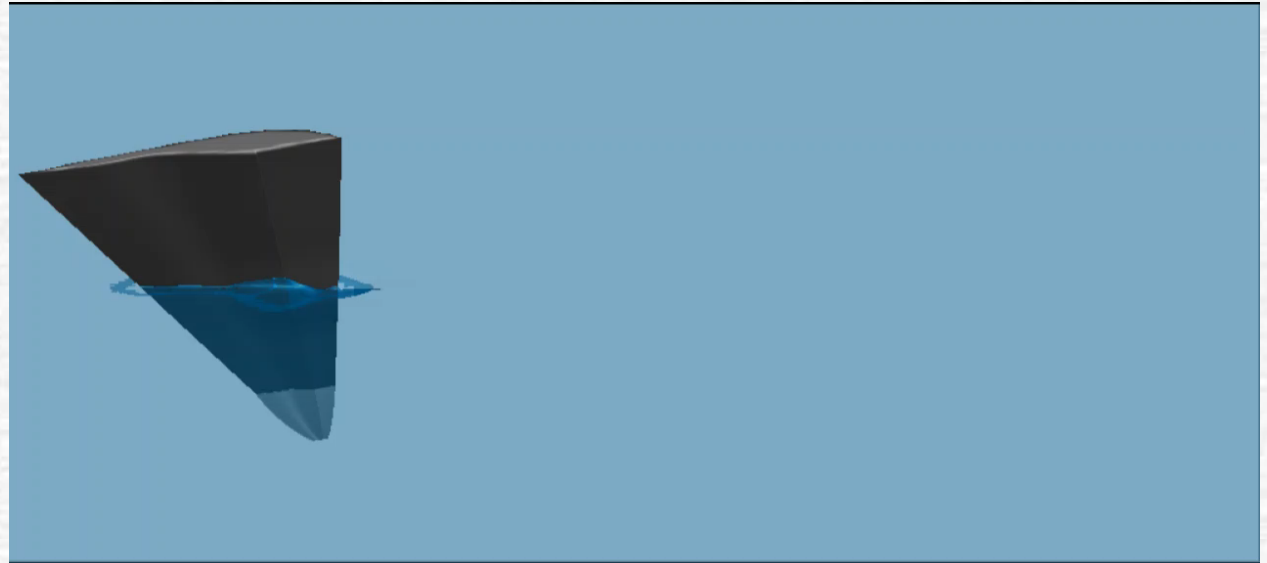Movie

Modeling (examples)

<u>Super-Cavitating Surface-Piercing hydrofoil</u>

5⁰

Modeling (examples, cont'd)

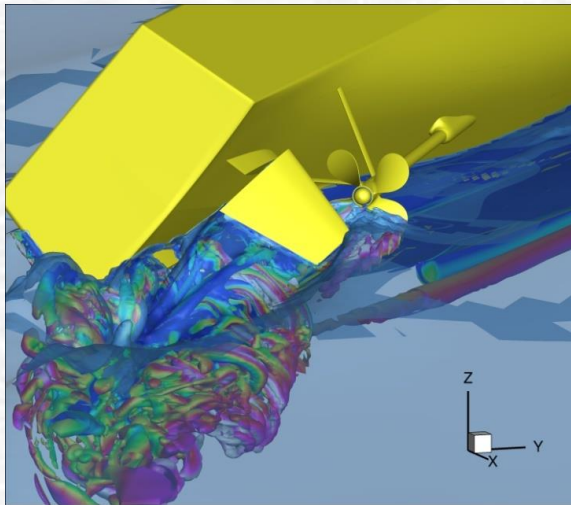Simulation of Numerette Planing Hull
Movie

Air flow for ONR Tumblehome in Planar motion mechanism (PMM) maneuvers
Movie

Broaching of ONR Tumblehome with rotating propellers
Movie
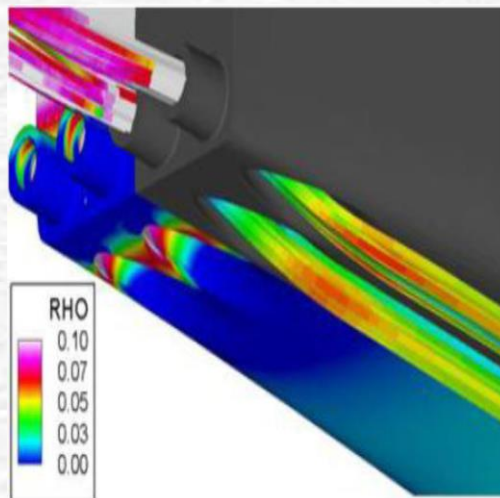
Waterjet flow modeling for JHSS and Delft catamaran
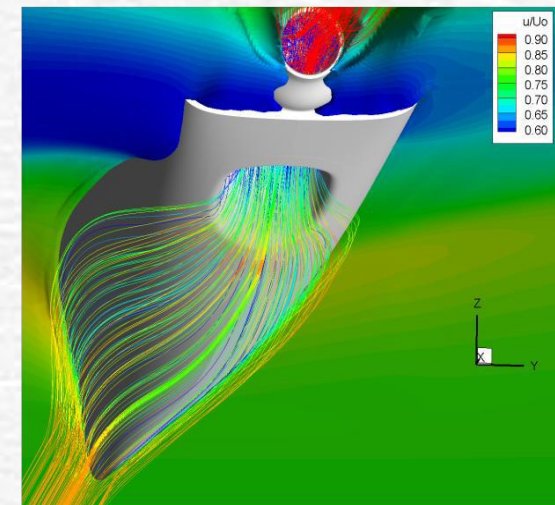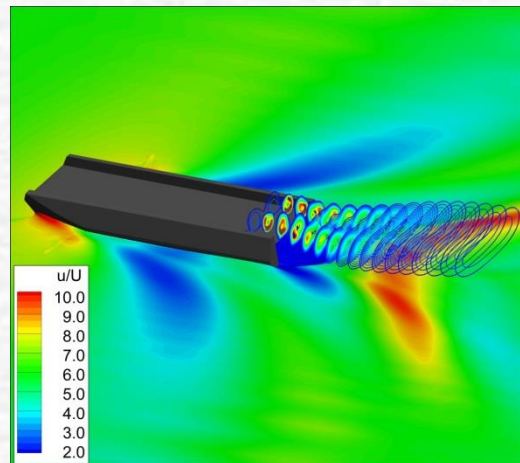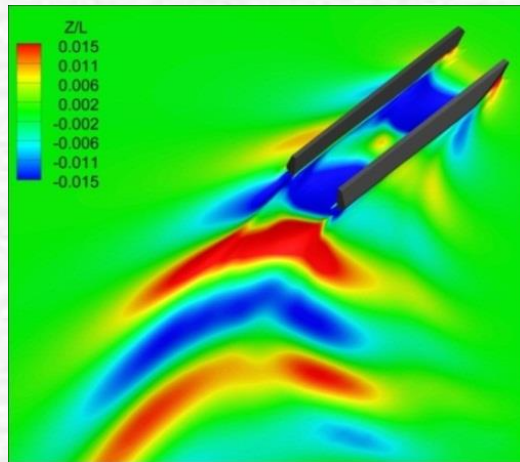
T-Craft (SES/ACV) turning circle in calm water with water jet propulsion (top) and straight ahead with air-fan propulsion (bottom)

Movie
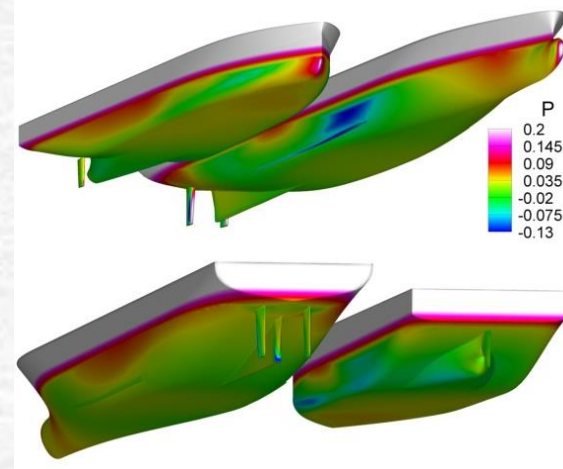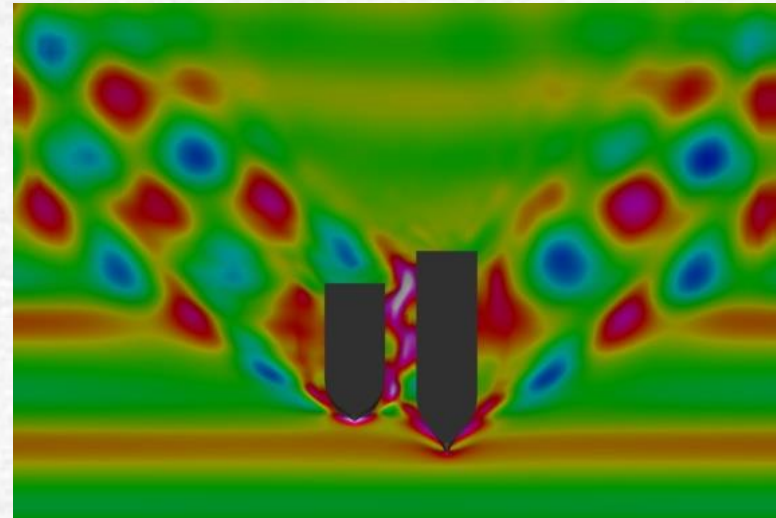
Regular head wave simulation for side by side ship-ship interactions

Movie

21
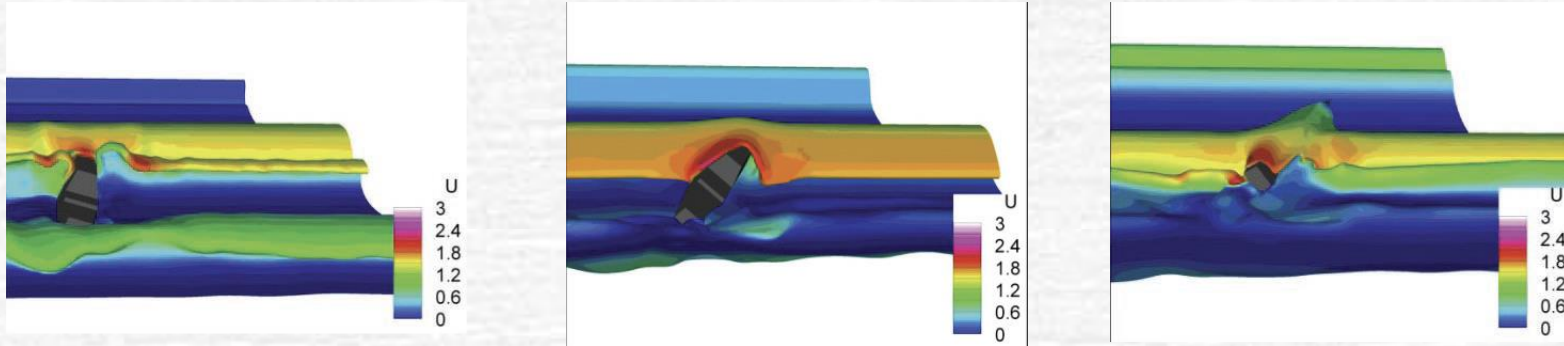
Movie

Ship in three-sisters rogue (freak) waves
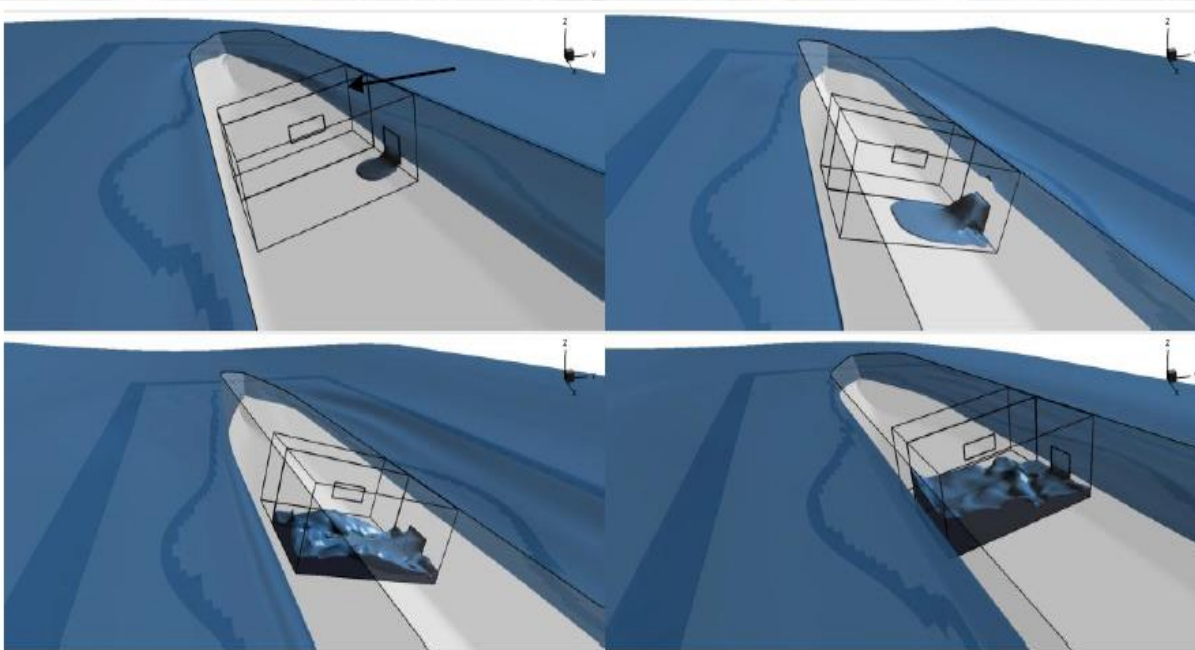
Damaged stability for SSRC cruiser with two-room compartment in beam waves

Movie

Vortical Structures and Instability Analysis
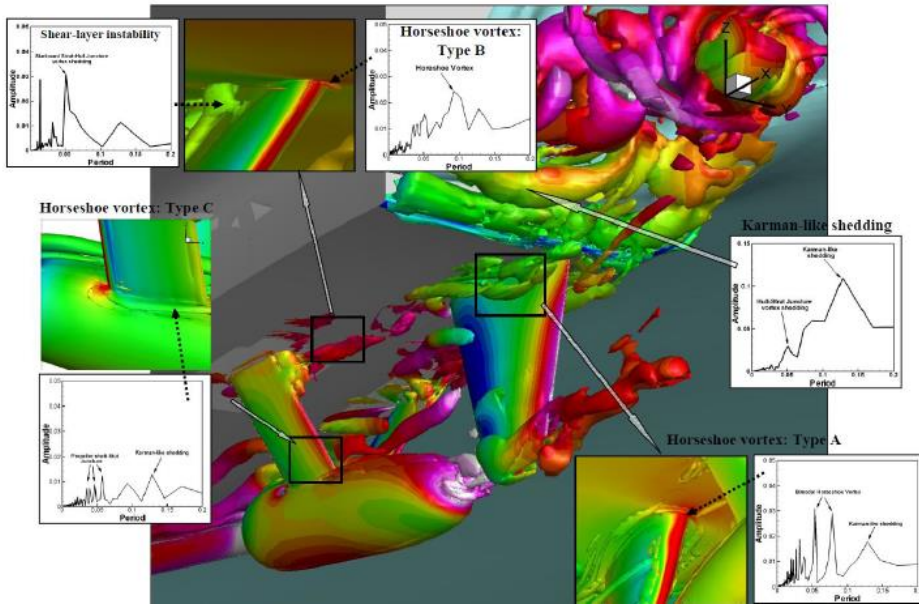
Fully appended Athena DES Computation

Re=2.9×10^8, Fr=0.25

Isosurface of Q=300 colored using piezometric pressure

- Karman-like shedding from Transom Corner
- Horse-shoe vortices from hull-rudder (Case A) and strut-hull (Case B) junction flow.
- Shear layer instability at hull-strut intersection

DTMB 5415 at β=20° DES Computation

Re=4.85×10^6,Fr=0.28

Isosurface of Q=300 colored using piezometric pressure

- The sonar dome (SD$_{TV}$) and bilge keel (BK$_{TV}$) vortices exhibits helical instability breakdown.
- Shear-layer instabilities: port bow (B$_{SL1}$, B$_{SL2}$) and fore-body keel (K$_{SL}$).
- Karman-like instabilities on port side bow (B$_K$).
- Wave breaking vortices on port (FS$_{BW1}$) and starboard (FS$_{BW2}$). Latter exhibits horse shoe type instability.

Movie
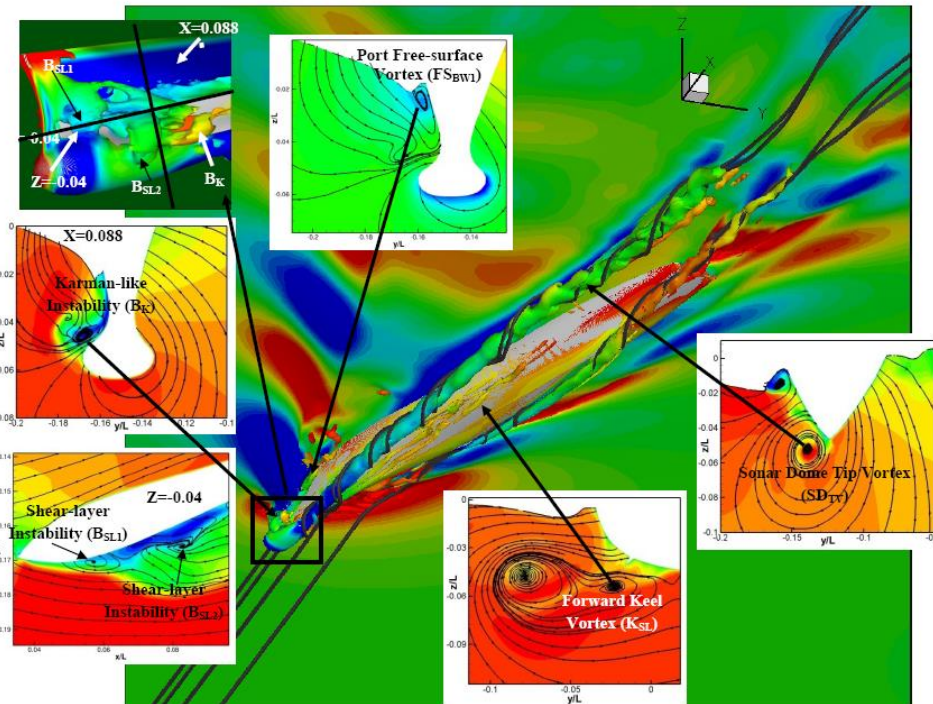
Modeling (examples, cont'd)
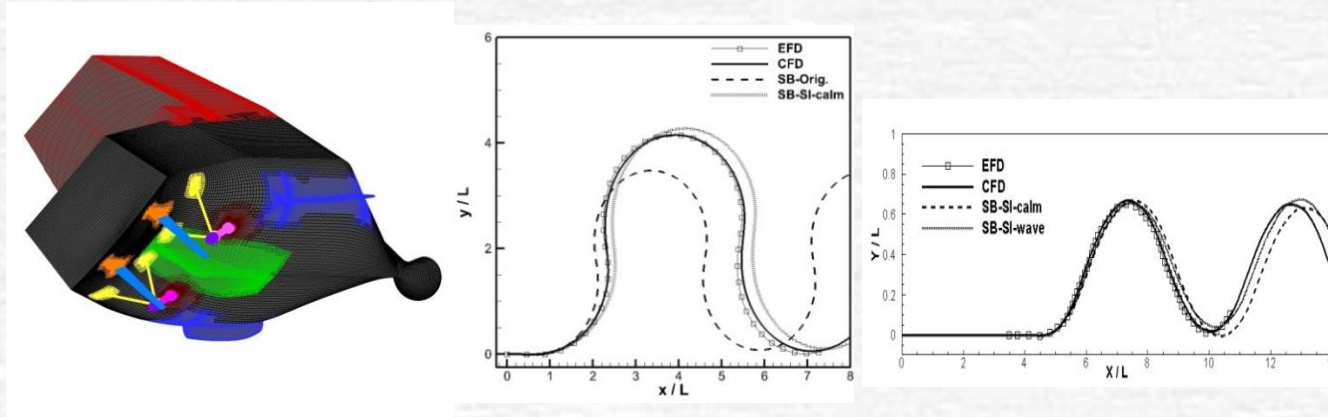
Movie (CFD)

Movie (EFD
at Iowa wave basin)

CFD simulations to improve system identification (SI) technique
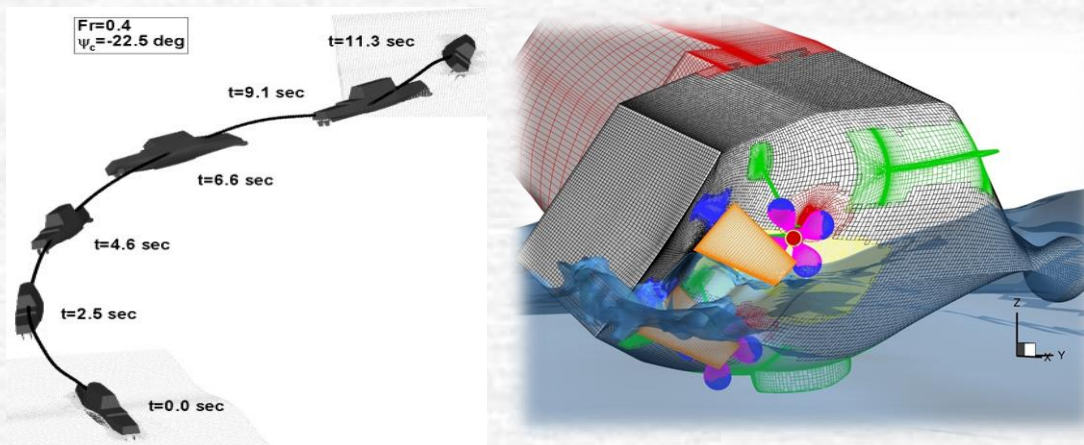
Broaching simulation of free running ONR Tumblehome

Movie (CFD)

Movie (EFD)

Stepped planing hulls
CFDShip-Iowa V5.5

Experiment
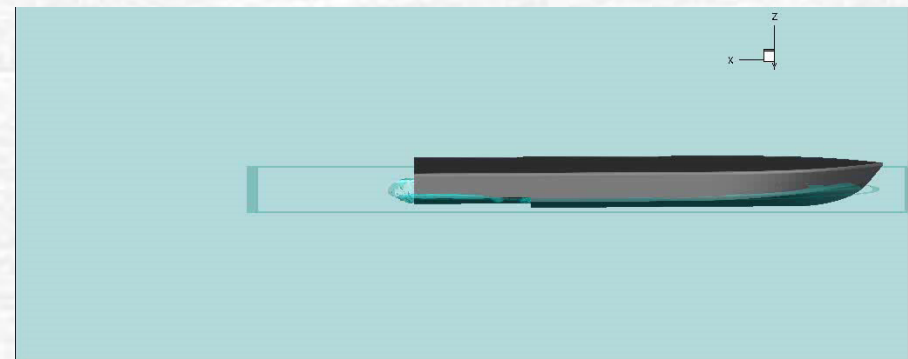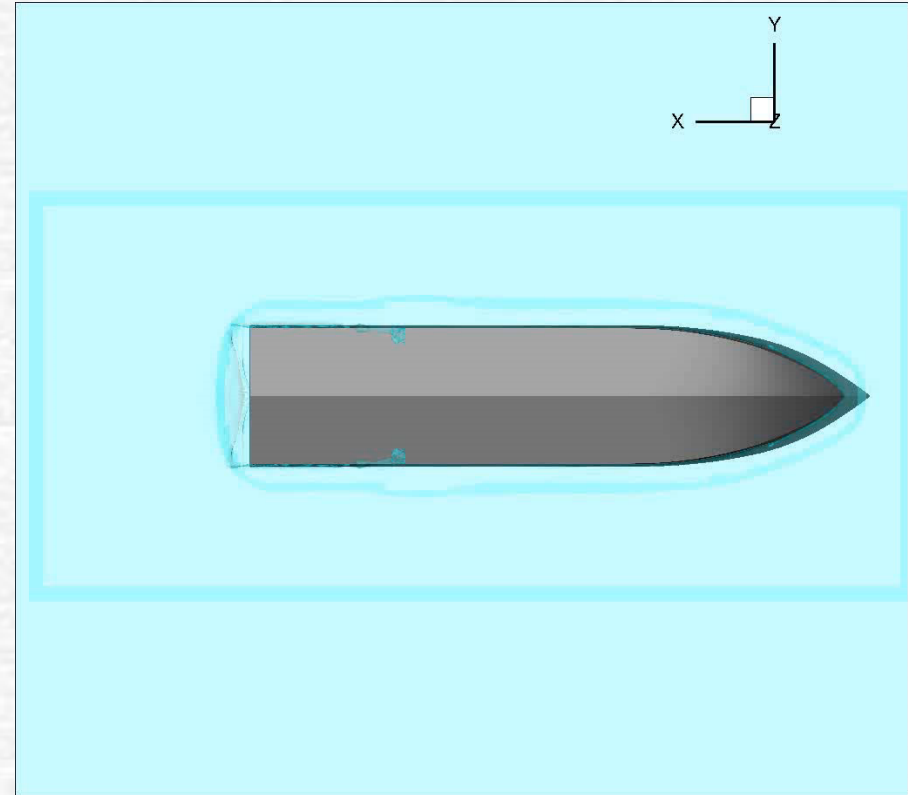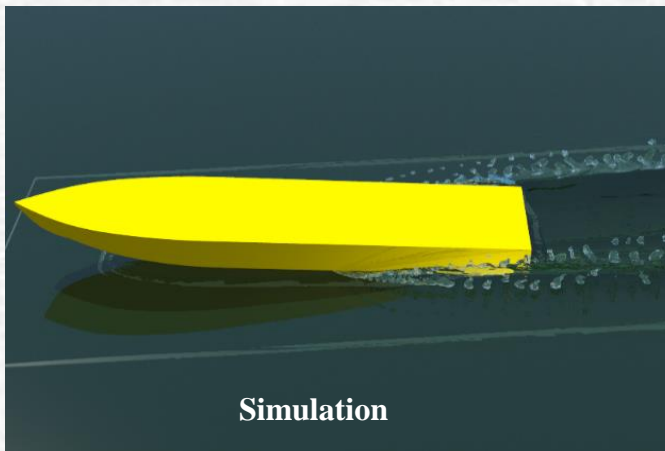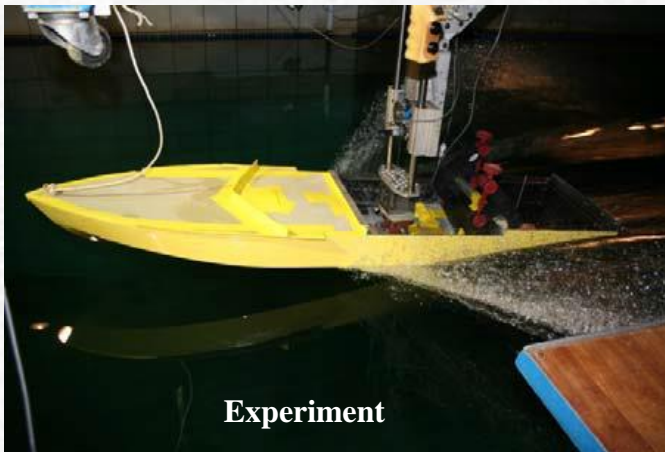
Simulation

6DoF of GPPH

Turning circle maneuvering in calm water
25deg steering angle, port side

American Boat and Yacht Council,
avoidance line test
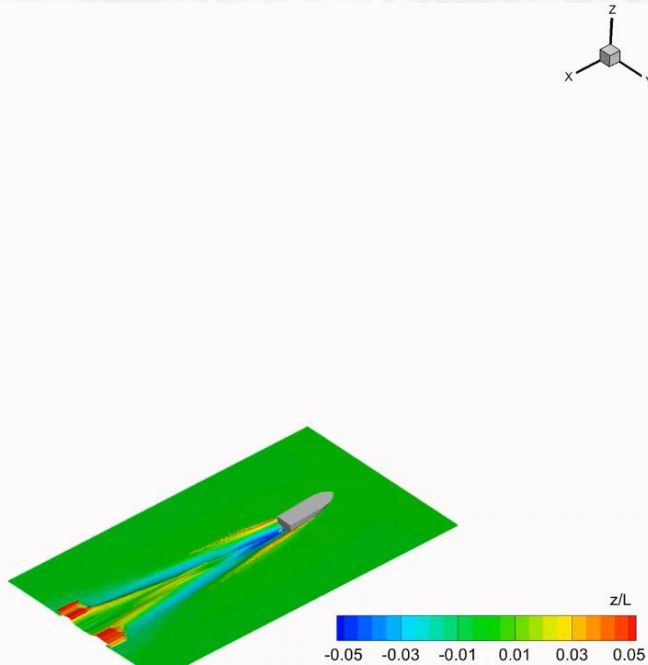
CFDShip-Iowa V5.5
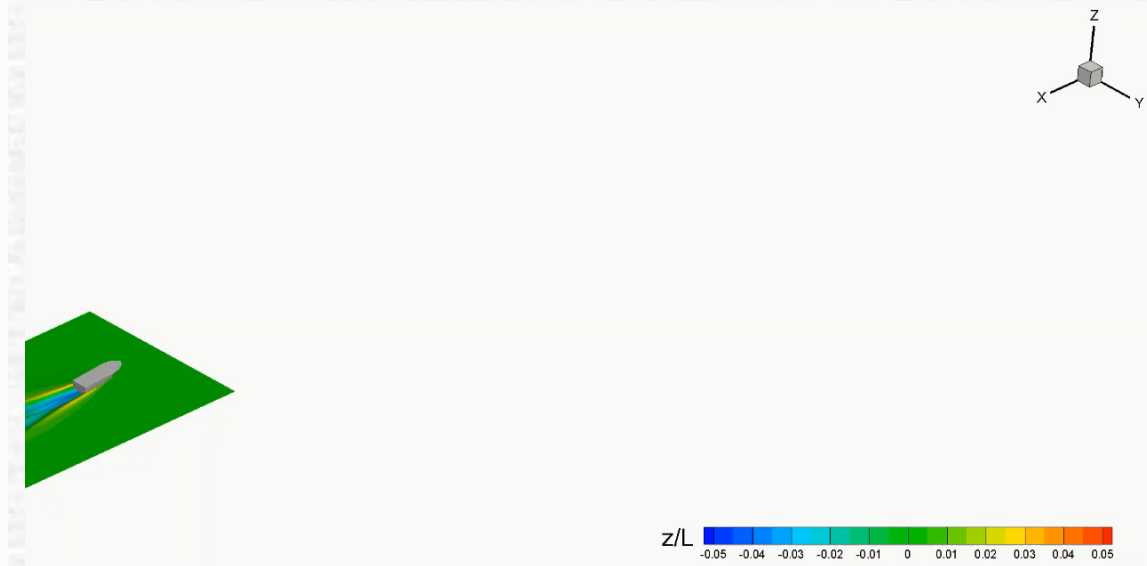
CFDShip-Iowa V4.5
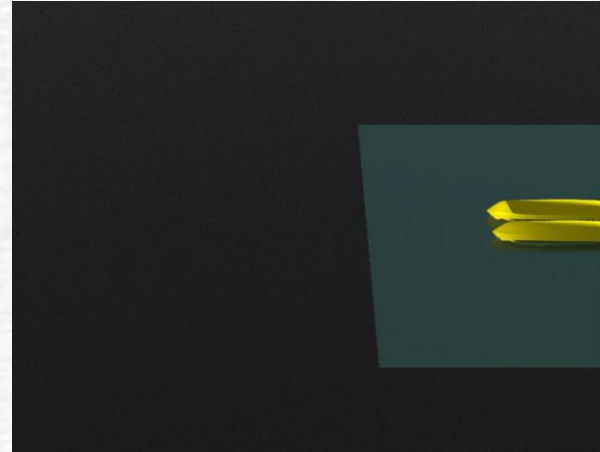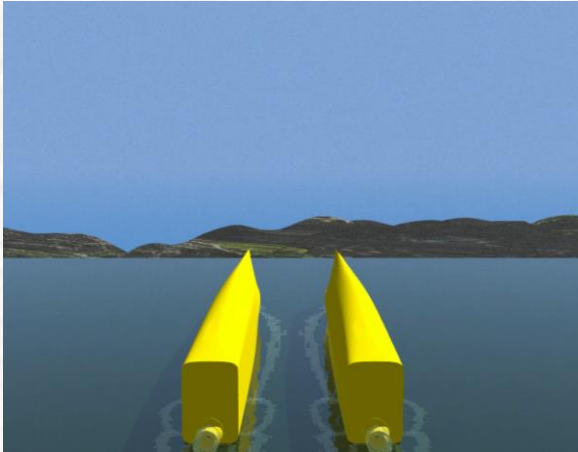
Simulation of DC using CFDShip-V5.5 for Fr=0.6.
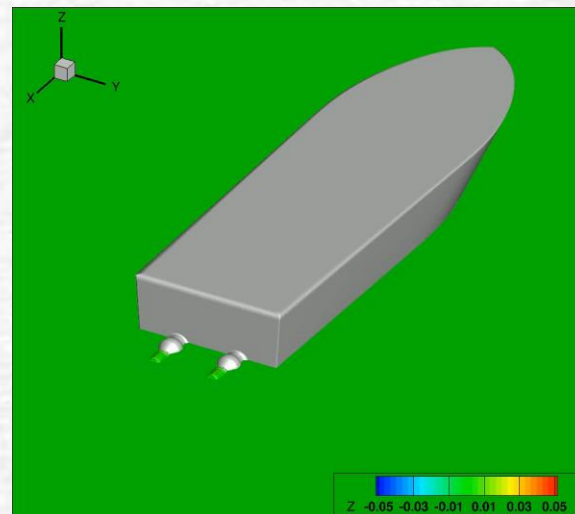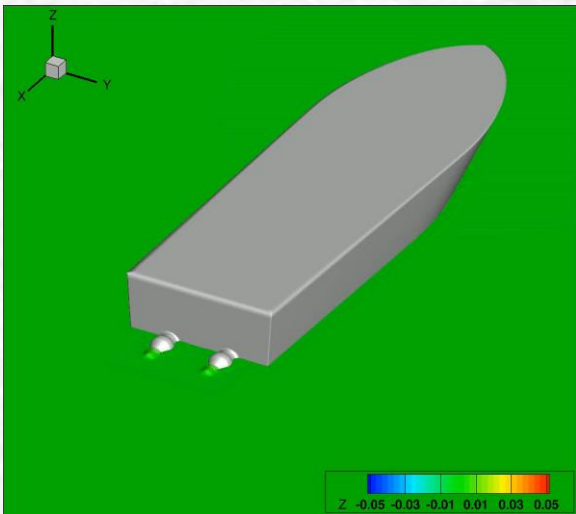
Simulation of GPPH using CFDShip-V4.5 (left) and V5.5 (right).

Numerical methods

- The continuous Initial Boundary Value Problems (IBVPs) are discretized into algebraic equations using numerical methods. Assemble the system of algebraic equations and solve the system to get approximate solutions

- Numerical methods include:
  1. Discretization methods
  2. Solvers and numerical parameters
  3. Grid generation and transformation
  4. High Performance Computation (HPC) and post-processing

Discretization methods

- Finite difference methods (straightforward to apply, usually for regular grid) and finite volumes and finite element methods (usually for irregular meshes)
- Each type of methods above yields the same solution if the grid is fine enough. However, some methods are more suitable to some cases than others
- Finite difference methods for spatial derivatives with different order of accuracies can be derived using Taylor expansions, such as 2nd order upwind scheme, central differences schemes, etc.
- Higher order numerical methods usually predict higher order of accuracy for CFD, but more likely unstable due to less numerical dissipation
- Temporal derivatives can be integrated either by the explicit method (Euler, Runge-Kutta, etc.) or implicit method (e.g. Beam-Warming method)

Discretization methods (Cont'd)

- **Explicit methods** can be easily applied but yield conditionally stable Finite Different Equations (FDEs), which are restricted by the time step; **Implicit methods** are unconditionally stable but need efforts on efficiency.
- Usually, higher-order temporal discretization is used when the spatial discretization is also of higher order.
- **Stability**: A discretization method is said to be stable if it does not magnify the errors that appear in the course of numerical solution process.
- **Pre-conditioning** method is used when the matrix of the linear algebraic system is ill-posed, such as multi-phase flows, flows with a broad range of Mach numbers, etc.
- **Selection of discretization methods** should consider efficiency, accuracy and special requirements, such as shock wave tracking.

Discretization methods (example)

- 2D incompressible laminar flow boundary layer

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

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial}{\partial x}\left(\frac{p}{e}\right) + \mu\frac{\partial^2 u}{\partial y^2}$$
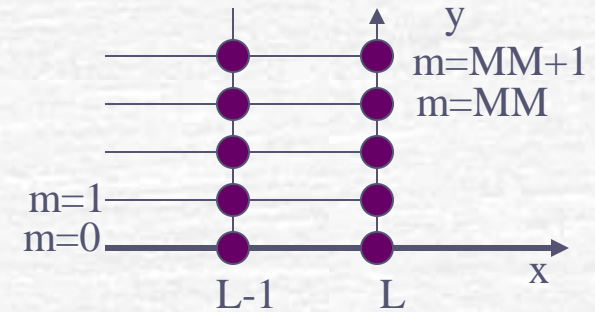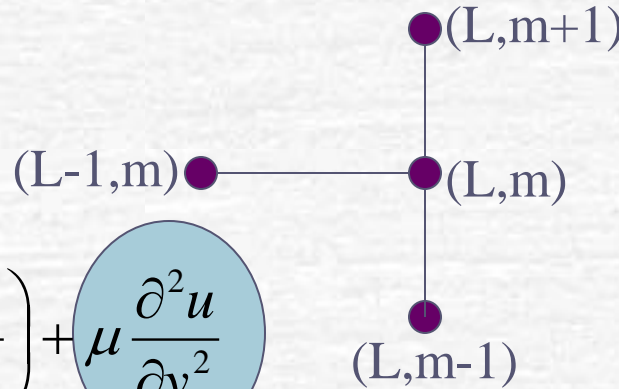
(L,m+1)

(L-1,m) (L,m)

(L,m-1)

y

m=MM+1
m=MM

m=1
m=0

L-1   L   x

$$u\frac{\partial u}{\partial x} = \frac{u_m^l}{\Delta x}\left[u_m^l - u_m^{l-1}\right]$$

$$\mu\frac{\partial^2 u}{\partial y^2} = \frac{\mu}{\Delta y^2}\left[u_{m+1}^l - 2u_m^l + u_{m-1}^l\right]$$

$$v\frac{\partial u}{\partial y} = \frac{v_m^l}{\Delta y}\left[u_{m+1}^l - u_m^l\right]$$

FD $\text{Sign}(v_m^l)<0$

$$= \frac{v_m^l}{\Delta y}\left[u_m^l - u_{m-1}^l\right]$$

BD $\text{Sign}(v_m^l)>0$

2nd order central difference
i.e., theoretical order of accuracy
$P_{kest}= 2$.

1st order upwind scheme, i.e., theoretical order of accuracy $P_{kest}= 1$

31

Discretization methods (example)

$$B_2$$
$$B_3$$
$$B_1$$

$$\left[\frac{u_m^l}{\Delta x} + v_m^l \frac{-\dfrac{1}{\Delta y} FD}{\dfrac{1}{\Delta y} BD} - \frac{2\mu}{\Delta y^2} \right] u_m^l + \left[\frac{\mu}{\Delta y^2} + \frac{v_m^l}{\Delta y} FD \right] u_{m+1}^l + \left[\frac{\mu}{\Delta y^2} - \frac{v_m^l}{\Delta y} BD \right] u_{m-1}^l$$

$$= \frac{u_m^l}{\Delta x} u_m^{l-1} - \frac{\partial}{\partial x}(p/e)_m^l$$

$$B_4$$

$$B_1 u_{m-1}^l + B_2 u_m^l + B_3 u_{m+1}^l = B_4 u_m^{l-1} - \frac{\partial}{\partial x}\left(p/e\right)_m^l$$

$$\begin{bmatrix} B_2 & B_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 & 0 \\ & & \bullet & \bullet & \bullet & \bullet & & \\ 0 & 0 & 0 & 0 & 0 & B_1 & B_2 & B_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & B_1 & B_2 \end{bmatrix} \times \begin{bmatrix} u_1^l \\ \bullet \\ \bullet \\ \bullet \\ u_{mm}^l \end{bmatrix} = \begin{bmatrix} B_4 u_1^{l-1} - \dfrac{\partial}{\partial x}\left(\dfrac{p}{e}\right)_1^l \\ \bullet \\ \bullet \\ \bullet \\ B_4 u_{mm}^{l-1} - \dfrac{\partial}{\partial x}\left(\dfrac{p}{e}\right)_{mm}^l \end{bmatrix}$$

Solve it using Thomas algorithm

To be stable, Matrix has to be Diagonally dominant.

Solvers and numerical parameters

- **Solvers** include: tridiagonal, pentadiagonal solvers, PETSC solver, solution-adaptive solver, multi-grid solvers, etc.

- **Solvers** can be either direct (Cramer's rule, Gauss elimination, LU decomposition) or iterative (Jacobi method, Gauss-Seidel method, SOR method)

- **Numerical parameters** need to be specified to control the calculation.
  - Under relaxation factor, convergence limit, etc.
  - Different numerical schemes
  - Monitor residuals (change of results between iterations)
  - Number of iterations for steady flow or number of time steps for unsteady flow
  - Single/double precisions

Numerical methods (grid generation)

- Grids can either be structured (hexahedral) or unstructured (tetrahedral). Depends upon type of discretization scheme and application
  - Scheme
    - Finite differences: structured
    - Finite volume or finite element: structured or unstructured
  - Application
    - Thin boundary layers best resolved with highly-stretched structured grids
    - Unstructured grids useful for complex geometries
    - Unstructured grids permit automatic adaptive refinement based on the pressure gradient, or regions interested (FLUENT)

structured
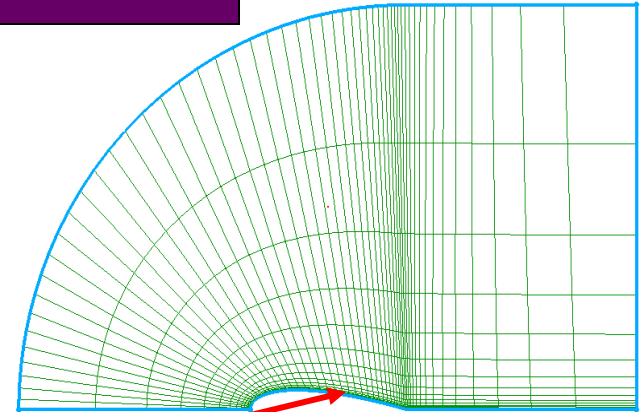
unstructured
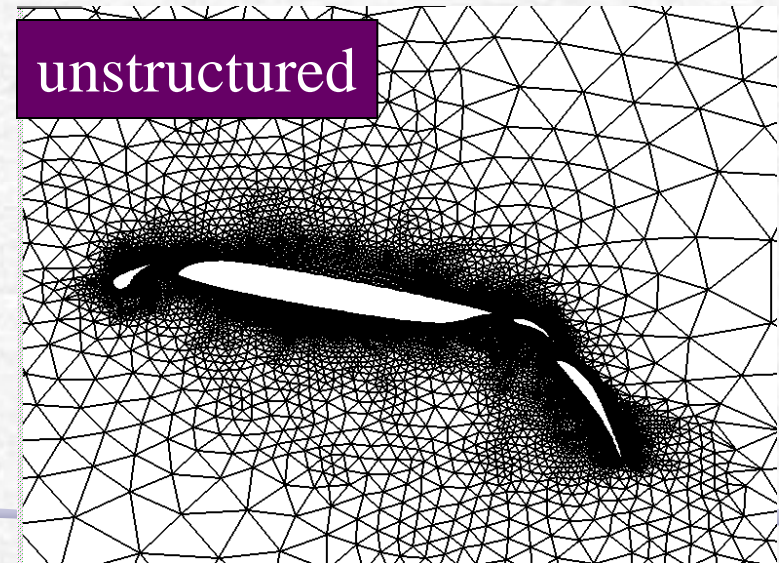
Numerical methods (grid transformation)

η

Transform

ξ

O x O

Physical domain   Computational domain
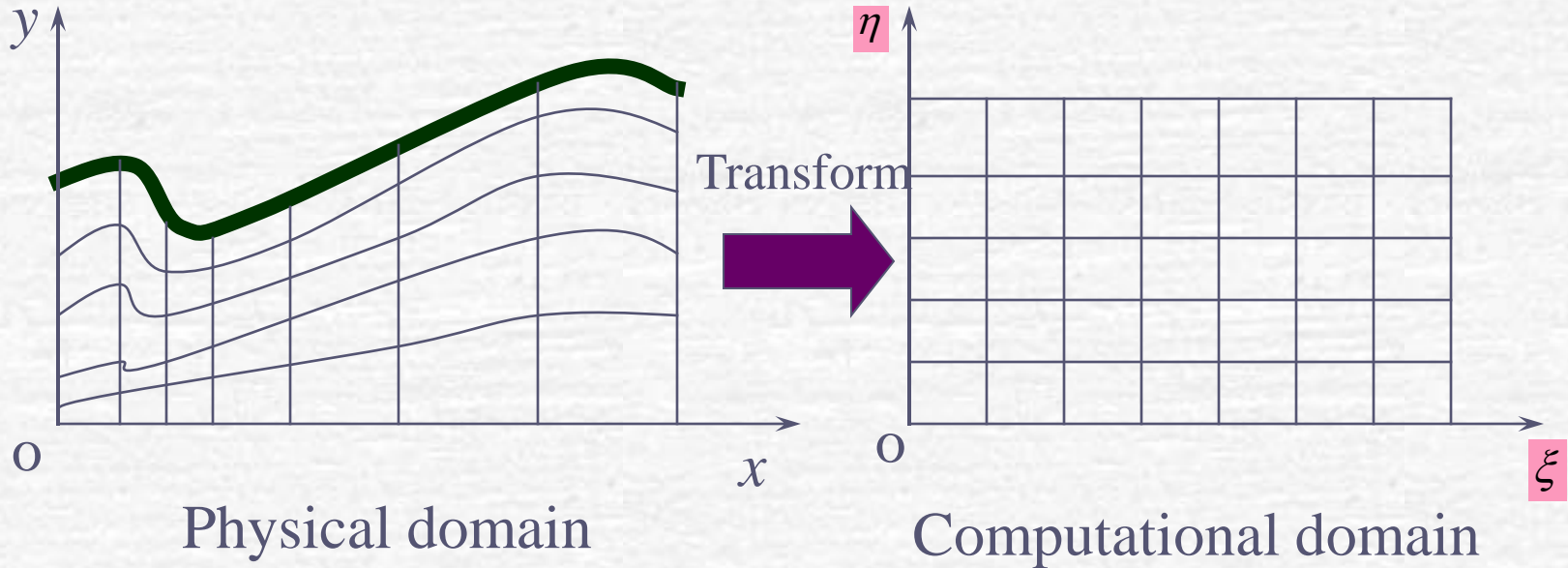
• Transformation between physical (x,y,z) and computational (ξ,η,ζ) domains, important for body-fitted grids. The partial derivatives at these two domains have the relationship (2D as an example)

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial f}{\partial \eta}\frac{\partial \eta}{\partial x} = \xi_x\frac{\partial f}{\partial \xi} + \eta_x\frac{\partial f}{\partial \eta}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial f}{\partial \eta}\frac{\partial \eta}{\partial y} = \xi_y\frac{\partial f}{\partial \xi} + \eta_y\frac{\partial f}{\partial \eta}$$

High performance computing

- CFD computations (e.g. 3D unsteady flows) are usually very expensive which requires parallel high performance supercomputers with the use of multi-block technique.

- As required by the multi-block technique, CFD codes need to be developed using the Massage Passing Interface (MPI) Standard to transfer data between different blocks.

- Emphasis on improving:
  - Strong scalability, main bottleneck pressure Poisson solver for incompressible flow.
  - Weak scalability, limited by the memory requirements.
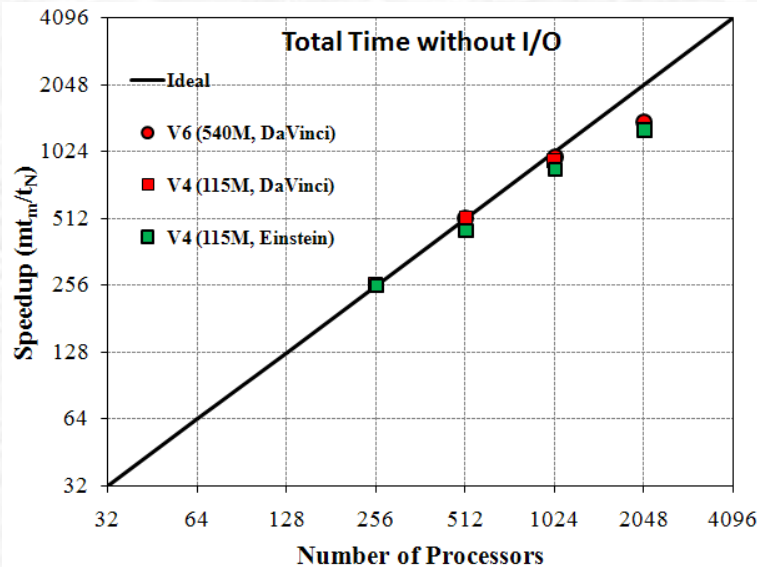
Figure: Strong scalability of total times without I/O for CFDShip-Iowa V6 and V4 on NAVO Cray XT5 (Einstein) and IBM P6 (DaVinci) are compared with ideal scaling.
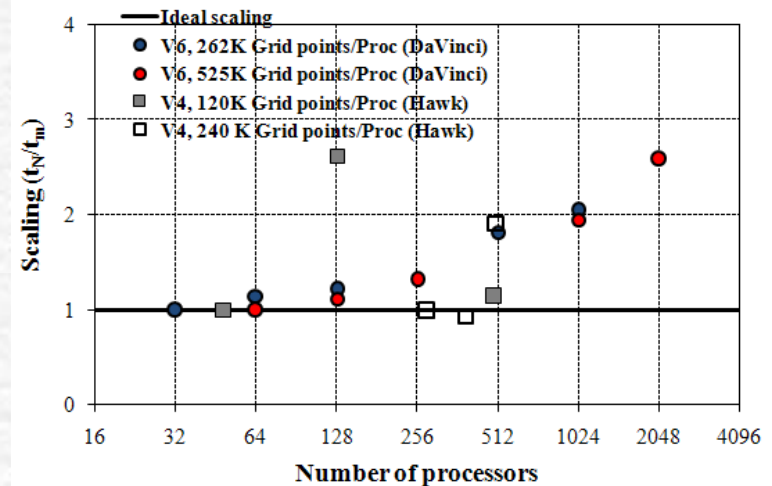
Figure: Weak scalability of total times without I/O for CFDShip-Iowa V6 and V4 on IBM P6 (DaVinci) and SGI Altix (Hawk) are compared with ideal scaling.

Post-Processing

- Post-processing: 1. Visualize the CFD results (contour, velocity vectors, streamlines, pathlines, streak lines, and iso-surface in 3D, etc.), and 2. CFD UA: verification and validation using EFD data (more details later)

- Post-processing usually through using commercial software
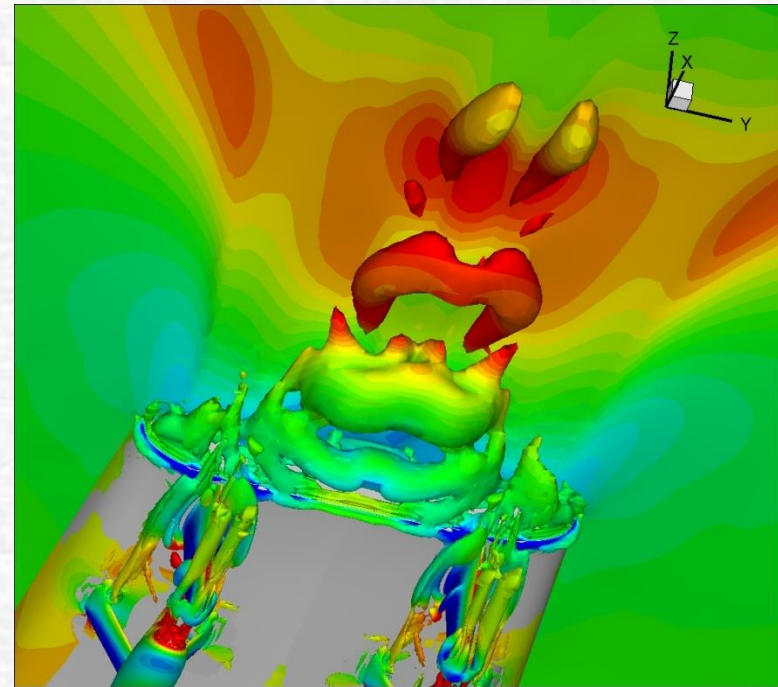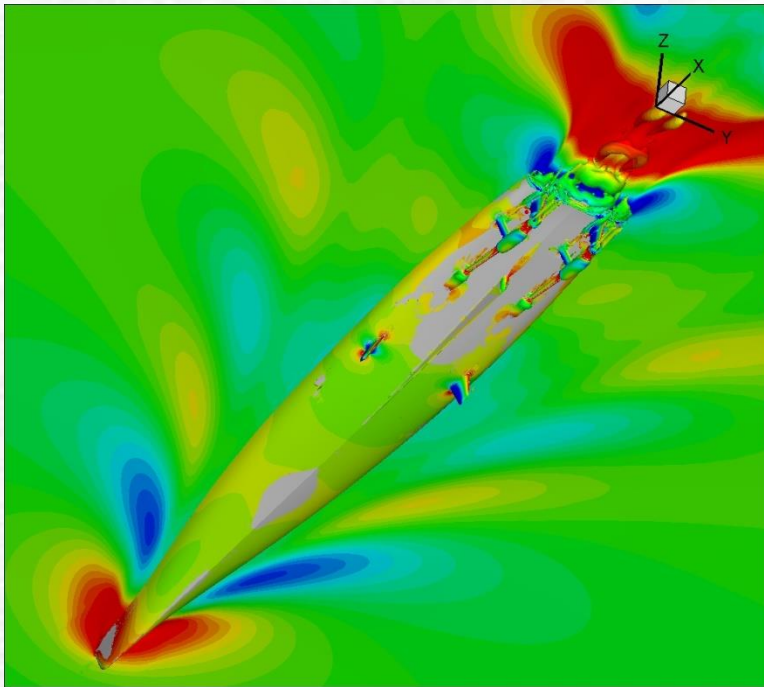
Figure: Isosurface of Q=300 colored using piezometric pressure, free=surface colored using z for fully appended Athena, Fr=0.25, Re=2.9×10^8. Tecplot360 is used for visualization.

Types of CFD codes

- **Commercial CFD code**: ANSYS FLUENT, Star-CCM+, CFDRC, CFX/AEA, etc.
- **Open source CFD code:** OpenFOAM
- **Research CFD code**: CFDSHIP-IOWA
- **Public domain software** (PHI3D, HYDRO, and WinpipeD, etc.)
- Other CFD software includes the Grid generation software (e.g. Pointwise (gridgen), Gambit) and flow visualization software (e.g. Tecplot, Paraview, ANSYS EnSight, FieldView)

CFDSHIPIOWA

ANSYS Interface

Lab1: Pipe Flow	Lab 2: Airfoil Flow	Lab3: Diffuser	Lab4: Ahmed car
1. Definition of "CFD Process" 2. Boundary conditions 3. Iterative error 4. Grid error 5. Developing length of laminar and turbulent pipe flows. 6. Verification using AFD 7. Validation using EFD	1. Boundary conditions 2. Effect of order of accuracy on verification results 3. Effect of grid generation topology, "C" and "O" Meshes 4. Effect of angle of attack/turbulent models on flow field 5. Verification and Validation using EFD	1. Meshing and iterative convergence 2. Boundary layer separation 3. Axial velocity profile 4. Streamlines 5. Effect of turbulence models 6. Effect of expansion angle and comparison with LES, EFD, and RANS.	1. Meshing and iterative convergence 2. Boundary layer separation 3. Axial velocity profile 4. Streamlines 5. Effect of slant angle and comparison with LES, EFD, and RANS.

CFD process

- Purposes of CFD codes will be different for different applications: investigation of bubble-fluid interactions for bubbly flows, study of wave induced massively separated flows for free-surface, etc.

- Depend on the specific purpose and flow conditions of the problem, different CFD codes can be chosen for different applications (aerospace, marines, combustion, multi-phase flows, etc.)

- Once purposes and CFD codes chosen, "CFD process" is the steps to set up the IBVP problem and run the code:

  1. Geometry
  2. Physics (Setup)
  3. Mesh
  4. Solve
  5. Results

CFD Process

Geometry	Physics	Mesh	Solution	Results

Geometry Parameters **(ANSYS Design Modeler)**	Flow properties **(ANSYS Fluent-Setup)**	Unstructured **(ANSYS Mesh)**	Steady/ Unsteady **(ANSYS Fluent-Setup)**	Forces Report **(ANSYS Fluent-Results)**
Domain Shape and Size **(ANSYS Design Modeler)**	Viscous Model **(ANSYS Fluent-Setup)**	Structured **(ANSYS Mesh)**	Iterations/ Steps **(ANSYS Fluent-Solution)**	XY Plot **(ANSYS Fluent-Results)**
	Boundary Conditions **(ANSYS Fluent-Setup)**		Convergent Limit **(ANSYS Fluent-Solution)**	Verification & Validation **(ANSYS Fluent-Results)**
	Initial Conditions **(ANSYS Fluent-Solution)**		Precisions **(ANSYS Fluent-Solution)**	Contours, Vectors, and Streamlines **(ANSYS Fluent-Results)**
			Numerical Scheme **(ANSYS Fluent-Solution)**	

Green regions indicate ANSYS modules

Geometry

- Selection of an appropriate coordinate
- Determine the domain size and shape
- Any simplifications needed?
- What kinds of shapes needed to be used to best resolve the geometry? (lines, circular, ovals, etc.)
- For commercial code, geometry is usually created using commercial software (either separated from the commercial code itself, like Gambit)
- For research code, commercial software (e.g. pointwise) is used.

Physics (Setup)

- Flow conditions and fluid properties

  1. **Flow conditions**: inviscid, viscous, laminar, turbulent, single-phase, multi-phase, and phase change, etc.

  2. **Fluid properties**: density, viscosity, surface tension, and thermal conductivity, etc.

  3. Flow conditions and properties usually presented in dimensional form in industrial commercial CFD software, whereas in non-dimensional variables for research codes.

- Selection of models: different models usually fixed by codes, options for user to choose

- Initial and Boundary Conditions: not fixed by codes, user needs specify them for different applications.

Mesh

- Meshes should be well designed to resolve important flow features which are dependent upon flow condition parameters (e.g., Re), such as the grid refinement inside the wall boundary layer

- Mesh can be generated by either commercial codes (Pointwise/Gridgen, Gambit, etc.) or research code (using algebraic vs. PDE based, conformal mapping, etc.)

- The mesh, together with the boundary conditions need to be exported from commercial software in a certain format that can be recognized by the research CFD code or other commercial CFD software.

Solve

- Setup appropriate numerical parameters
- Choose appropriate Solvers
- Solution procedure (e.g. incompressible flows)

  Solve the momentum, pressure Poisson equations and get flow field quantities, such as velocity, turbulence intensity, pressure and integral quantities (lift, drag forces)

Results

- Reports the saved time history of the residuals of the velocity, pressure and temperature, etc.

- Report the integral quantities, such as total pressure drop, friction factor (pipe flow), lift and drag coefficients (airfoil flow), etc.

- XY plots could present the centerline velocity/pressure distribution, friction factor distribution (pipe flow), pressure coefficient distribution (airfoil flow).

- AFD or EFD data can be imported and put on top of the XY plots for validation

Results

- Analysis and visualization
  - Calculation of derived variables
    - Vorticity
    - Wall shear stress
  - Calculation of integral parameters: forces, moments
  - Visualization (usually with commercial software)
    - Simple 2D contours
    - 3D contour isosurface plots
    - Vector plots and streamlines (streamlines are the lines whose tangent direction is the same as the velocity vectors)
    - Animations
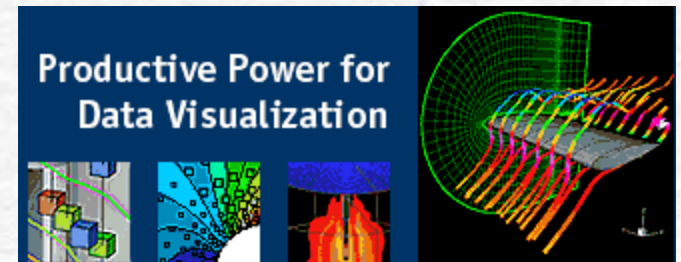
Results (Uncertainty Assessment)

- **Simulation error**: the difference between a simulation result S and the truth T (objective reality), assumed composed of additive modeling δ_{SM} and numerical δ_{SN} errors:

  Error: $\delta_S = S - T = \delta_{SM} + \delta_{SN}$ Uncertainty: $U_S^2 = U_{SM}^2 + U_{SN}^2$

- **Verification**: process for assessing simulation numerical uncertainties U_{SN} and, when conditions permit, estimating the sign and magnitude Delta δ^*_{SN} of the simulation numerical error itself and the uncertainties in that error estimate U_{SN}

$$\delta_{SN} = \delta_I + \delta_G + \delta_T + \delta_P = \delta_I + \sum_{j=1}^{J} \delta_j \qquad U_{SN}^2 = U_I^2 + U_G^2 + U_T^2 + U_P^2$$

I: Iterative, G : Grid, T: Time step, P: Input parameters
Iterative convergence requires U_I to be at least one order of magnitude smaller than U_G and U_T.

- **Validation**: process for assessing simulation modeling uncertainty U_{SM} by using benchmark experimental data and, when conditions permit, estimating the sign and magnitude of the modeling error δ_{SM} itself.

$$U_V^2 = U_D^2 + U_{SN}^2$$

$$E = D - S = \delta_D - (\delta_{SM} + \delta_{SN})$$

$$\boxed{|E| < U_V \qquad \text{Validation achieved}}$$

D: EFD Data; U_V: Validation Uncertainty

Results (UA, Verification)

- **Convergence studies**: Convergence studies require a minimum of m=3 solutions to evaluate convergence with respective to input parameters. Consider the solutions corresponding to fine $\hat{S}_{k1}$, medium $\hat{S}_{k2}$, and coarse meshes $\hat{S}_{k3}$

$$\varepsilon_{k21} = \hat{S}_{k2} - \hat{S}_{k1} \qquad \varepsilon_{k32} = \hat{S}_{k3} - \hat{S}_{k2}$$

$$R_k = \varepsilon_{k21}/\varepsilon_{k32}$$

(i). Monotonic convergence: $0 < R_k < 1$
(ii). Oscillatory Convergence: $R_k < 0; |R_k| < 1$
(iii). Monotonic divergence: $R_k > 1$
(iv). Oscillatory divergence: $R_k < 0; |R_k| > 1$

Monotonic Convergence

Monotonic Divergence

Oscillatory Convergence
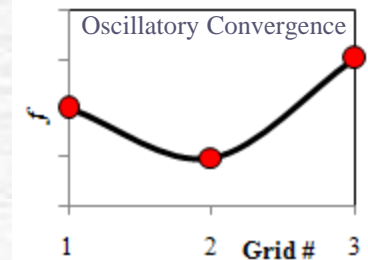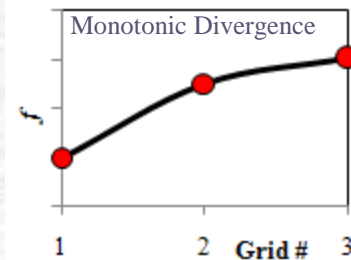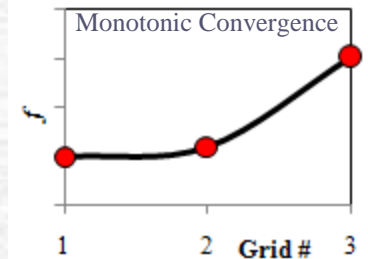
- **Grid refinement ratio**: uniform ratio of grid spacing between meshes.

$$r_k = \Delta x_{k_2}/\Delta x_{k_1} = \Delta x_{k_3}/\Delta x_{k_2} = \Delta x_{k_m}/\Delta x_{k_{m-1}}$$

Results (Verification, RE)

- **Generalized Richardson Extrapolation (RE):** For **monotonic convergence**, generalized RE is used to estimate the error δ^*_k and order of accuracy p_k due to the selection of the *kth* input parameter.

- The error is expanded in a power series expansion with integer powers of Δx_k as a finite sum.

- The accuracy of the estimates depends on how many terms are retained in the expansion, the magnitude (importance) of the higher-order terms, and the validity of the assumptions made in RE theory

Results (Verification, RE)

$$\delta_{SN} = \delta_{SN}^{*} + \varepsilon_{SN}$$

ε_{SN} is the error in the estimate

$$S_C = S - \delta_{SN}^{*}$$

S_C is the numerical benchmark

Power series expansion

Finite sum for the *k*th parameter and *m*th solution

$$\hat{S}_{k_m} = S_{k_m} - \delta_{I_{k_m}}^{*} = S_C + \delta_{k_m}^{*} + \sum_{j=1, j \neq k}^{J} \delta_{jm}^{*} \qquad \delta_{k_m}^{*} = \sum_{i=1}^{n} \left(\Delta x_{k_m} \right)^{p_k^{(i)}} g_k^{(i)}$$

$$\hat{S}_{k_m} = S_C + \sum_{i=1}^{n} \left(\Delta x_{k_m} \right)^{p_k^{(i)}} g_k^{(i)} + \sum_{j=1, j \neq k}^{J} \delta_{jm}^{*}$$

$p_k^{(i)}$ order of accuracy for the *i*th term

$$\hat{S}_{k_1} = S_C + \left(\Delta x_{k_1} \right)^{p_k^{(1)}} g_k^{(1)} + \sum_{j=1, j \neq k}^{J} \delta_{j1}^{*} \qquad \hat{S}_{k_2} = S_C + \left(r_k \Delta x_{k_1} \right)^{p_k^{(1)}} g_k^{(1)} + \sum_{j=1, j \neq k}^{J} \delta_{j2}^{*}$$
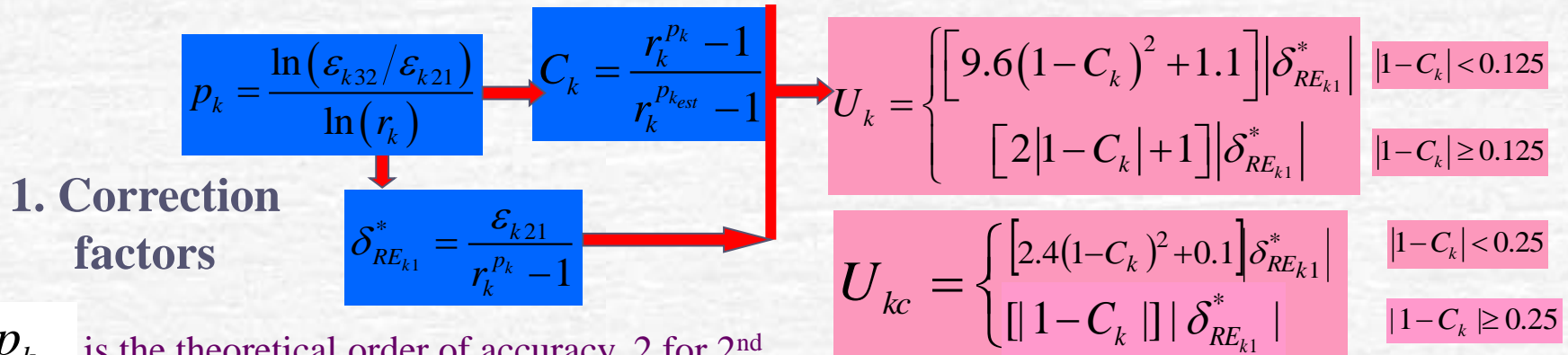
$$\hat{S}_{k_3} = S_C + \left(r_k^2 \Delta x_{k_1} \right)^{p_k^{(1)}} g_k^{(1)} + \sum_{j=1, j \neq k}^{J} \delta_{j3}^{*}$$

Three equations with three unknowns

$$p_k = \frac{\ln\left(\varepsilon_{k_{32}} / \varepsilon_{k_{21}} \right)}{\ln(r_k)} \qquad \delta_{k_1}^{*} = \delta_{RE_{k1}}^{*} = \frac{\varepsilon_{k_{21}}}{r_k^{p_k} - 1}$$

- **Monotonic Convergence**: Generalized Richardson Extrapolation

$$p_k = \frac{\ln\left(\varepsilon_{k32}/\varepsilon_{k21}\right)}{\ln\left(r_k\right)}$$

$$C_k = \frac{r_k^{p_k}-1}{r_k^{p_{k_{est}}}-1}$$

1. Correction factors

$$\delta_{RE_{k1}}^* = \frac{\varepsilon_{k21}}{r_k^{p_k}-1}$$

$$U_k = \begin{cases} \left[9.6\left(1-C_k\right)^2+1.1\right]\left|\delta_{RE_{k1}}^*\right| & |1-C_k|<0.125 \\ \left[2\left|1-C_k\right|+1\right]\left|\delta_{RE_{k1}}^*\right| & |1-C_k|\geq 0.125 \end{cases}$$

$$U_{kc} = \begin{cases} \left[2.4(1-C_k)^2+0.1\right]\delta_{RE_{k1}}^* & |1-C_k|<0.25 \\ [\,|1-C_k|\,]\,|\delta_{RE_{k1}}^*| & |1-C_k|\geq 0.25 \end{cases}$$

$p_{k_{est}}$ is the theoretical order of accuracy, 2 for 2nd order and 1 for 1st order schemes

C_k is the correction factor

U_k is the uncertainties based on fine mesh solution. U_{kc} is the uncertainties based on numerical benchmark S_C

2. GCI approach

$$U_k = F_s\left|\delta_{RE_{k1}}^*\right| \qquad U_{kc} = \left(F_s-1\right)\left|\delta_{RE_{k1}}^*\right|$$

F_S: Factor of Safety

3. FS approach

$$P = \frac{P_k}{P_{kest}} \qquad U_k = \begin{cases} (2.45-0.85P)\left|\delta_{REk1}^*\right| & \text{if} \quad 0<P\leq 1 \\ (16.4P-14.8)\left|\delta_{REk1}^*\right| & \text{if} \quad P>1 \end{cases}$$

- **Oscillatory Convergence**: Uncertainties can be estimated, but without signs and magnitudes of the errors. $U_k = \frac{1}{2}\left(S_U - S_L\right)$

- **Divergence**

In this course, only grid uncertainties studied. So, all the variables with subscribe symbol k will be replaced by g, such as "U_k" will be "U_g"

Results (Verification, Asymptotic Range)

- **Asymptotic Range:** For sufficiently small Δx_k, the solutions are in the asymptotic range such that higher-order terms are negligible and the assumption that $p_k^{(i)}$ and $g_k^{(i)}$ are independent of Δx_k is valid.

- When Asymptotic Range reached, P_g will be close to the theoretical value P_{gest}, and the ratio P of accuracy of grid and theoretical accuracy, will be close to 1.

- To achieve the asymptotic range for practical geometry and conditions is usually not possible and number of grids m>3 is undesirable from a resources point of view

Results (UA, Verification, cont'd)

- **Verification for velocity profile using AFD:** To avoid ill-defined ratios, L2 norm of the ε_{G21} and ε_{G32} are used to define R_G and P_G

$$\langle R_G \rangle = \left\| \varepsilon_{G_{21}} \right\|_2 \Big/ \left\| \varepsilon_{G_{32}} \right\|_2 \qquad \langle p_G \rangle = \frac{\ln\left(\left\| \varepsilon_{G_{32}} \right\|_2 \Big/ \left\| \varepsilon_{G_{21}} \right\|_2 \right)}{\ln(r_G)}$$

Where $<>$ and $\| \|_2$ are used to denote a profile-averaged quantity (with ratio of solution changes based on L2 norms) and L2 norm, respectively.

NOTE: For verification using AFD for axial velocity profile in laminar pipe flow (CFD Lab1), there is no modeling error, only grid errors. So, the difference between CFD and AFD, E, can be plot with +Ug and –Ug to see if solution was verified.
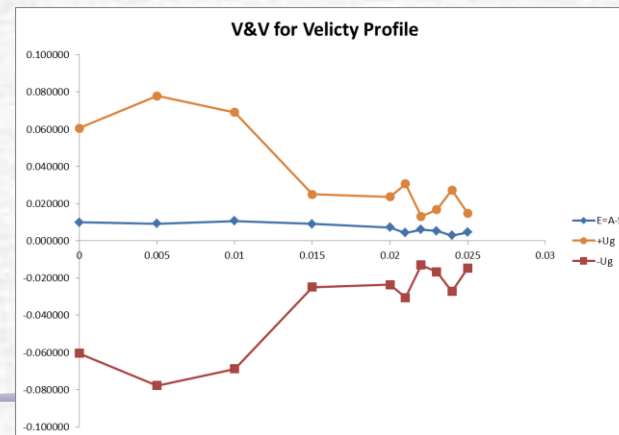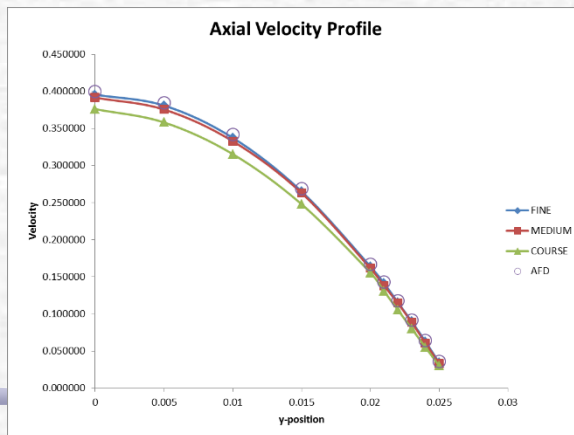
Results (Verification: Iterative Convergence)

- Typical CFD solution techniques for obtaining steady state solutions involve beginning with an initial guess and performing time marching or iteration until a steady state solution is achieved.
- The number of order magnitude drop and final level of solution residual can be used to determine stopping criteria for iterative solution techniques (1) Oscillatory (2) Convergent (3) Mixed oscillatory/convergent

$$U_I = \left| \frac{1}{2}(S_U - S_L) \right|$$
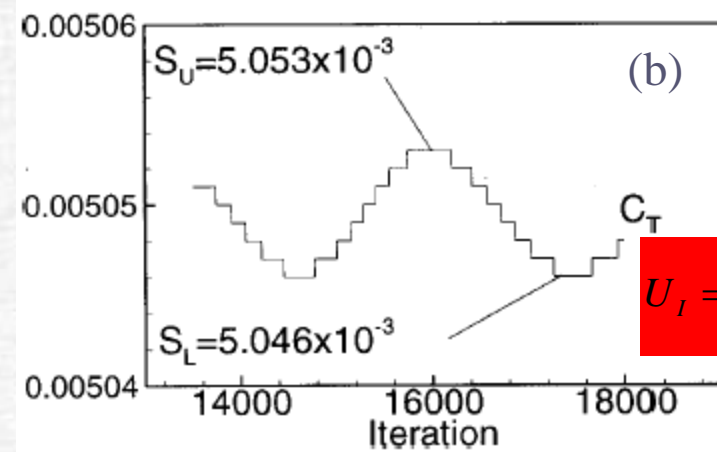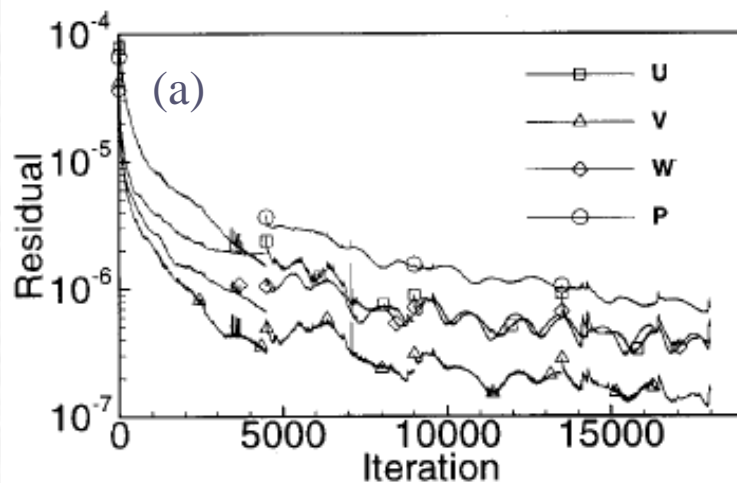
Iteration history for series 60: (a). Solution change (b) magnified view of total resistance over last two periods of oscillation (Oscillatory iterative convergence)

Results (UA, Validation)

- Validation procedure: simulation modeling uncertainties was presented where for successful validation, the comparison error, E, is less than the validation uncertainty, Uv.
- Interpretation of the results of a validation effort

$$|E| < U_V \quad \text{Validation achieved}$$

$$U_V < |E| \quad \text{Validation not achieved}$$

$$E = D - S = \delta_D - (\delta_{SM} + \delta_{SN})$$
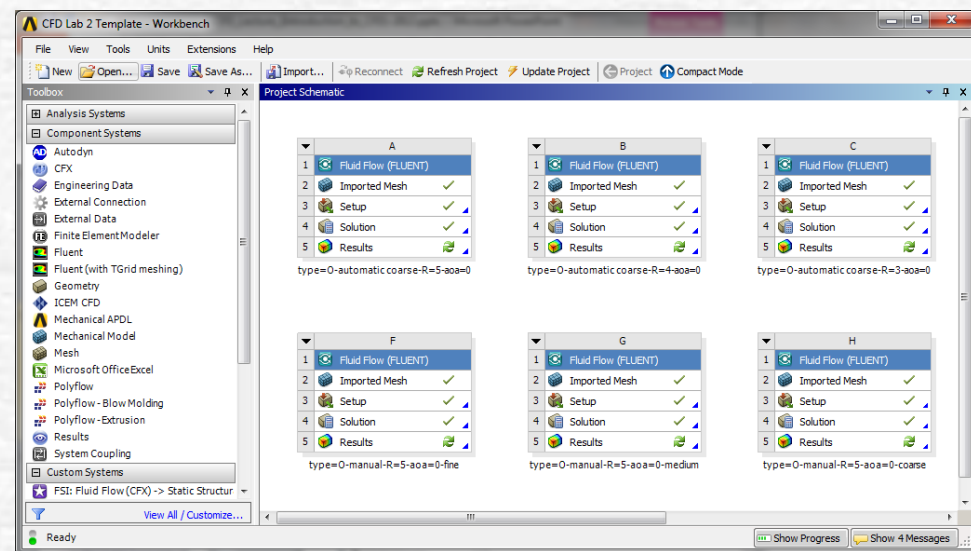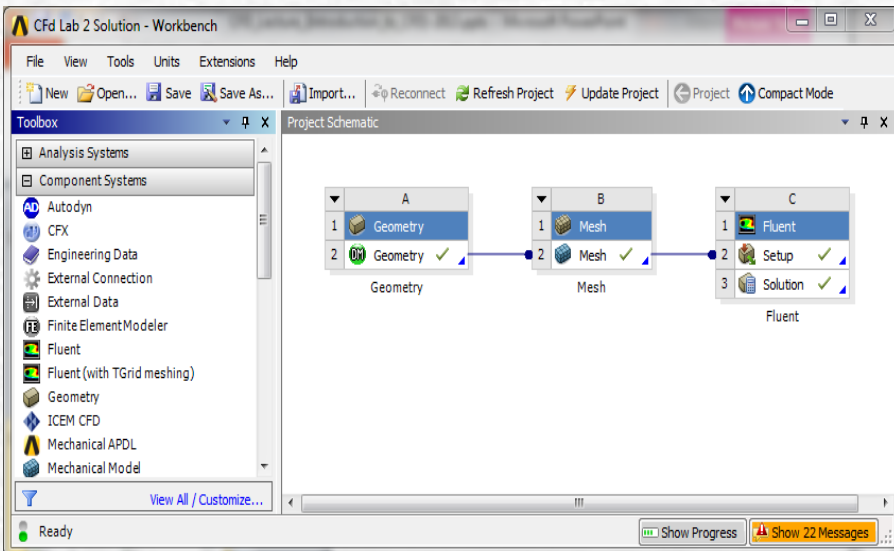
$$U_V = \sqrt{U_{SN}^2 + U_D^2}$$

- Validation example
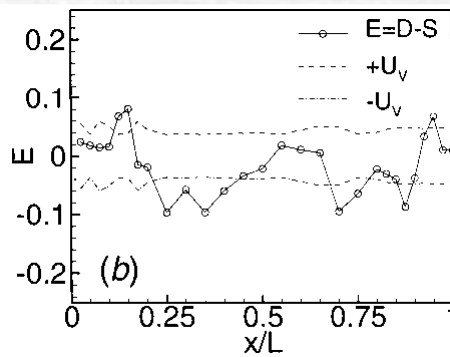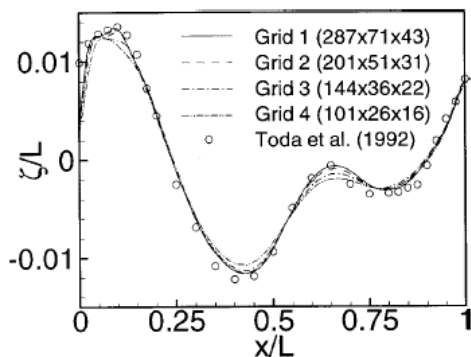
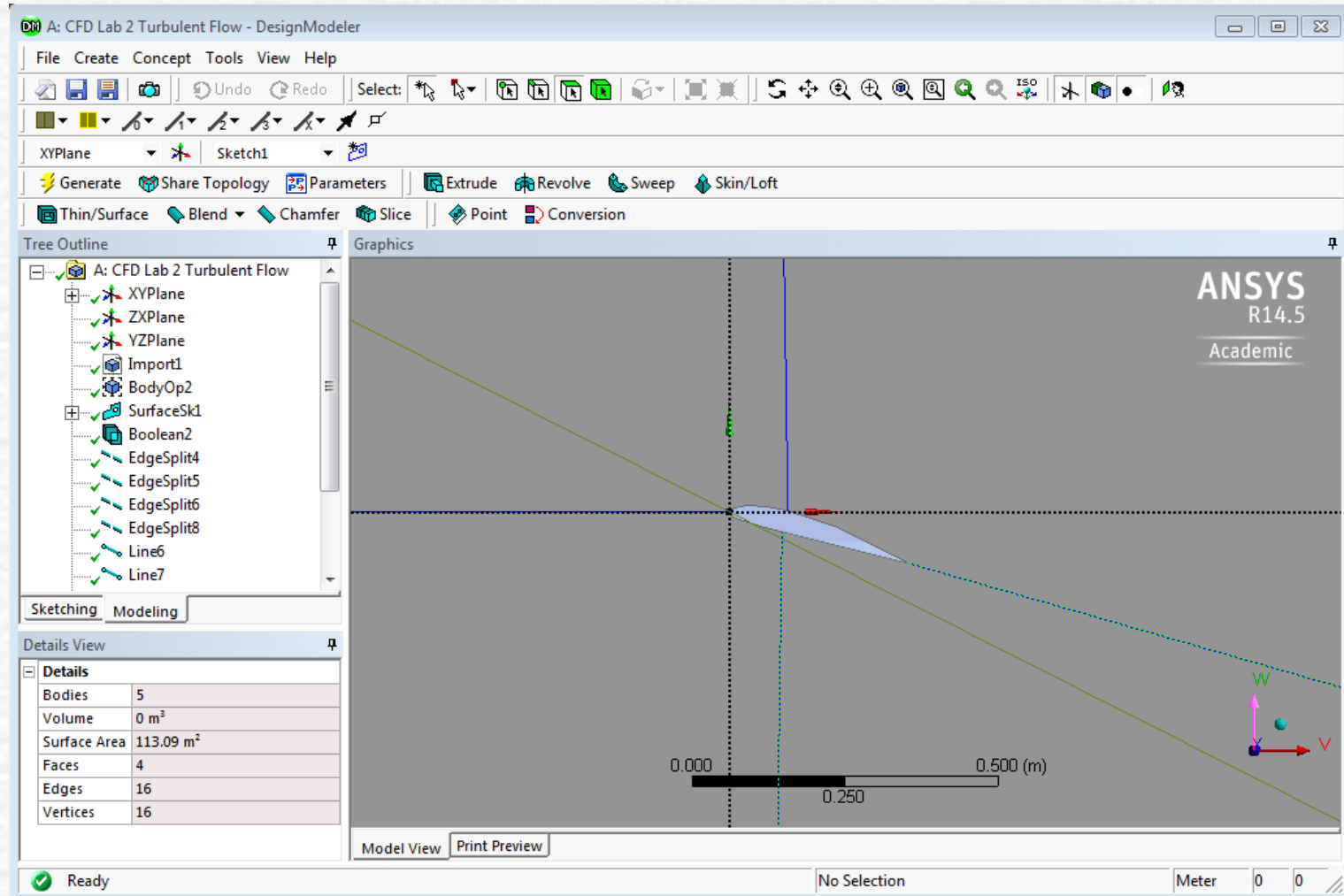Example: Grid study and validation of wave profile for series 60

ANSYS Workbench

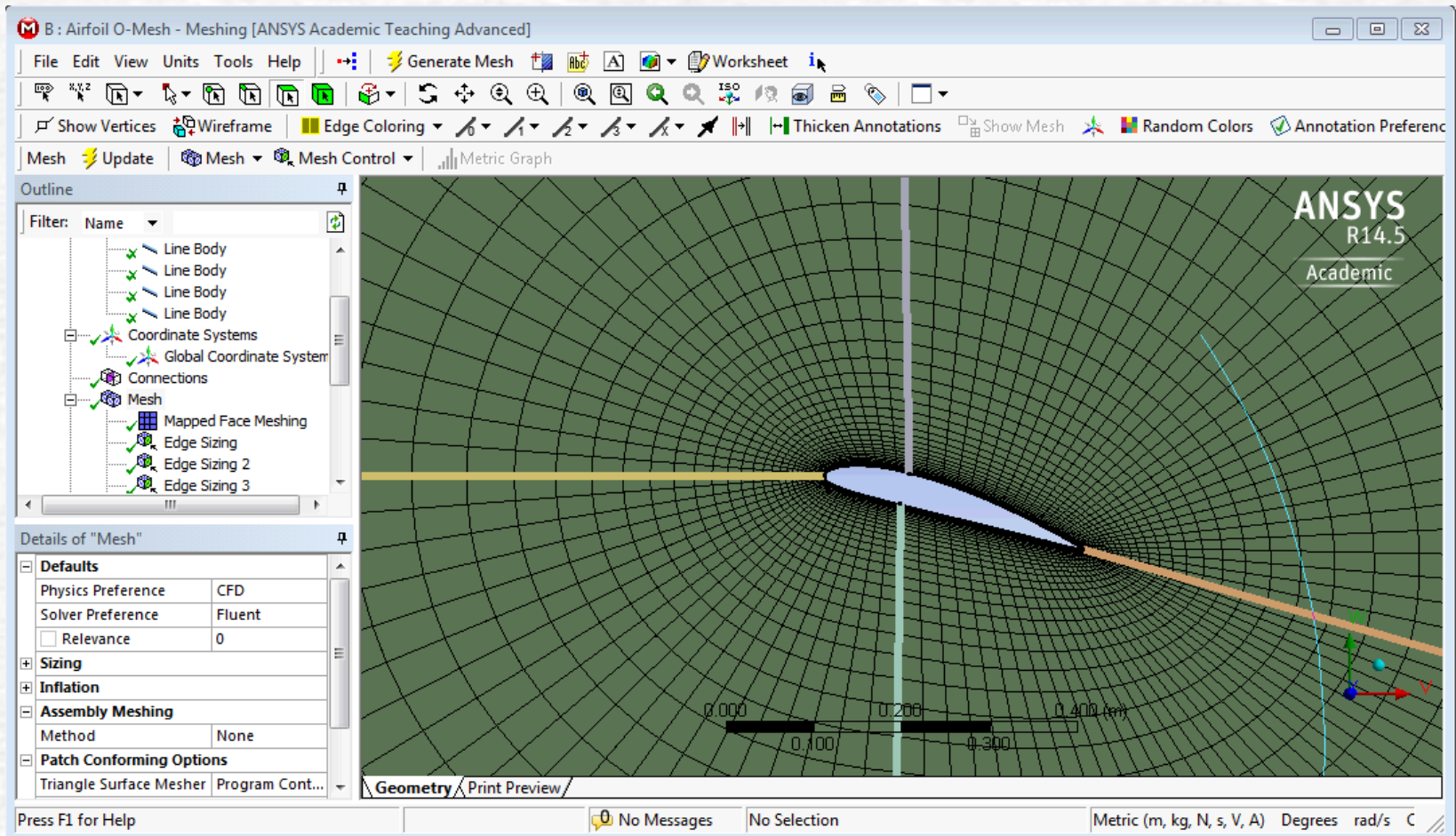- Design project schematics with ANSYS Workbench

ANSYS Design Modeler
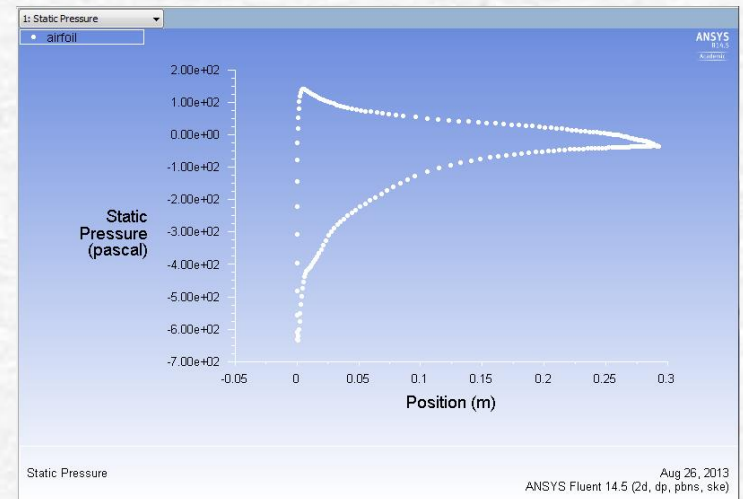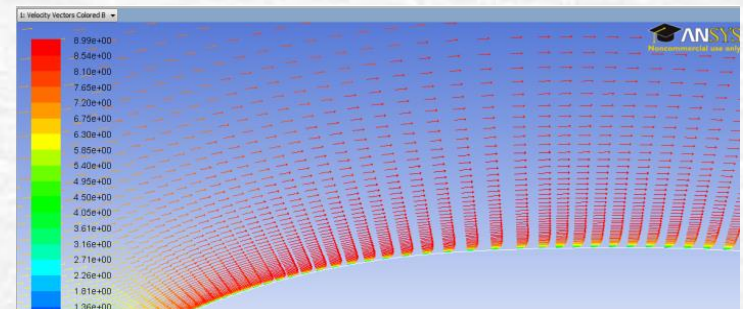
- Create geometry using ANSYS Design Modeler
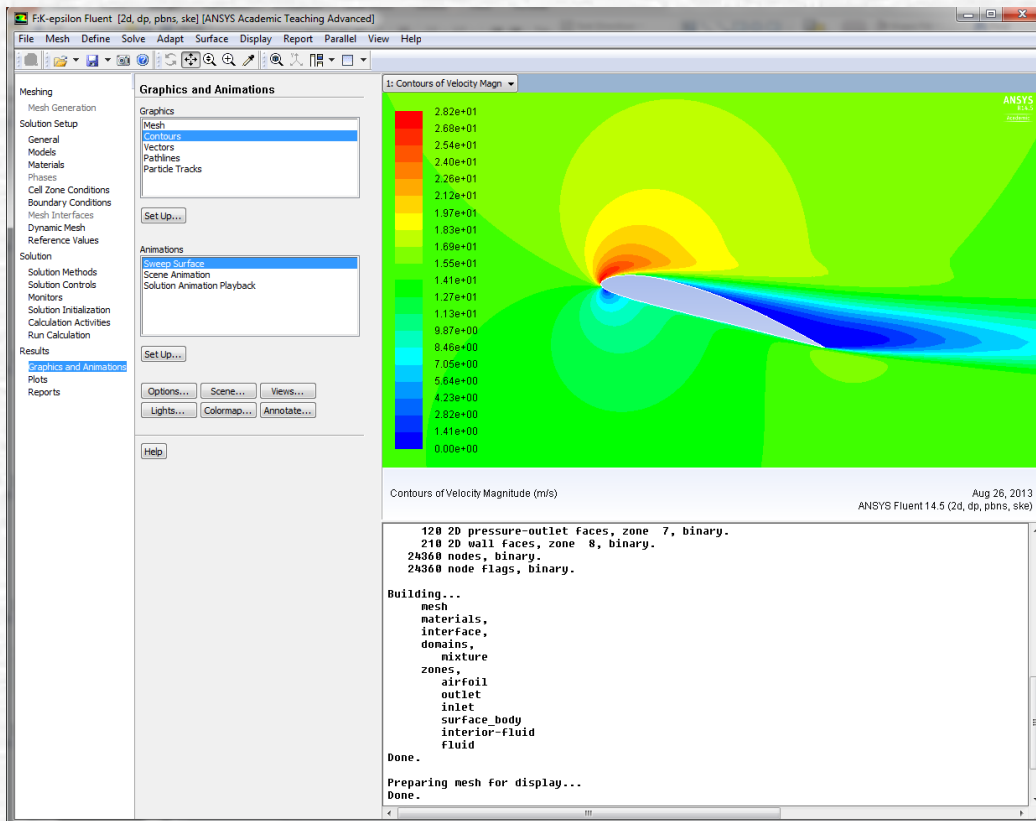
ANSYS Mesh

- Create mesh using ANSYS Mesh

ANSYS Fluent

- Setup and solve problem, and analyze results using ANSYS Fluent

ME:5160 CFD Labs

Schedule

CFD Lab	Lab1: Pipe Flow	Lab 2: Airfoil Flow	Lab3: Diffuser	Lab4: Ahmed car
Due Date	Sep 25	Oct. 11	Nov. 3	Dec. 2

- CFD lectures by Zhaoyuan-Wang (zhaoyuan-wang@uiowa.edu)

- You can use any computer lab in SC for accessing ANSYS software

- Refer to V&V Papers (Ref1, Ref2) and instruction (V&V Instruction) on class website

- Use the lab report instructions for preparing your reports

- Visit class website for more information http://css.engineering.uiowa.edu/~me_160