Investigations and Guidelines for Maneuvering Simulations

Pierre Le Bihan, Benoit Mallol, Charles Hirsch

NUMECA International, Brussels/Belgium
benoit.mallol@numeca.be, charles.hirsch@numeca.be

1 Introduction

The ONRT test case for course keeping provides an excellent set of data to validate guidelines as it covers many features that CFD software should have, to perform accurate and robust maneuvering simulations in a relatively short time frame: overset grids, adaptive grid refinements, propellers and rudders motion controllers as well as internal wave generation with specific treatments at the boundaries have been used among interesting other features. Besides, the geometry details reflect the complexity often observed at an industrial level. The objective is to validate guidelines for this interesting case but also make them generic enough to be replicated in different maneuvering conditions.

Fig. 1: ONRT geometry

2 Case description

The ONR Tumblehome model is a preliminary design of a modern surface combatant ship. The ship is fully appended, including skeg and bilge keels. Twin spade rudders, and propellers with shafts and brackets are present in the model. Fig. 1 shows the ship model and Table 1 lists the main characteristics and dimensions of the ship. Free-running course keeping tests, in regular waves were performed at IIHR Hydraulics Wave Basin Facility and the results were reported (Sanada et al., 2013) for the Japan 2015 workshop.

Dimension	Full-scale	IIHR
Scale	1.0	1/49
Lpp[m]	154.0	3.147
Bwl[m]	18.78	0.384
D [m]	14.5	0.266
T [m]	5.494	0.112
Δ [kg]	8.507e6	72.6
Kxx [m²]	-	0.34Bwl
Kyy, Kzz [m²]	-	0.264Lwl

Table 1: Main characteristics of the ship

The simulations include a preliminary self-propulsion study and various heading angles: X =0° (head wave), 45°, 90°, 135° and 180°. Rudders are deflected with the following proportional controller: δ(t)= $K_p[\psi(t) - \psi_c]$ where δ is rudder angle, ψ is yaw angle, $\psi_c = 0$ is target yaw angle, and proportional gain $K_P = 1$, similarly to the IIHR's experiments. The ship movement is solved with 6 degrees of freedom (DOF) and the propeller revolution rate is determined by a preliminary self-propulsion study with the target approaching speed $U_0 = 1.11\ m/s$ $(Fr = 0.2)$ in calm water conditions.

To perform these simulations, FINE™/Marine CFD simulation suite is used including HEXPRESS™, an automated fully hexahedral unstructured mesh generator, together with the flow solver ISIS-CFD (developed by the CNRS and the Ecole Centrale de Nantes), a 3D unstructured flow solver able to simulate Euler or Navier-Stokes (laminar or turbulent) flows in a steady or unsteady way.

3 Methodology

3.1 Domains definition and meshing strategy

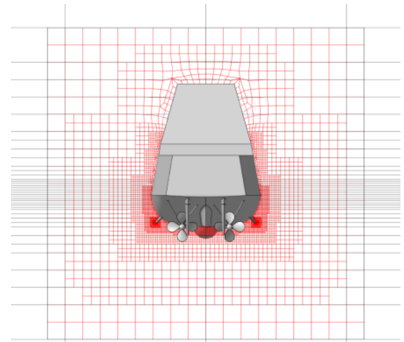

Fig. 2 : Mesh around the ship

The background domain is defined to simulate all the wave directions tested in the IIHR experiments, including following waves. In this configuration, the boat sails faster than the wave propagation speed. To avoid the situation where the ship catches up the wave front, a long enough domain is needed. As such, the size is selected to be the same as the actual IIHR towing tank (40m long, 20m wide). The domain is then considered as a background grid, fixed in the Earth reference frame, and the ship is located in an overlapping grid travelling through the towing tank. Another important benefit of this approach is that the ship's large motions induced by the waves can be handled without mesh deformation for all wave or maneuvering conditions. To make the overset as robust as possible some mesh criteria should be respected to improve the transfer of information at the interpolation location. The cell size ratio between overlapping and background domains should be equal to 1. Since the background domain is fixed, imposing a refinement at every possible location of the ship's grid would result in a too heavy mesh. To work around this issue, the mesh is dynamically refined during the simulation to respect this condition only at the overlapping locations, thanks to FINE™/Marine's adaptive grid refinement (AGR) technique. The AGR with the overset mesh continuity criterion is activated, and the size of the domains is set to get the exact same cell size at the overlapping boundary after this dynamic refinement. A free surface AGR criterion is also activated. The target cell size at the free surface is set to $\frac{L_{PP}}{1024}$, the recommended value for resistance calculation. These settings give 20 cells peak-to-peak, also matching with the ITTC (2011) recommendations for seakeeping simulations. For information, the total number of cells during the simulation with AGR is around 21.4M cells, whereas the mesh would have contained 36M cells with an equivalent static refinement box without AGR. For the ship grid, no initial grid refinement has been created at the free surface: AGR will also take care of it.

Since large relative motions of the appendages are expected and they are quite close to each other, an overset approach is used in the paper. In this case, overset method is not strictly necessary instead of sliding grids for the rudders but it validates this technique in case of strict necessity. In this situation where an overlapping grid is located inside another overlapping grid, a simple distance approach is used by the solver to determine the best interpolation location. That allows the rudder grids to cross the hull surface and simplify the meshing process. The grids are generated using HEXPRESS™ thanks to the grid convergence studies done by d'Aure et al. (2015) on seakeeping case and Zubova et al. (2016) for zigzag case for instance. The different numbers of cells per domain are detailed in Table 2.

Domain	Nb. of cells
Ship	3.7M
Rudders	1.15M
Propellers	2.26M
Background	7.42M
AGR	3.5M
Total	21.4M

Table 2: Cells repartition

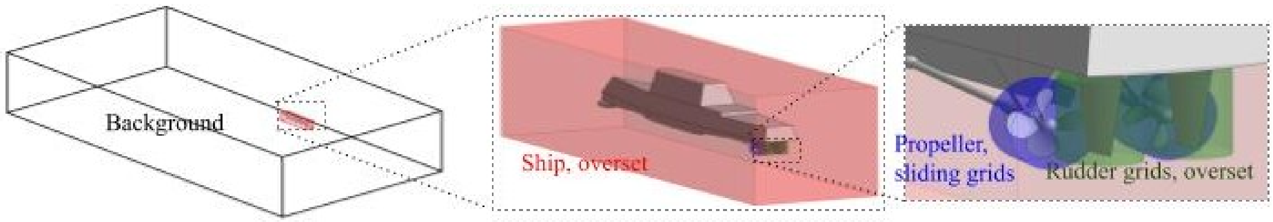

Fig. 3: Domains definition

3.2 Turbulence modeling

Giving the expertise of the CFD marine community with $k - \omega$ (SST Menter) and all the validations performed with FINE™/Marine with this model, it will be used together with wall function. This choice is also justified since DES model is still too costly and Zubova et al. (2017) showed that EASM returns a larger error compared to $k - \omega$ (SST Menter) in ship maneuvering simulations with flow separations.

3.3 Time step selection

The choice of the time step value is of high importance to avoid CPU time penalties. Four main constraints can be identified and Table 3 summarizes the possible values. The time step values for the rudder rate and propeller rotational speed should ensure a Courant number below 0.3 at the interface between domains. Thus the mesh cell size should always be taken into account for the time step value selection. Besides, a rotating frame method (RFM) can be used to calculate the flow inside the rotating sliding grids of the propellers. The flow in each rotating cell is solved using the rotating reference frame equations but it does not account for the relative motion of the adjacent domains of simulation. This method is a steady-state approximation and its usage can be justified since the rudder angles are relatively small and should not be too much impacted by the incoming flow perturbations which are aligned with the rudder position. However, it should not be used in case of zigzag or turning circle simulations. The time step value can then be calculated such that the propeller will perform a complete rotation in 20 time steps. The time step value is 27 times higher with a RFM approach than without. Another approach to model the propeller is the actuator disk. However, the RFM approach doesn't require the performance curve of the propeller, which is required for self-propulsion simulation. This is a great advantage when those curves are not provided. In such a situation, no preliminary computations have to be performed to predict the full open water data.

Criterion	Value [s]
The ships advancing speed	1.42e-2
The rudder rate	2.02e-2
The propeller rotational speed without RFM	2.07e-04
The propeller rotational speed with RFM	5.61e-3
The waves period	1.42e-2

Table 3: Possible time step value

3.4 Wave generation

The wave parameters follow the experimental setup: regular waves with a wavelength equal to ship length ($\lambda = L_{wl}$) and a wave steepness (H / λ) of 0.02. An Internal Wave Generator (IWG) is used to generate the wave directly inside the domain based on additional momentum source terms applied to the Navier-Stokes equations, which is numerically more natural than generating waves at boundaries. To generate the wave field with a reduced computation time, the simulation is made of two different steps. First, an internal wave generator covering the entire domain inflates the waves. Contrary to other methods for initialization, IWG approach initializes not only the volume fraction but the velocity field and pressure field as well. Using this method, waves are initialized everywhere in the domain within one wave period (1,42s). Fig. 4 shows the wave field during the inflating phase at 3 different times. After this initialization, a standard internal wave generator, with a width of 1 wavelength is used near the left border. A 3 wavelength width forced layer zone is set on the opposite side of the towing tank. It imposes the exact wave signal from the IWG to remove any wave reflections from solid bodies or domain boundaries.

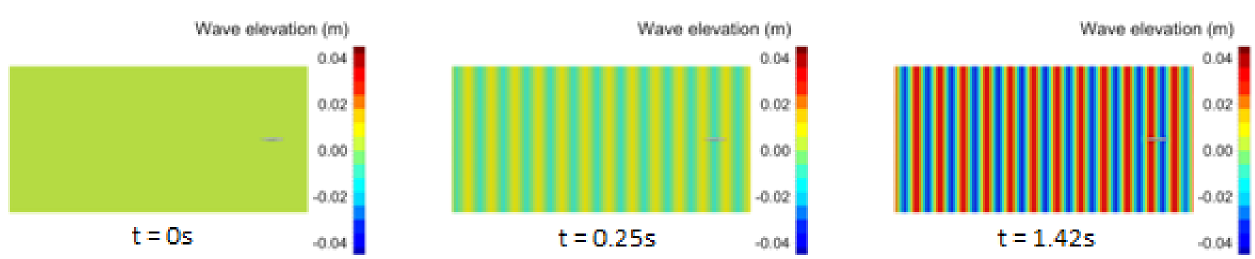

Fig. 4: Wave field at different time of the inflating phase

3.5 Self-propulsion controller

The first simulation is a preliminary self-propulsion study in calm water conditions. The objective is to determine the propeller rotational speed balancing forces when the ship is sailing at $U_0 = 1.11\ m/s$. The FINE™/Marine's code dedicated to self-propulsion study has been extended to handle multi-propeller cases and to control the rudders at the same time in order to keep the desired heading.

This self-propulsion controller cancels the forces projected onto the X-axis by adjusting the propeller rotational speed, with an imposed ship speed. The controller was developed to ensure stability, robustness, and speed, but most of all, to be valid for any kind of configuration. Extensive tests were performed to determine the default parameters allowing the controller to have this general behavior. Hence, the default parameters for the controller were used to study the self-propulsion. For a faster convergence, a first steady acceleration phase is performed. The ship is accelerated up to its target speed U_0 and the propeller to an estimated rotational speed. Herein the value found by Sanada et al. (2013) is taken as an initial value. Only trim and sinkage are solved. In the second phase, the propeller controller is activated and the 5 dof are solved. Although a pure sliding-grid approach might capture more physical phenomena and so return a result closer to the experiments, the objective of this computation is to find the propulsion point for the upcoming free-running simulations. Since the propellers will be modeled using a RFM in those computations, the same model should be used in the self-propulsion study.

4 Results and validation

4.1 Preliminary self-propulsion study

The predicted sinkage and trim (shown in Table 4) agree well with the available experimental data, the error is below 0.2mm for the sinkage and 0.02° for the trim. The self-propulsion controller behaves as expected, forces are balanced.

	CFD	EFD (IIHR)	Difference
n [rad/s]	56.43	56.36	0.07
Sinkage [m]	0.0025	0.0023	0.0002
Pitch [deg]	-0.018	-0.039	0,021

Table 4: Self-propulsion results

Fig. 5 shows the time evolution of the propeller rotational speed. The computed self-propulsion point is achieved for a propeller rotational speed of 56.43 rad/s, compared to the experimental data of 56.36 rad/s. The fact that the propeller revolution rate from experiments almost balances the forces indicates that the flows around the hull, the rudders, and the propeller, as well as their interaction, are well captured. It proves the feasibility of self-propulsion simulation using the present approach and provides a good starting point for the next course keeping simulations.

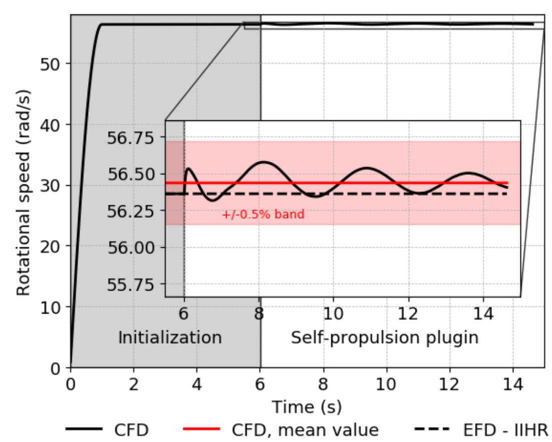

Fig. 5: Time history of propeller's revolution rate

4.2 Course keeping in wave

Fig. 6 shows the motion of the ship with all the wave angles. The solid line represents the computed results and the crosses the experimental data presented by Sanada et al. (2013). They are synchronized with $t = 0$ when the first wave peak reaches the bow. In contrast with the towing tank experiments, the present simulations start, with a wave field initialized everywhere in the domain thanks to the wave inflating method performed at the initialization step. That is the reason why the results are shown after 5 wave periods when both numerical and experimental data have reached this steady oscillating state. That explains the offset in heading angle for $X = 45°$ and $X = 90°$.

For all the wave directions the motions (yaw, pitch and roll) are well predicted. For the cases $X = 0^\circ$, $X = 45^\circ$ and $X = 90^\circ$, the amplitude of motion differs from the experiments by less than 5%. Bigger differences are obtained for the $X = 135^\circ$ and $X = 180^\circ$.

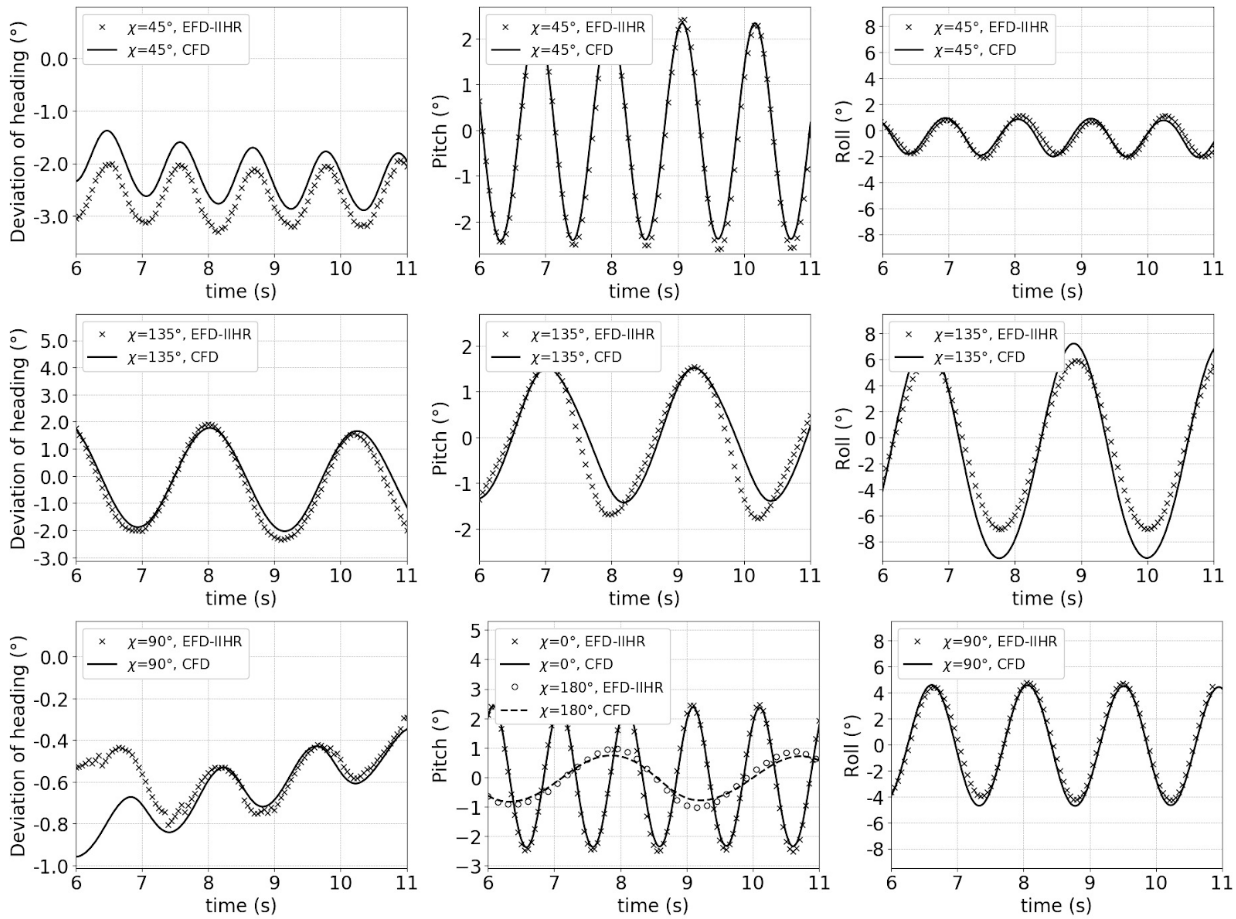

Fig. 6 : Computed ship motion and IIHR's experiments

Fig.7 shows the surge speed for heading and following waves. The global trend is well captured by CFD for the oscillation period and the amplitude especially for following waves. However, the heading wave signal amplitude is different: a sinusoidal signal is obtained in experiments whereas CFD predicts a flat signal at the peak locations. Similar behavior has been reported by other CFD users and remains unexplained for the moment. Wang et al. (2017) suggest that the propeller revolution rate is not kept exactly constant in the experiments. The time history of the propeller speed, thrust and torque from the experiment is not available to confirm this assumption.

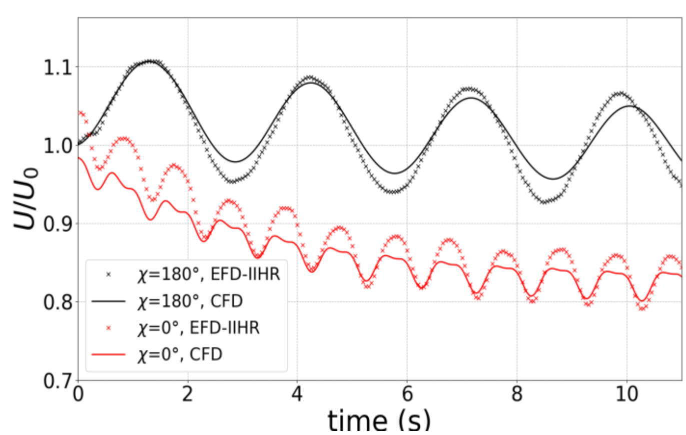

Fig. 7: Time history of surge velocity for 0° and 180° waves

5 Conclusions

Many CFD techniques have been used and validated at the same time to ensure satisfactory maneuvering simulations. For instance, adaptive grid refinement seems to be well indicated to reduce the CPU cost and ensure the best interpolation at any moment of the simulation. The results are accurate for many variables at a reasonable CPU cost. Indeed, the total cost of the simulation for 1 wave angle (including wave initialization) using 240 cores is around 104 clock hours (25×10^3 CPU hours). The extension of the self-propulsion controller allows to adapt the propeller's rotational speed and the rudder angles. The RFM method helps to significantly decrease the required CPU time but should not be used for maneuvering conditions such as turning circle and zigzag. Current investigations are also going towards a dynamic time step based on the relative motion of each grid to ensure the proper transfer of information and to accelerate the computation. Structured meshes will also be investigated to make sure forces are always well computed on propellers and rudders at large angles.

References

Y. Sanada, K. Tanimoto, K. Takagi, Y. Toda and F. Stern (2013). Trajectories and Local Flow Field Measurements around ONR Tumblehome in Maneuvering Motion. Ocean Engineering, Vol. 72, pp. 45-65

B. d'Aure, B. Mallol and C. Hirsch (2015). Resistance and seakeeping CFD simulations for the korean container ship. Proceeding of NuTTS 2015, Marstrand, Sweden.

A. Zubova, A. del Toro Llorens, B. Mallol and C. Hirsch (2016). Progress towards CFD guidelines for zig-zag simulation in waves. Proceeding of NuTTS 2016, St. Pierre d'Oleron, France.

A. Zubova, A. del Toro Llorens, B. Mallol and C. Hirsch (2017). Extended CFD Guidelines for Zigzag Simulations in Self-propulsion Conditions at Low Froude Number. Proceeding of NuTTS 2017, Wageningen, The Netherlands.

J. Wang, L. Zou, D. Wan (2017). CFD simulations of free running ship under course keeping control. Ocean Engineering, 141, 450–464

ITTC – Recommended Procedures and Guidelines (2011).