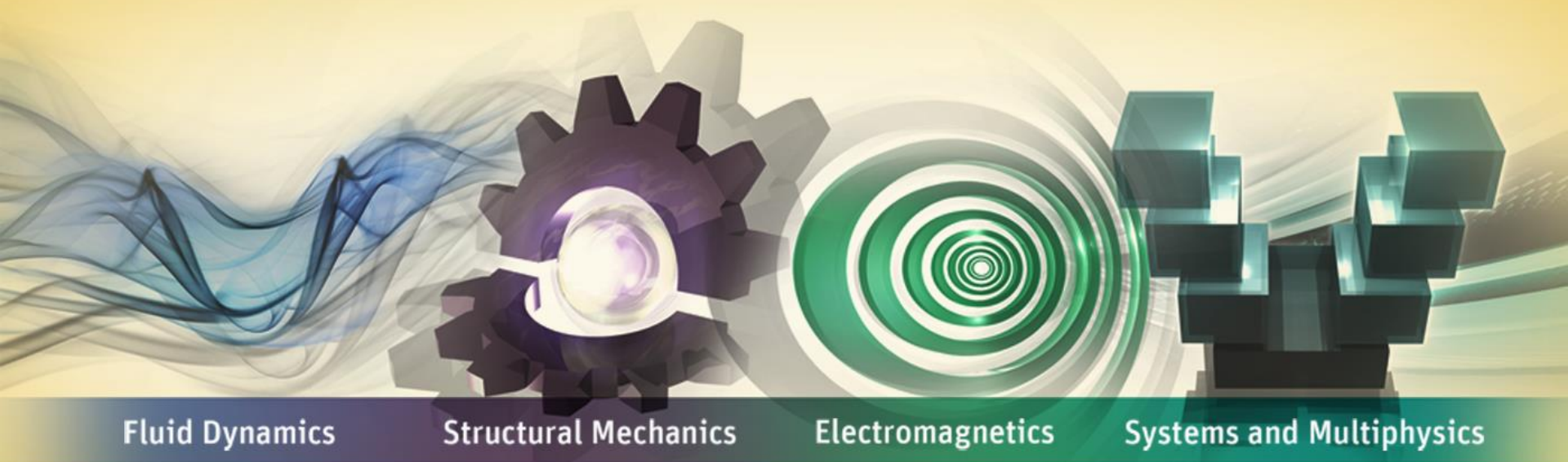


Simulating Sinkage & Trim for Planing Boat Hulls



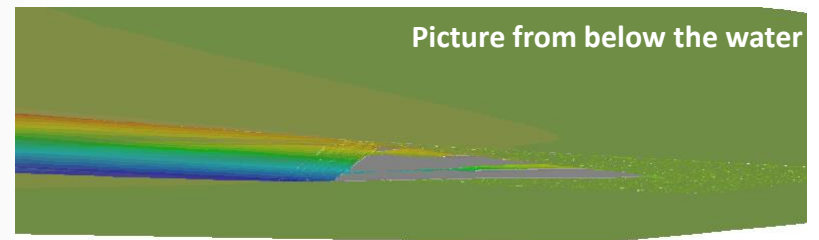
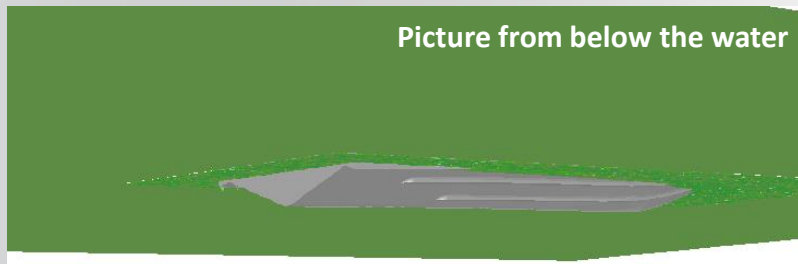
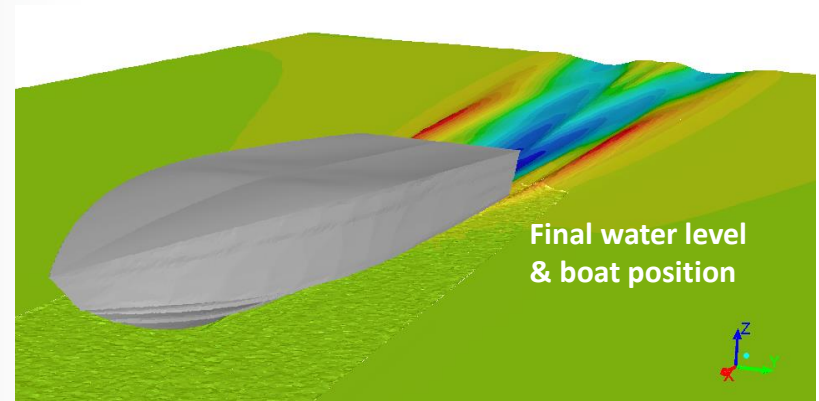
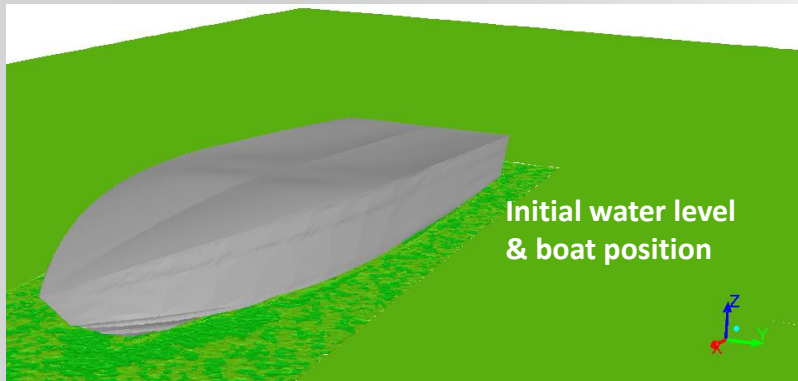
A Fluent Dynamic Mesh 6DOF Tutorial

Introduction

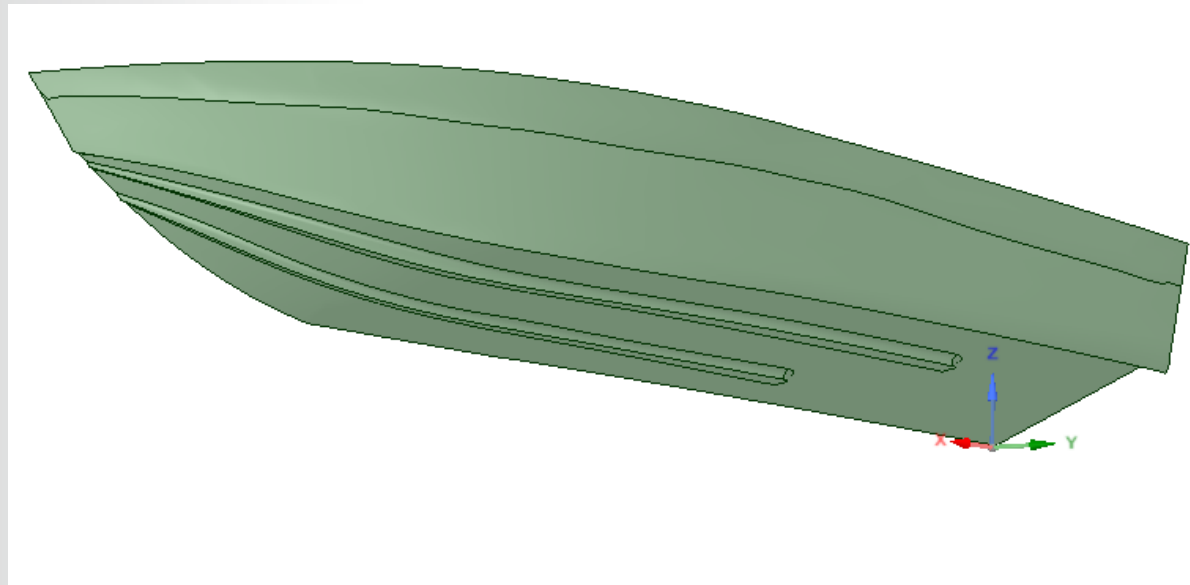
- **Workshop Description**
 - This workshop describes how to perform a transient 2DOF simulation of a planing speed boat in ANSYS Fluent
 - The most important output from this simulation is:
 - Drag
 - Trim angle and Sinkage (water level on the hull)
 - Techniques used
 - Open channel VOF (Volume of Fluid)
 - Fluent's 6DOF solver, solves the rigid body motion of the hull
 - Moving Deforming Mesh (MDM), deforms the mesh to allow motion of the boat

Simulation to be Performed

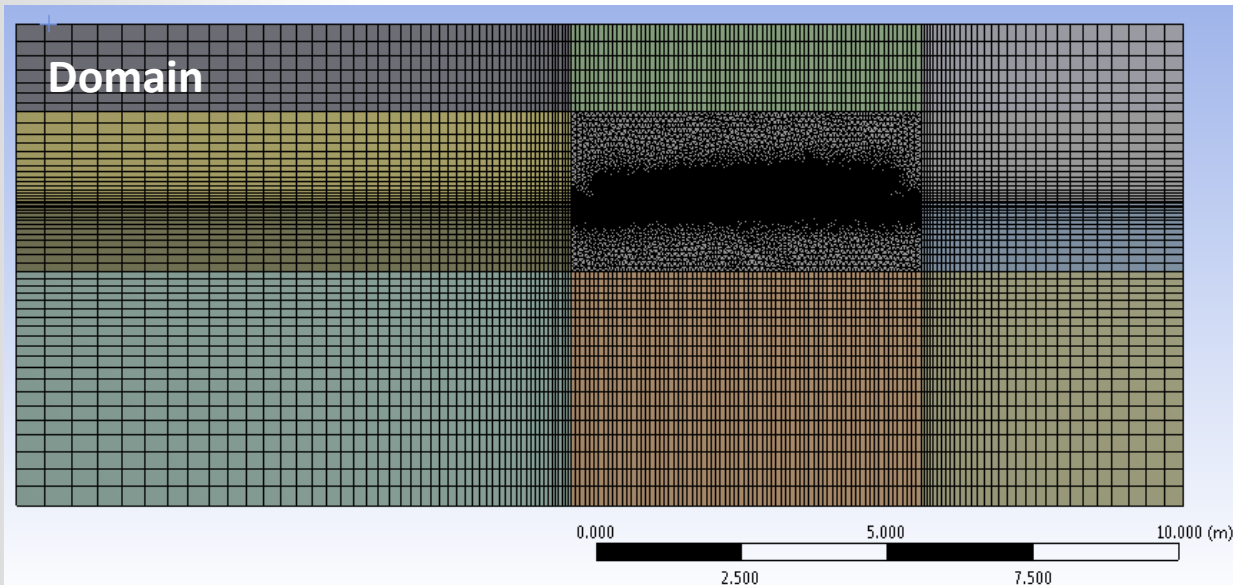
- The images below show the boat's initial and final position as well as the developed wave pattern



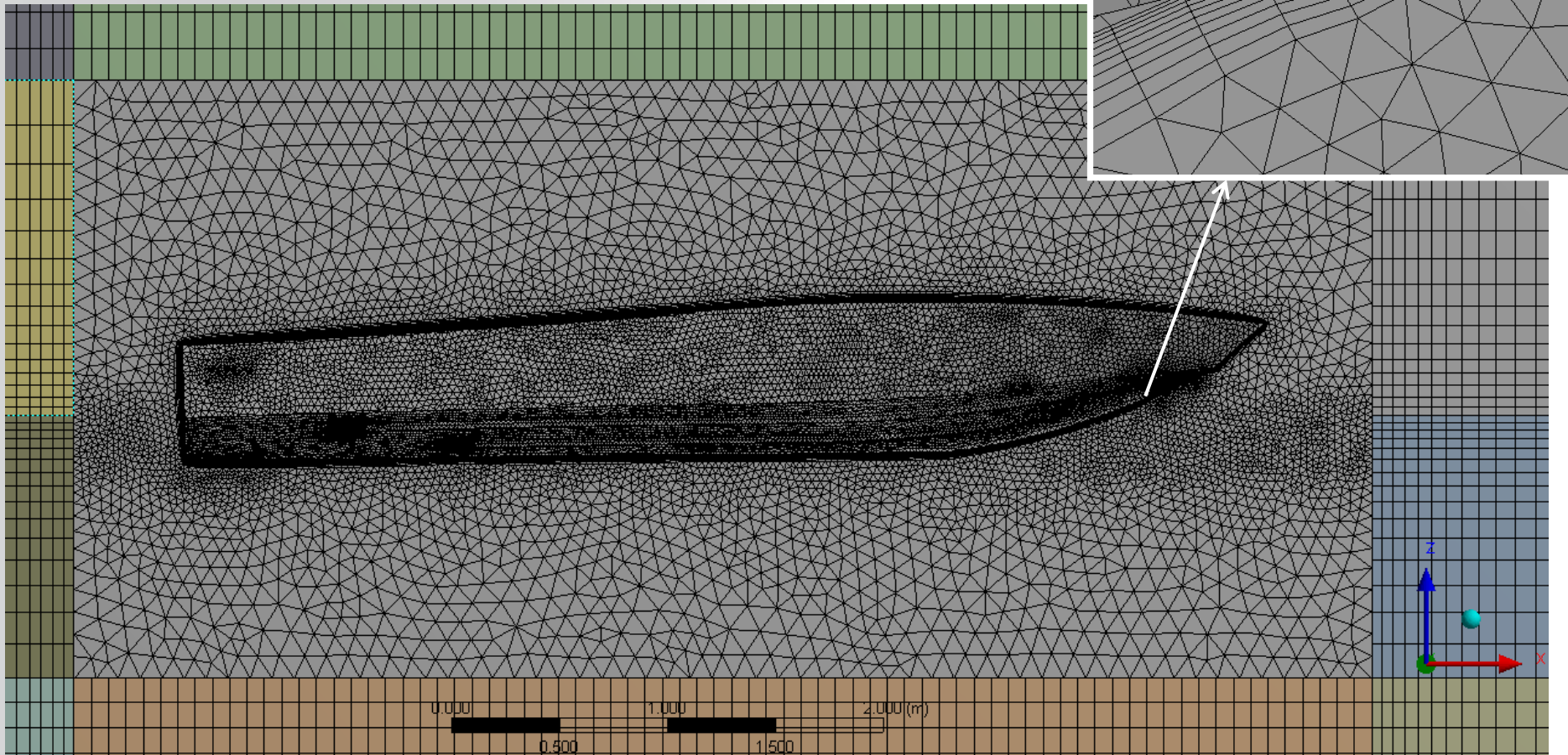
- A generic boat hull was used
 - This boat hull was created for demonstration purposes and is not a real/commercial hull
 - Symmetry plane defined at $y=0$
 - Boat length = 5m
 - Weight = 500kg (250kg for half boat)



- A relatively small domain and coarse mesh was used for this workshop
 - Domain length ~4 boat lengths
 - 1 boat length upstream the hull and 2 boat lengths downstream the hull
 - Mesh size: ~1.6M cells
 - (1.3M tets and prisms close to the boat & 0.3M hex cells in the outer domain)
 - Non-conformal mesh interfaces used between tet/hex zones to keep a low cell count
 - Mesh refinement at water level



- 10 prims layers created on the hull
 - Body of influence used to refine the tet mesh at the water level



Mesh operations in Fluent

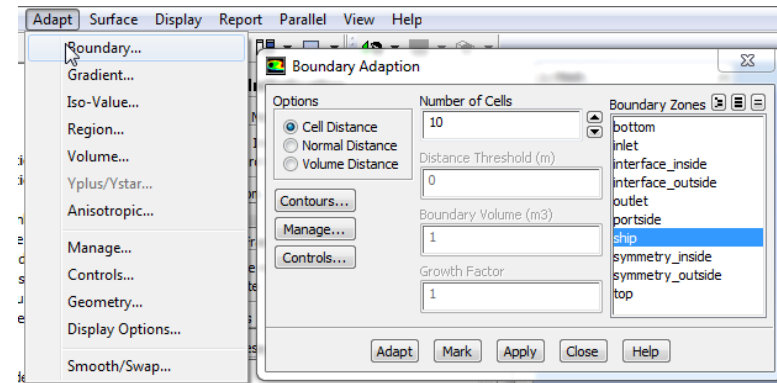
- To ensure a robust mesh smoothing it is a good practice, for 6DOF cases, to define the prism layer as a separate cell zone.
- This can easily be done in Fluent
 - It is however recommended to perform this mesh operation in Fluent outside of workbench
 - It can be done in Workbench but is not compatible with automatic geometry/mesh updates

Mesh operations in Fluent

- Separate the prisms into a new cell zone:
 - Read the mesh into Fluent and initialize the case

- Mark the prisms adjacent to the boat hull using Adapt/Boundary

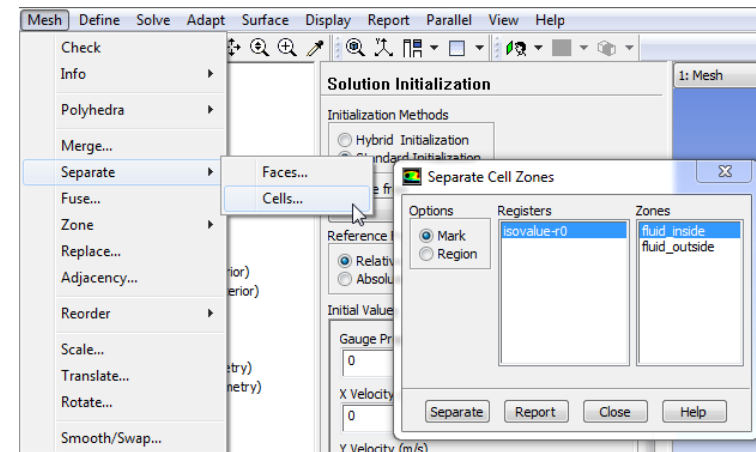
- The prism layer must have been created without “stair stepping” to get a pure split between tets and prisms
(this method will however still work with stairstepped prisms)



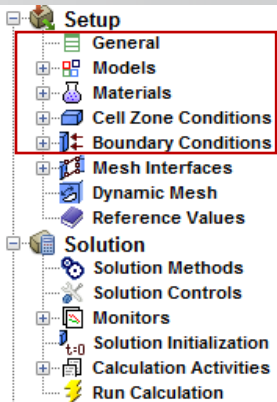
- Use the marked cells to separate the tet/prism zone into 2 zones

- It is best to do this operation before setting up any interfaces or MDM zones
- Note that you also will separate the symmetry into a tet and a prism adjacent zone

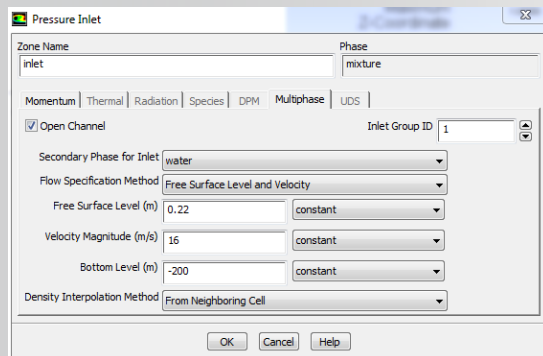
- Separate msh files can be combined using:
 - Mesh/Zone/Append Case File



Simulation Settings

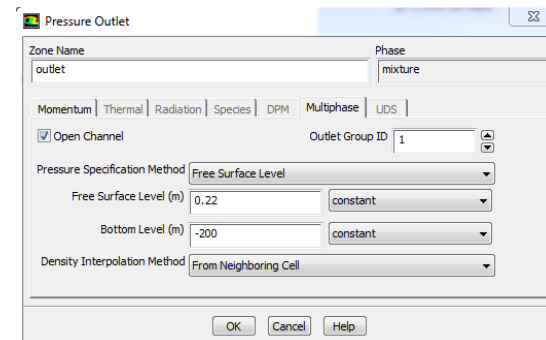


- **General: Transient & Gravity**
- **Models:**
 - **Implicit VOF with open channel flow and sharp interface modeling**
 - Primary phase: air, Secondary phase: water
 - Phase interaction: constant surface tension=0.072 and default wall adhesion (90 degree contact angle on the hull)
 - **K-omega SST with production limiter and turbulence damping (factor 10)**
- **Materials: Incompressible air and water**
- **Boundary Conditions**
 - Define all outer boundary faces, except inlet, outlet and the boat, as symmetries
 - **Pressure Inlet:**



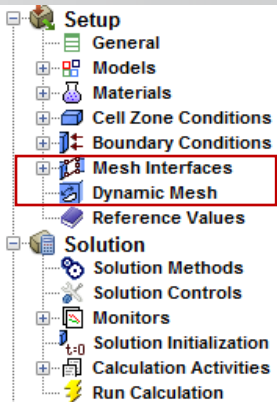
Turbulence Settings:
3% Turbulence intensity,
Turb. Visc. Ratio =10

Pressure Outlet:

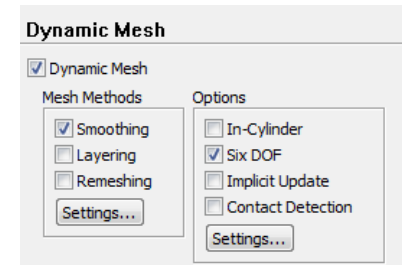


- **Operating Conditions: 1atm operating pressure, gravity and air as operating density**

Simulation Settings



- **Mesh Interfaces**
 - Define mesh interface between the hex mesh and the tet mesh
 - Make sure to have separate zone id's for each interface surface on the tet side (Needed for the MDM settings)
- **Dynamic Mesh**
 - Activate Diffusion Based smoothing and 6DOF
 - Use a diffusion parameter (boundary-distance) ~ 1
 - Control the gravity settings for 6DOF and activate “write motion history”
 - Create the 6DOF UDF and compile it
 - Example:



```
#include "udf.h"

DEFINE_SDOF_PROPERTIES(ship, prop, dt, time, dtime)
{
    prop[SDOF_MASS] = 250;
    prop[SDOF_IYY] = 673;
    prop[SDOF_IXX] = 1;
    prop[SDOF_IZZ] = 1;
    prop[SDOF_ZERO_TRANS_X] = TRUE;
    prop[SDOF_ZERO_TRANS_Y] = TRUE;
    prop[SDOF_ZERO_TRANS_Z] = FALSE;
    prop[SDOF_ZERO_ROT_X] = TRUE;
    prop[SDOF_ZERO_ROT_Y] = FALSE;
    prop[SDOF_ZERO_ROT_Z] = TRUE;
}
```

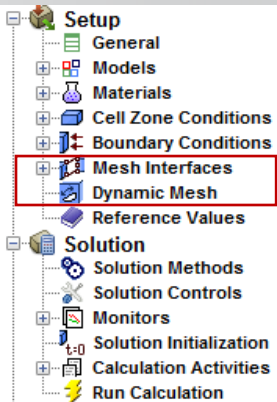
Comments:

“ship” is the name of the 6DOF property

prop[SDOF_MASS] = 250; mass for a half boat
 prop[SDOF_IYY] = 673; moment of inertia for a half boat
 prop[SDOF_IXX] = 1; $\neq 0$
 prop[SDOF_IZZ] = 1;
 prop[SDOF_ZERO_TRANS_Z] = FALSE; Allowed degrees of freedom
 prop[SDOF_ZERO_ROT_Y] = FALSE;

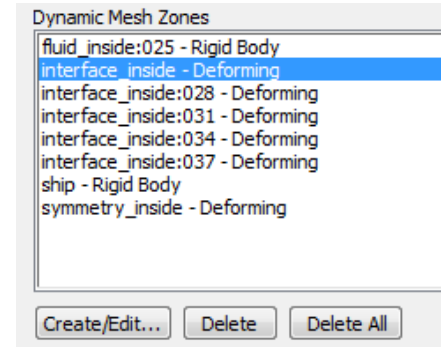
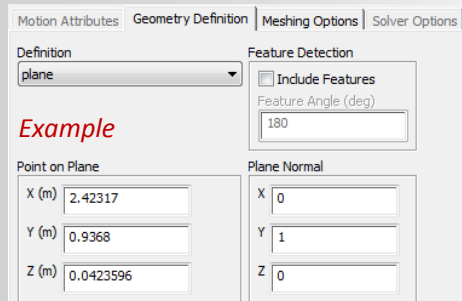
(The full mass and moment of inertia could be defined in the UDF if combined with the 6DOF property: SDOF_SYMMETRY_Y)

Simulation Settings

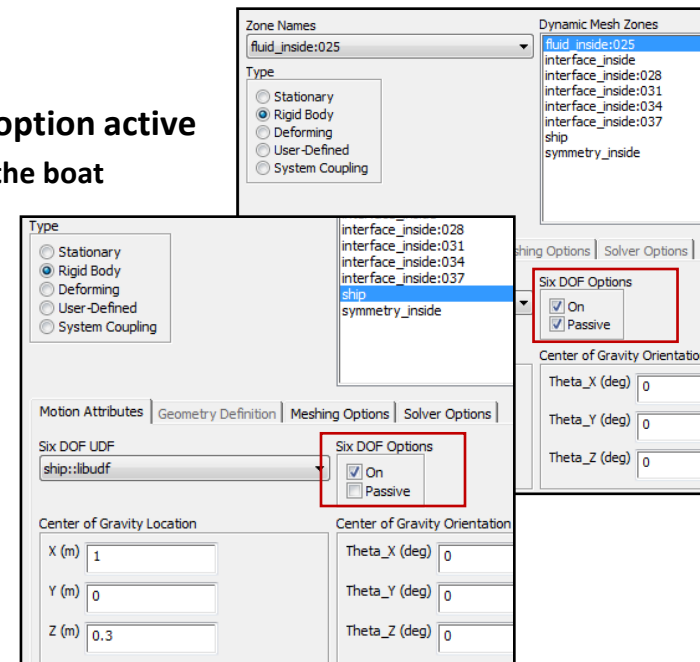


• Dynamic Mesh Zones

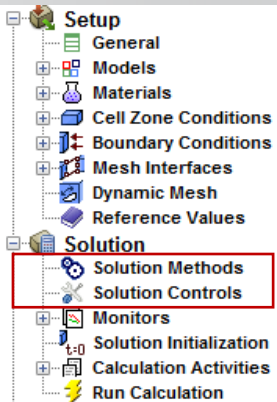
- Define each of the interface surfaces as deforming zones
- Define the symmetry adjacent to the tet zone as deforming zone
 - Specify planar geometry definitions for each of the above (the sign of the normal vector is not important)



- Define the boat hull as a rigid body zone with 6DOF option active
 - States that forces and moments should be calculated on the boat
 - Specify the correct center of mass for the boat
 - “Solver Options” will be needed, see slide :17
 - Define solution parameter = 0.2 (coefficient-based)
- Do the same setup for the prism layer zone but add the 6DOF option “passive” for this zone
 - Allows the zone to get the same motion as the boat without evaluating forces and moments on this zone



Simulation Settings



• Solution Methods:

Solution Methods

Pressure-Velocity Coupling

Scheme

☐ Coupled with Volume Fractions

Spatial Discretization

Gradient

Pressure

Momentum

Volume Fraction

Turbulent Kinetic Energy

Transient Formulation

☐ Non-Iterative Time Advancement

☐ Frozen Flux Formulation

☐ High Order Term Relaxation

Solution Controls:

Solution Controls

Flow Courant Number

Explicit Relaxation Factors

Momentum

Pressure

Under-Relaxation Factors

Density

Body Forces

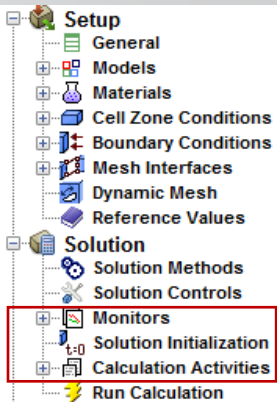
Volume Fraction

Turbulent Kinetic Energy

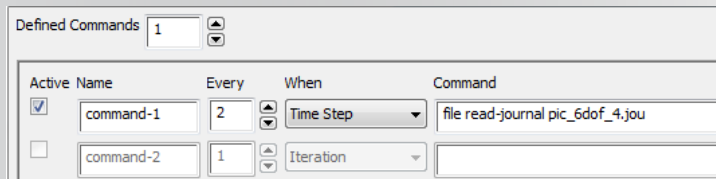
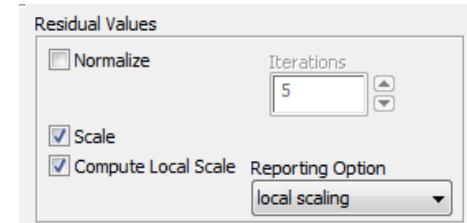
Specific Dissipation Rate

- These settings allows you to quickly “march” towards the equilibrium between force/moments and boat position.
 - It might however not be possible to start of with these settings for some cases. See slide 23 for more info

Simulation Settings



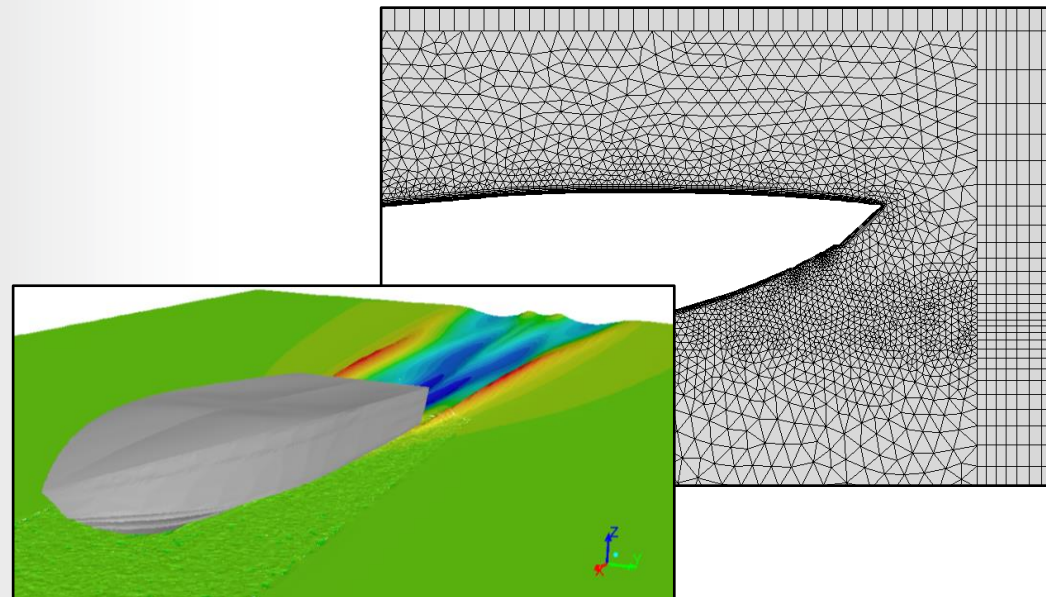
- It can be worth switching to local residual scaling for cases with largely varying cell sizes (Monitors/Residuals)
- Monitors will effectively allow you to follow the convergence behavior for your case
Define a:
 - Drag monitor with force vector in the flow direction
 - Surface monitor of the integral of pressure on the boat hull and plot it for each iteration
 - Surface monitor that shows the motion of the boat. E.g. vertex maximum of z-coordinate
- Initialize using standard initialization computed from inlet with open channel method “Flat”
- Define a calculation activity that prints out pictures of the VOF surface and mesh deformation
 - Example:



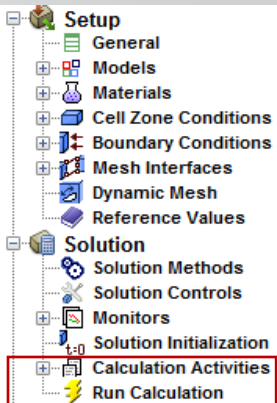
Pic_6dof_4.jou Journal file:

```

disp s-w 9
disp set r-o s-e-v yes yes
disp surf-m 2 10 16 ()
disp vi res-vi top-1
disp save-pic mesh1_%.png OK
disp set r-o s-e-v no
disp surf-m 0 ()
disp cont mix z-coord , ,
disp vi res-vi iso-3
disp save-pic iso3_%.png OK
    
```



Running the Simulation



- Achieving a stable simulation:

- Achieving stability can be very difficult for these simulations. The main reason is that the pressure on the boat hull (e.g. forces and moments) often reacts drastically when the boat is moved by the 6DOF solver. These pressure “peaks” does not fully disappear before updated forces are sent to the 6DOF solver, generating an even larger motion of the boat

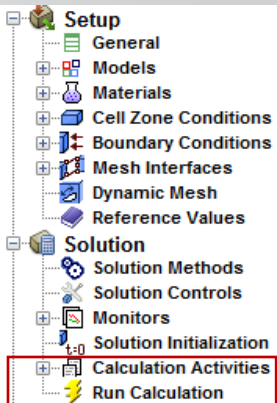
→ Case Diverges

- Tools for overcoming this instability:

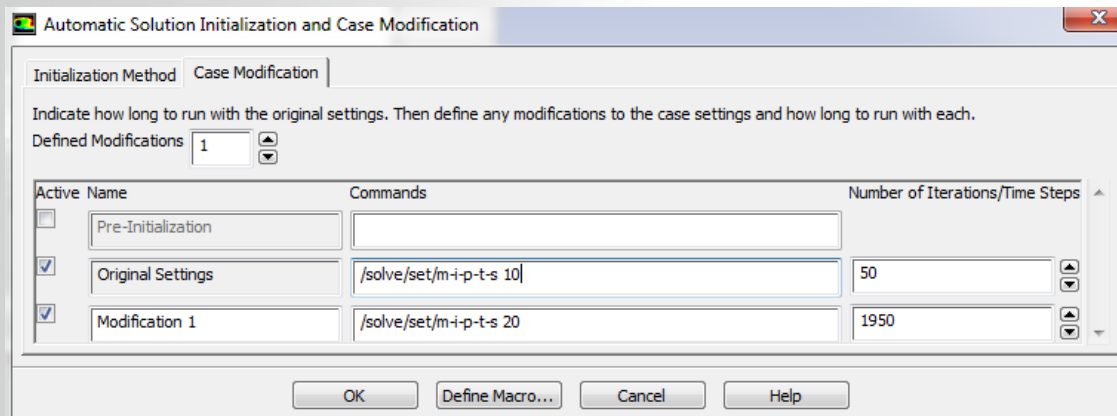
- Solution stabilization
- Implicit mesh update
- Compressible liquid
- Number of iterations
- Starting with lower order accuracy
- Initial position of the boat & initial flow

Discussed later

Running the Simulation

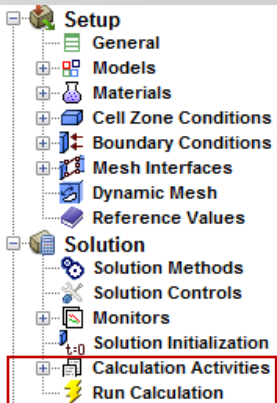


- This workshop case is stable with higher order schemes (from start) and constant water density. Implicit update is not used for this case either
- Solution Stabilization is however needed and will be needed for most cases
- What is also needed is to run a few initial time steps with fewer iterations/dt
 - This “ramping” can be automatically performed using “Automatic Case Modifications”

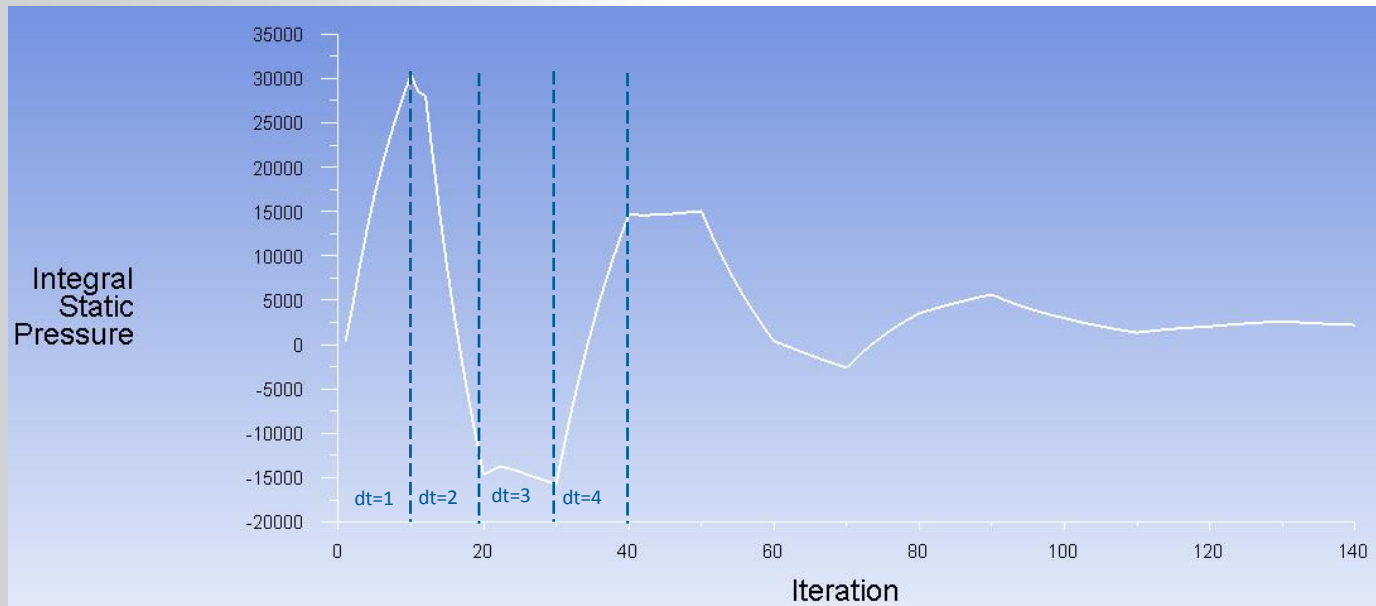


- 50 time steps with 10iterations/dt
- 1950 time steps with 20iterations/dt
- Time step size = 0.0025

Running the Simulation



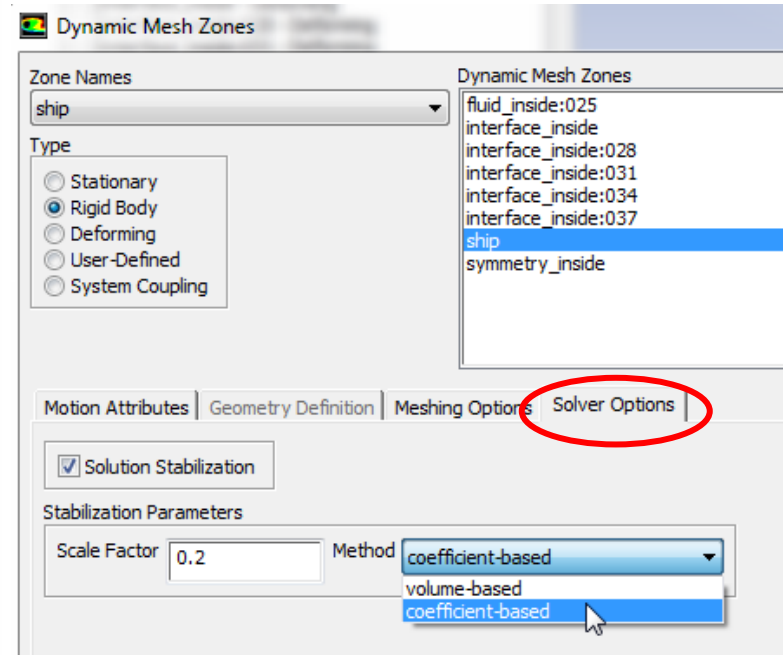
- Studying the pressure integral monitor on the boat hull for the initial time steps shows why it is effective to initially use less iterations/dt for this case



- Allowing more iterations/dt for the first time steps would lead to even larger “pressure peaks” and give a less stable start
- It is worth noting that we are not interested in fully resolving the transient behavior of this initial stage. What we want is to “march” towards the equilibrium between forces and boat position
- This startup will never be realistic anyway since we are exposing the boat to full speed directly in the first time step (similar to accelerating from initial boat position to full speed in one time step)

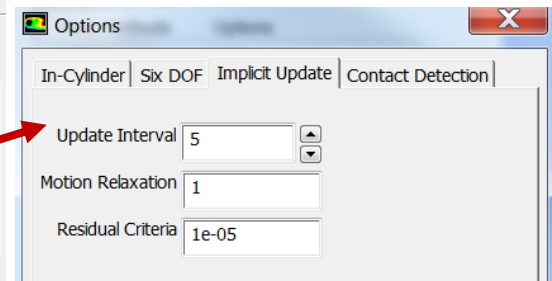
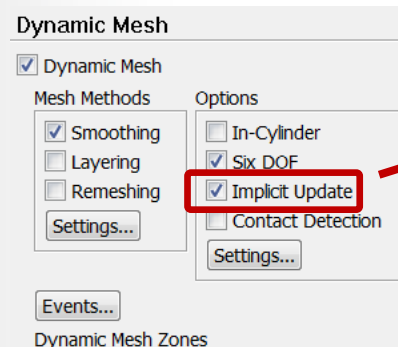
Solution Stabilization

- Solution Stabilization is available in Fluent to stabilize 6DOF and FSI cases. The option can be found in the Dynamic Mesh settings on the Solver Options panel
- The solution stabilization changes a coefficient used for the pressure update at the current non-linear iteration. This coefficient changes the pressure convergence behavior close to the moving wall. It does however not change the converged solution
 - For the volume-based option, the stabilization is a function of the cell volumes attached to the selected System Coupling/6DOF region
 - For the coefficient-based option, the stabilization is a function of the linear matrix coefficients in the continuity equation
 - Note that the volume-based and coefficient-based method will need different values to give similar stabilization
 - The following command activates stabilization for 6DOF cases in R15:
(rpsetvar 'dynamesh/sdof-solver-options? #t)



Implicit Update

- **Update Interval (UI)**
 - For UI = 1 the motion will update every iteration
 - For UI = 5 it will update the motion every 5th iteration
- **Motion Relaxation ω**
 - **Definition:** $x_k = \omega(x_{computed,k}) + (1 - \omega)x_{k-1}$
 - where k is the node position at iteration k.
 - The first computed motion at a new time step will never be under-relaxed, only the following motion updates
- **Residual Criteria**
 - If the motion requires more iterations to converge than the flow field, a warning will be displayed in the TUI



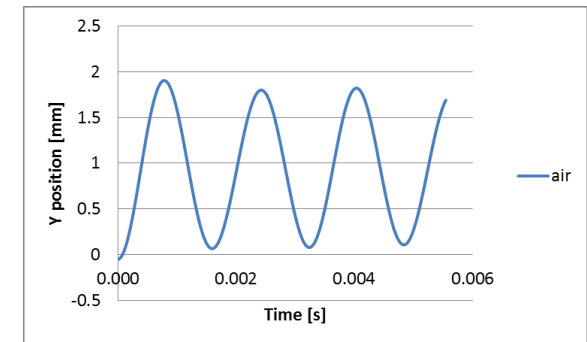
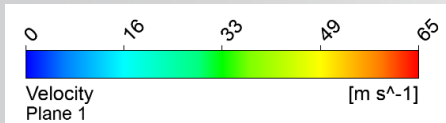
Example

Solution Stabilization & Implicit Update

- Example
 - A simple fluid domain with pressure outlet at both ends. The bottom surface is coupled to 6DOF where an external spring force is applied to the bottom surface. The spring is not at rest at $t = 0$
 - This case is straight forward to run when the fluid is compressible gas



Animation



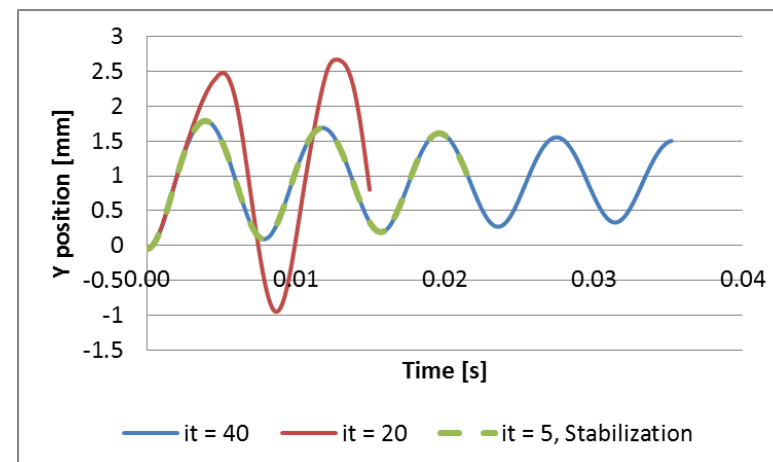
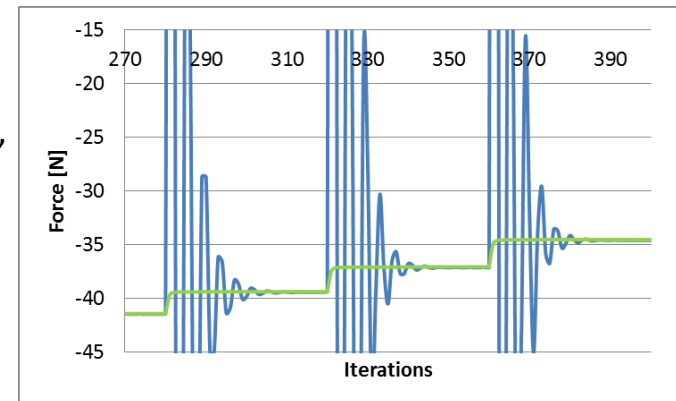
- Switching the fluid to water will create a strongly coupled case that will diverge
 - The wall motion will result in a pressure peak that won't converge to a realistic value before it's time to update the motion again → The case will "blow up"

Example Cont.

Solution Stabilization & Implicit Update

- **Solution stabilization**

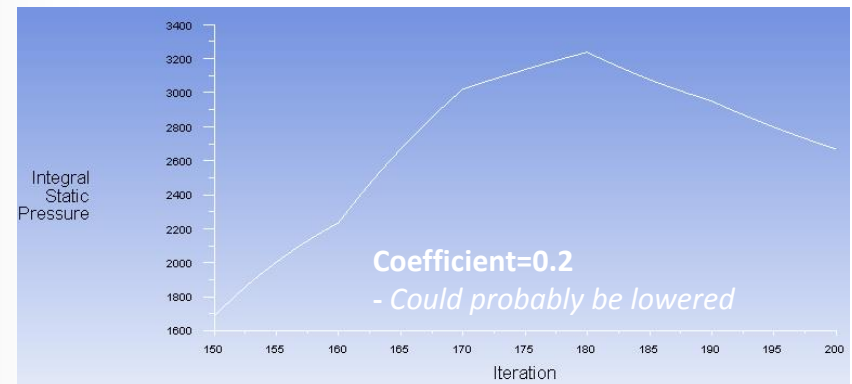
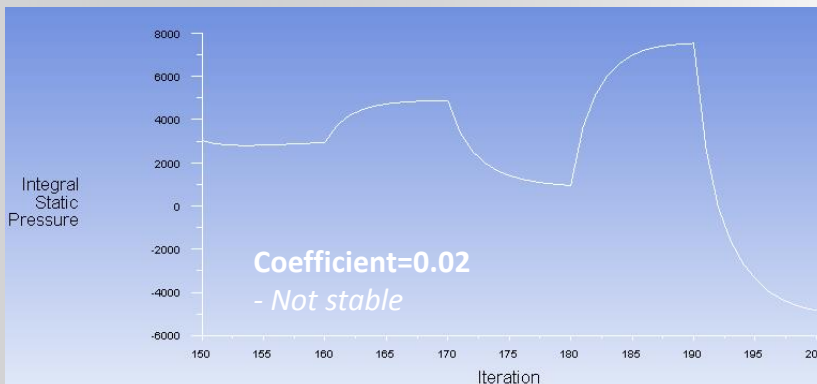
- **The graph shows the force convergence on the moving wall for 3 time steps (40 iter/dt)**
 - **The blue** graph is with implicit update (every iteration), motion relaxation = 0.1 and without stabilization
 - **The green** graph is with solution stabilization and implicit update (every it.), relaxation = 1 (no relaxation)
- **Both solutions will converge to the same result but solution stabilization will allow you to take less iterations and get a solution much faster**
- **The graph at the bottom right shows the final result from 3 simulations using the same time step**
 - **Red:** Implicit update, relax = 0.1, 20 iter/dt
 - **Blue:** Implicit update, relax = 0.1, 40 iter/dt
 - **Green:** solution stabilization and implicit update, relax = 1 and 5 iter/dt
- **This last graph clearly shows the advantage of using stabilization rather than relaxation**
 - Some cases won't even be possible to solve without stabilization



Comments on Stabilizing the Solution

- **Solution Stabilization**

- Is the most effective method to get rid of instabilities and enable a relatively quick progress towards the equilibrium position of the boat. The appropriate coefficient will unfortunately be case dependent and time step dependent.
- For most speed boat 6DOF simulations, the interest is to calculate the quasi steady state equilibrium boat position rather than the true transient behavior. For such cases it is not so important to fully converge each time step, hence the coefficient doesn't need to be “fully tuned in”
- A good practice regarding choosing the coefficient value is to use a value slightly higher than a value that gives divergence.
 - Example: Coefficient Based Solution Stabilization value = 0.02 gives an oscillating pressure behavior that eventually leads to divergence (or a too large motion of the boat). A value of 0.2 gives a slightly damped pressure response in each time step but not a pressure oscillation that increases over time. 0.2 would be a good value to use if a quasi steady state position is the goal for the simulation
 - If the time step is being increased later in the simulation you might need to increase the stabilization. Monitoring the pressure convergence on the boat hull for each iteration will show you if the solution starts to oscillate
 - A too high coefficient can lead to unphysical behavior close to the hull and eventually divergence the case



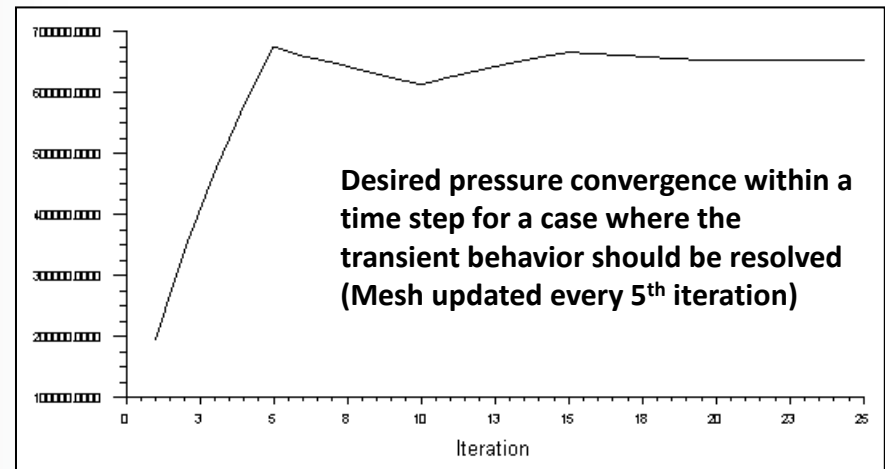
Comments on Stabilizing the Solution

- **Implicit Update**

- If your boat won't reach a quasi steady state equilibrium position but instead show tendencies to have an oscillating trim angle (and sinkage), you may want to carefully converge each time step to more accurately compute the amplitude and frequency of these oscillations
- For such cases it is important to update the boat position several times during the time step. Implicit update will enable this functionality
- Combining Solution Stabilization with implicit update will be the most effective way to reach convergence for each time step (relaxing the motion is less effective than solution stabilization, see example on slide 20)
- Using implicit update can also allow you to take bigger time steps

- **Compressible liquid**

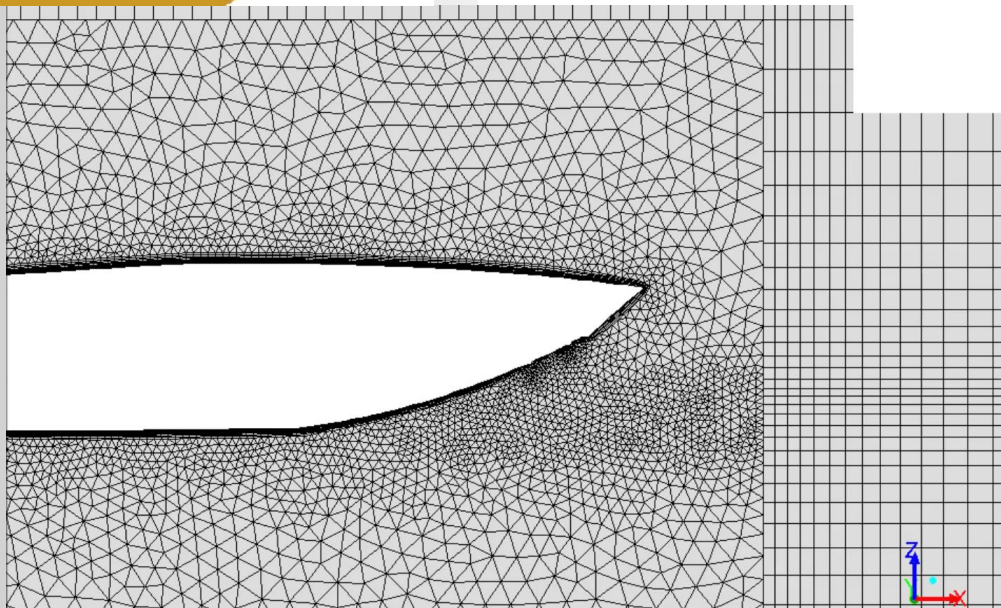
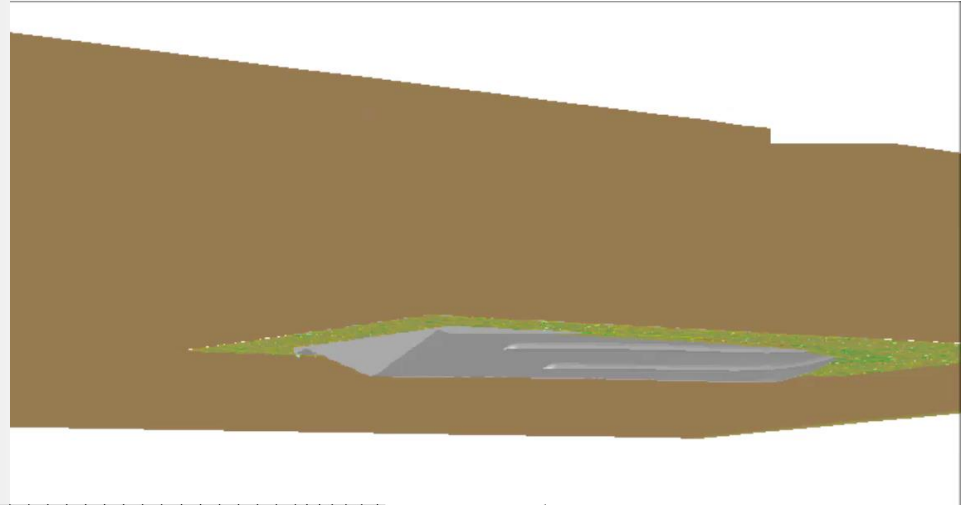
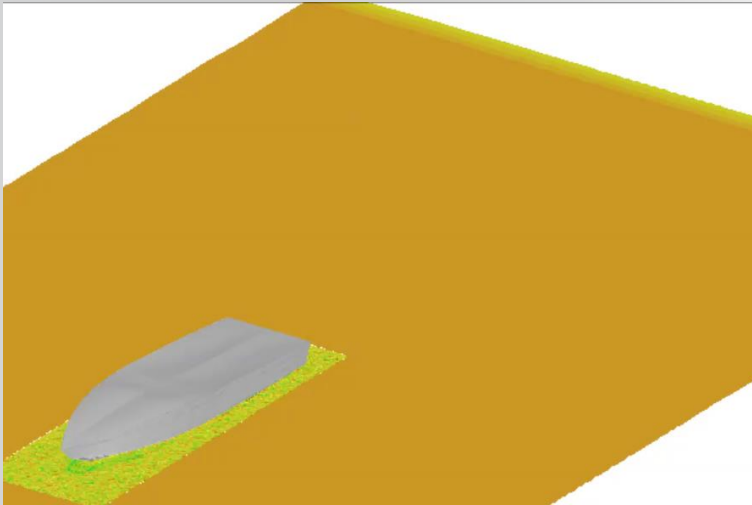
- Including a small compressibility for water can give some stability to pressure convergence for a moving rigid body in water. Compressible liquid is however rarely sufficient on it's own but can be successfully combined with solution stabilization



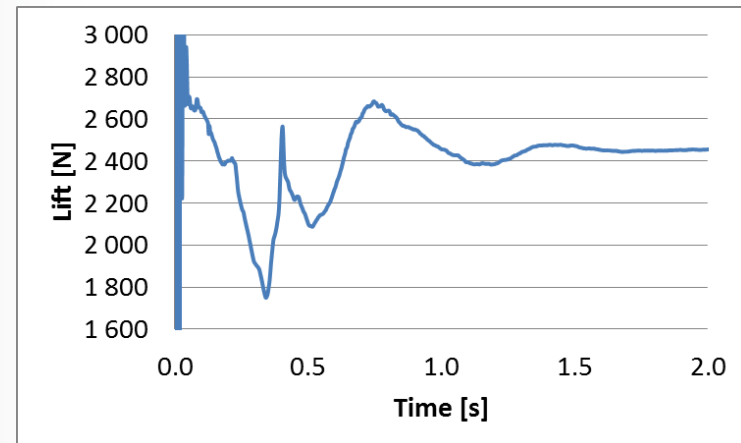
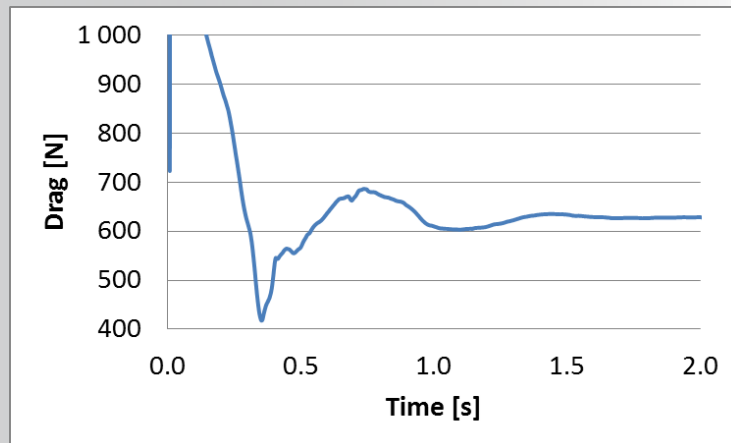
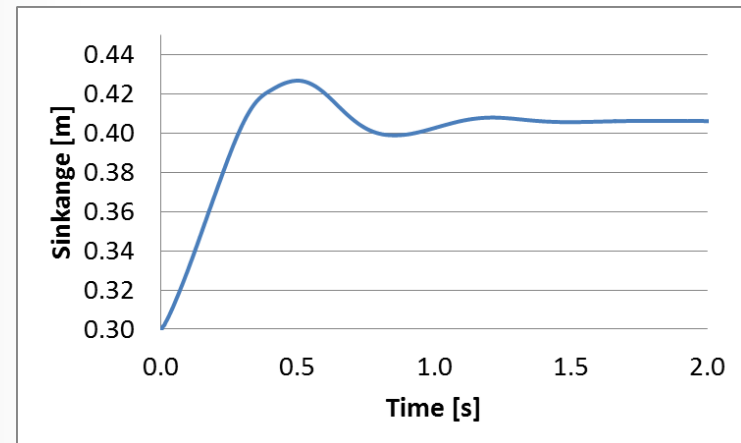
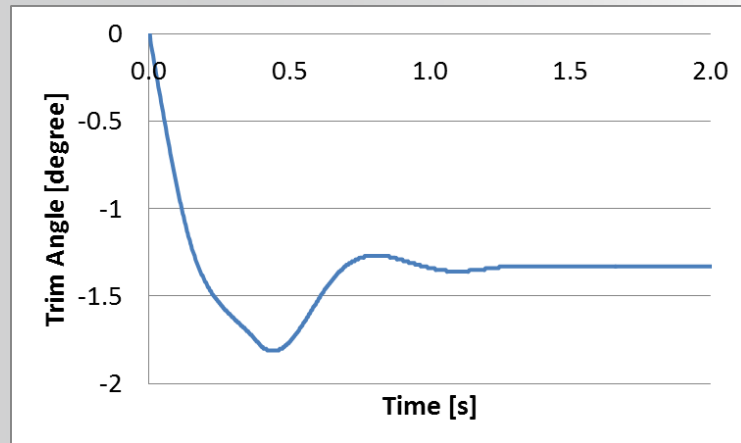
Comments on Stabilizing the Solution

- **Starting with lower order accuracy**
 - Some cases/meshes are not as easy to run as this one. Some may require lower order schemes together with under relaxation in the beginning of the run.
 - When running with lower order accuracy it is worth monitoring the VOF surface development around the boat. There is a risk that the surface will “follow” with the mesh smoothing (e.g. not converged down to a flat surface in front of the boat after the boat is tilted). This is normally not a problem when running with default order of accuracy
 - First order VOF scheme also tends to give a higher lift force on the boat resulting a too large motion
 - If lower order accuracy is needed initially it is good practice to ramp up the accuracy as soon as possible
- **Initial position of the boat & initial flow field**
 - The simulation will be simpler to run if the initial boat position in the water is close to the final planning position. This will give less motion of the boat and result in easier mesh smoothing & convergence
 - This workshop example starts from a boat position equivalent to 2/3 of the boat weight. A boat position equal to the full mass would result in much larger initial movement of the boat
 - Remember that these simulations starts from max speed directly in the first time step. Starting from a boat position at rest will therefor result in a very drastic initial motion
 - Some cases can benefit from starting from a developed flow field (e.g. a steady state simulation of a stationary boat)

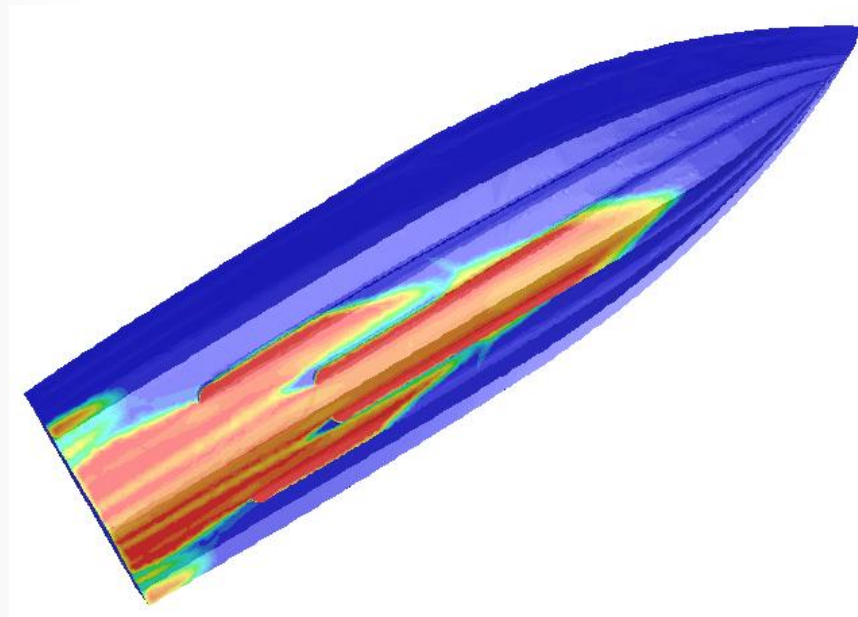
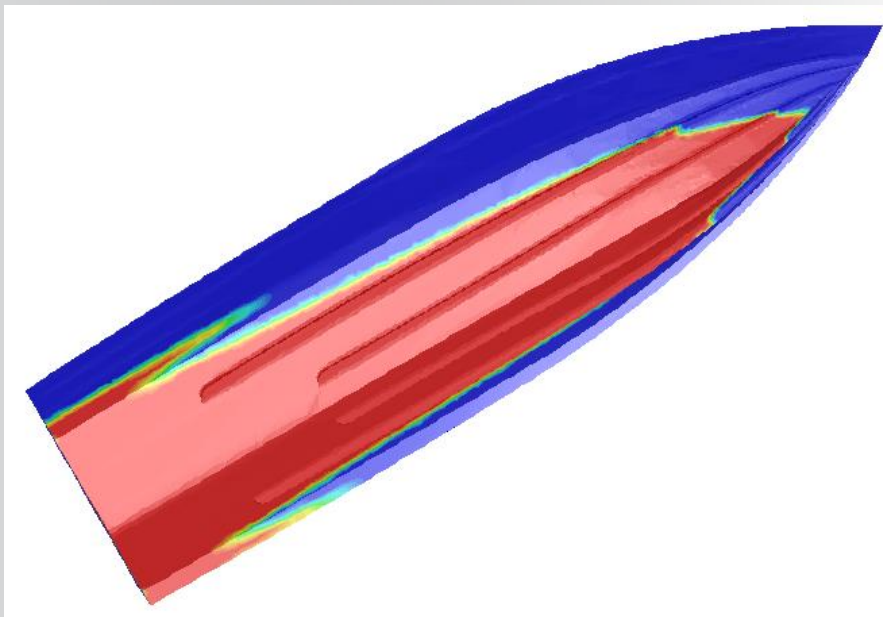
- Animations



- It took about 2s to reach the quasi steady state position for this case
 - Simulation time ~ 6hours on 64cores



- Initial and final water line



Files included with this workshop (ANSYS Version 16):

- **Boat_mesh.wbpz**
Archived Workbench project including geometry and mesh
- **msh_files.zip**
Exported .msh files of the hex zone and the tet zone (separate files)
- **new_boat_design_merged_meshes_no_settings.cas.gz**
Fluent case without settings but with merged mesh files
- **new_boat_design_30knots_init_all_settings.cas.gz**
Initial Fluent case with all settings
- **6DOF_clean.c**
6DOF UDF
- **pic_6dof_4.jou**
Journal for printing out pictures from the fluent case with all settings
- **Animation files (for the pdf version)**