

# NUMERICAL PREDICITON OF FREE RUNNING AT MODEL POINT FOR ONR TUMBLEHOME USING OVERSET GRID METHOD

Jianhua Wang<sup>1</sup>, Xiaojian Liu<sup>1,2</sup> and Decheng Wan<sup>1\*</sup>

1. State Key Laboratory of Ocean Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Collaborative Innovation Center for Advanced Ship and Deep-Sea Exploration, Shanghai 200240, China

2. Marine Design & Research Institute of China, Shanghai, 200240, China

\*Corresponding author: dcwan@sjtu.edu.cn

## 1. SUMMARY

The present work is focused on the numerical prediction of free running at model point in calm water for ONR Tumblehome model. The fully appended ship model is under free running conditions with rotating propellers to achieve its target speed and rudders to keep its straight running position. Simulated result, i.e. the rate of revolutions of propeller  $n$  is compared with experimental data carried out by IIHR basin.

All the computations are carried out by our in-house solver naoe-FOAM-SJTU, which is developed on the open source platform OpenFOAM and mainly composed of a dynamic overset grid module and a full 6DoF motion module with a hierarchy of bodies. With dynamic overset grids handling with the rotating components, Unsteady Reynolds Averaged Navier-Stokes (URANS) computations are carried out for the self propulsion case. The HPC center in the school of naval architecture, ocean & civil engineering, Shanghai Jiao Tong University is used for all the calculations with meshes up to 6.5 million cells.

## 2. INTRODUCTION

In the past few years, ship maneuverability has become more and more important for navigational safety, thus an accurate estimation of a ship's maneuvering ability especially for free running test at the design stage is very important. Up to the present, the main method for predicting ship maneuvering is model scale experiments in towing tanks. The reports of the ITTC Committees give the guidelines and recommended procedures in towing tank experiments and recently also in CFD calculations. The main purpose of this research is to investigate the capability of our in-house solver for ship maneuvering prediction especially for self propulsion with rotating propellers and

rudders. The present work is a contribution to the Tokyo 2015 workshop on CFD in Ship Hydrodynamics and gives the comparison between the CFD based simulation results and free running tests performed by participating towing tanks.

### 2.1 ONR Tumblehome Ship Model

The ONR Tumblehome model 5613 is a preliminary design of a modern surface combatant, which is fully appended with skeg and bilge keels. The model also has rudders, shafts and propellers with propeller shaft brackets. Fig. 1 shows the ship model and Table 1 lists the principle dimensions of the ship.



Fig. 1 ONR Tumblehome Ship

Table 1 Principle Dimensions of ONR Tumblehome ship

Dimension	Full-Scale	IIHR
Scale	1.00	1/49.0
$L_{pp}$ (m)	154.0	3.147
$B_{wl}$ (m)	18.78	0.384
$D$ (m)	14.5	0.266
$T$ (m)	5.494	0.112
$\Delta$ (kg)	8.507e6	72.6

### 2.2 Overviews

In this paper, the numerical methods including governing equations, solver and algorithm and overset grid method will be introduced first. Then the next section will describe the computational overviews, where the computational

domain, mesh configuration, PI controller for rotating components and test conditions will be described in detail. Following this is the numerical results and analysis, where the numerical results of free running tests including hydrodynamic force coefficients for ship hull and moving components will be presented. Additionally the results will be compared with the experimental data. Finally, a conclusion of the paper is drawn.

### 3. NUMERICAL METHODS

#### 3.1 Governing Equations

Navier-Stokes equations are generally used to describe the motion of fluid continuum. In terms of the unsteady incompressible fluid, the governing equations adopted here is the Unsteady Reynolds-Average Navier-Stokes (URANS) equations coupled with the volume of fluid (VOF) method. The equations can be written as a mass conservation equation and a momentum conservation equation, which are listed below:

$$\nabla \cdot \mathbf{U} = 0 \quad (1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot [(\rho \mathbf{U} - \mathbf{U}_g) \mathbf{U}] = -\nabla p_d - \mathbf{g} \cdot \mathbf{x} \nabla \rho \quad (2)$$

$$+ \nabla \cdot (\mu_{eff} \nabla \mathbf{U}) + (\nabla \mathbf{U}) \cdot \nabla \mu_{eff} + f_\sigma + f_s$$

Where  $\mathbf{U}$  is the velocity field and  $\mathbf{U}_g$  is the velocity of mesh points;  $p_d$  is the dynamic fluid pressure;  $\rho$  is the mixture density of the two-phase fluid;  $\mathbf{g}$  is the acceleration due to the gravity;  $\mu_{eff} = \rho(\nu + \nu_t)$  is the effective dynamic viscosity, in which  $\nu$  and  $\nu_t$  is the kinematic viscosity and kinematic eddy viscosity respectively; the latter is obtained by the SST  $k-\omega$  turbulence model (Menter, 1994);  $f_\sigma$  is the surface tension term in two phases model and  $f_s$  is the source term for sponge layer.

The volume of fluid (VOF) method with artificial compression technique is applied for locating and tracking the free surface (Hirt and Nichols, 1981). In the VOF method, each of the two-phase is considered to have a separately defined volume fraction ( $\alpha$ ), where 0 and 1 represent that the cell is filled with air and water respectively and  $0 < \alpha < 1$  stands for the interface between two-phase fluid. The density and dynamic viscosity for the mixed fluid can be presented as:

$$\rho = \alpha \rho_1 + (1 - \alpha) \rho_2 \quad (3)$$

$$\mu = \alpha \mu_1 + (1 - \alpha) \mu_2$$

The volume fraction function can be determined by solving an advection equation:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot [(\mathbf{U} - \mathbf{U}_g) \alpha] + \nabla \cdot [\mathbf{U}_r (1 - \alpha) \alpha] = 0 \quad (4)$$

where the last term on the left-hand side is an artificial compression term to limit the smearing of the interface and  $\mathbf{U}_r$  is a relative velocity used to compress the interface.

#### 3.2 Solver and Algorithm

The solver employed in this paper is naoe-FOAM-SJTU (Shen, Zhao, Wang and Wan, 2014), which is based on the open source CFD code OpenFOAM. It is developed to deal with complex motion problems such as large-amplitude ship motion in ship maneuvering and seakeeping, moving rudder and rotating propeller in ship self-propulsion. During the calculation, the RANS equations and VOF transport equation are discretized by the finite volume method (FVM), and for the discretized RANS equations, the merged PISO-SIMPLE (PIMPLE) algorithm is adopted to solve the coupled equation of velocity and pressure. The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm allows to couple the Navier-Stokes equations with an iterative procedure and the Pressure Implicit Splitting Operator (PISO) algorithm enables the PIMPLE algorithm to rectify the pressure-velocity correction. More detailed description for the SIMPLE and PISO algorithm can be found in Ferziger and Peric (1999) and Issa (1986). Additionally, several built-in numerical schemes in OpenFOAM are used in solving the PDE. The convection terms are solved by a second-order TVD limited linear scheme, and the diffusion terms are approximated by a second-order central difference scheme. Van Leer scheme (Van Leer, 1979) is applied for VOF equation discretization and a second-order backward method is applied for temporal discretization.

#### 3.3 Overset Grid Technique

Overset grid (or Chimera grid) is a grid system made up of blocks of overlapping structured or unstructured grids. In a full overset grid system, a complex geometry is decomposed into a system of geometrically simple overlapping grids. Boundary information is exchanged between these grids via interpolation of the fluid variables. Through this way overset grid method removes the restrictions of the mesh topology among different objects and allows grids move independently within the computational domain, and can be used to handle with complex problems listed above in the field of ship and ocean engineering.

The overset grid module is based on the numerical methods from OpenFOAM including the cell-centered scheme and unstructured grids. SUGGAR++ program is utilized to generate the domain connectivity information (DCI) for the overset grid interpolation. With the overset grid capability a full 6DoF motion module with a hierarchy of bodies is utilized, allowing the ship and its appendages to move simultaneously. Two coordinate systems are used to solve the 6DoF equations. One system is inertial (earth-fixed system  $o'x'y'z'$ ) and the other is non-inertial (ship-fixed system  $oxyz$ ). The inertial system can be fixed to earth or move at a constant speed respect to the ship. The

non-inertial system is fixed to the ship and translates and rotates according to the ship motions. Details of the 6DoF module with overset grid technique can be found in Shen (2015)

#### 4. COMPUTATIONAL OVERVIEWS

##### 4.1 Computational Domain

With dynamic overset grid technique, here we have four part of the computational grids, i.e. grid around ship hull, port and starboard grid for both propeller and rudder, the last one is the background grid. The four part grids have overlapping areas for information exchange. The computational domain arrangement around ship hull is shown in Fig. 1

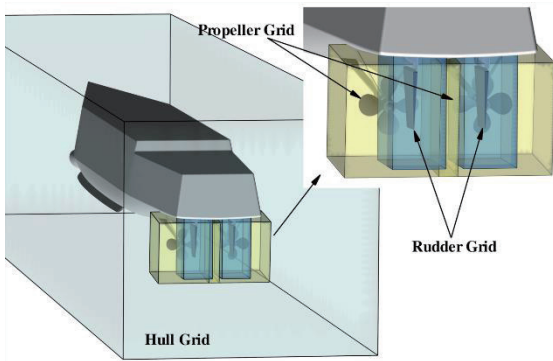


Fig. 2 Computational Domain around Ship

The rudders and propellers are able to rotate with respect to the ship hull and provide the thrust forces and turning moments for the ship. The background domain extends to  $-1.5L_{pp} < x < 5.0L_{pp}$ ,  $-1.5L_{pp} < y < 1.5L_{pp}$ ,  $-1.0L_{pp} < z < 0.5L_{pp}$ , and the hull domain is much smaller with a range of  $-0.2L_{pp} < x < 1.2L_{pp}$ ,  $-0.2L_{pp} < y < 0.2L_{pp}$ ,  $-0.1L_{pp} < z < 0.1L_{pp}$ .

##### 4.2 Mesh Configuration

All of the meshes used in this paper are generated by snappyHexMesh, a mesh generation tool provided by OpenFOAM. The total cell number is 6.46 M and the detail information for each part is show in Table 2. The local mesh distribution around ship hull is shown in Fig. 3

Table 2 Grid Information in Each Part

Grid	total	port	starboard
Total	6.46M	--	--
Background	0.99M	--	--
Hull	2.61M	--	--
Propeller	2.28M	1.14M	1.14M
Rudder	0.58M	0.29M	0.29M

##### 4.3 PI controller for Moving Components

For the self propulsion case, ONRT is equipped with two directly rotating propellers, which provide the thrust for the ship to advance, and two controller rudder for course-keeping. The free running computation is carried out at the model point following the experimental procedure.

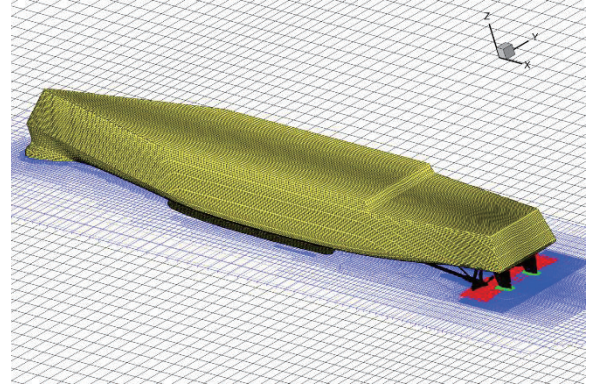


Fig. 3 Mesh Arrangement around Ship

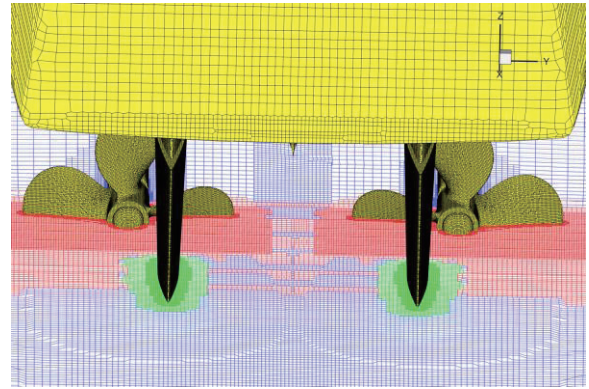


Fig. 4 Local Mesh for Propeller and Rudder

A proportional-integral (PI) controller is employed to adjust the rotational speed of the propeller to achieve the desired ship speed. The instantaneous RPS of the propeller is obtained as:

$$n = Pe + I \int_0^t e dt \quad (5)$$

where  $P$  and  $I$  are proportional and integral constants, respectively, and  $e$  is the error between target ship speed and instantaneous speed:

$$e = U_{\text{target}} - U_{\text{ship}} \quad (6)$$

The PI controller is activated at the beginning of the computation. The rate of revolutions of the propeller  $n$  is to be adjusted to obtain force equilibrium in the longitudinal direction:

$$T = R_{T(SP)} \quad (7)$$

where  $T$  is the computed thrust,  $R_{r(SP)}$  is the total resistance at self propulsion in calm water.

Rudders are deflected with following proportional control equation:

$$\delta(t) = K_p (\psi(t) - \psi_c) \quad (8)$$

where  $\delta(t)$  is rudder angle,  $\psi(t)$  is yaw angle,  $\psi_c = 0$  is target yaw angle, and proportional gain  $K_p = 1.0$ . Rudder angle update at the end of each time to achieve the target yaw angle.

#### 4.4 Test Conditions

The present work is for self propulsion simulation of ONR Tumblehome model. According to the case setup, the fully appended ship is set to free running at model point in calm water with rotating propellers and rudders. The approaching speed is  $V_A = 1.11$  m/s ( $F_r = 0.20$ ), and target yaw angle is  $\psi_c = 0$ .

The boundary conditions are identical with zero velocity and zero gradient of pressure imposed on inlet and far-field boundaries. The boundary condition of interface between each two grids is set to overlap for later information exchange. The propeller grids are fitted behind the ship hull and rotating about the propeller shaft, and the rudder grids are rotating about rudder axis. Notice that there is a physical gap between the propeller and the hull, as well as the rudder and hull to prevent occurrence of orphans in this region.

When dealing with self propulsion problems, the initial condition for the computation is interpolated from the final solution of the towed condition with the utility *mapFields* supported by OpenFOAM. This pre-processing step can save a lot of computational time by starting with a developed flow field and boundary layer. The initial ship speed was set to the target cruise speed of 1.11 m/s and the propeller was static. The proportional and integral constants of the PI controller were set to  $P = 800$  RPS·s/m and  $I = 600$  RPS/m. Large PI constants accelerate the convergence of the propeller revolution rate and reduce the total computation time, but if too large they cause overshoots in propeller RPS and ship velocity.

### 5. NUMERICAL RESULTS AND ANALYSIS

All the numerical results and analysis are based on the non-dimensional parameters, which are made by advancing speed  $U$ , water length  $L_{WL}$ , fluid density  $\rho$ , gravitational acceleration  $g$ , and the dynamic viscosity  $\mu$ . The non-dimensional expressions for Froude number and Reynold number are:

$$F_r = U / \sqrt{gL_{WL}} \quad (9)$$

$$R_e = \rho U L_{WL} / \mu \quad (10)$$

All CFD predicted force coefficients are using the provided wetted surface area at rest  $S_0$ , propeller diameter  $D_p$ , and propeller rate of revolution. Force coefficients are defined as follows:

$$C_T = \frac{R_T}{\frac{1}{2} \rho U^2 S_0} \quad (11)$$

$$K_T = \frac{T}{\rho n^2 D_p^4} \quad (12)$$

$$K_Q = \frac{T}{\rho n^2 D_p^5} \quad (13)$$

CFD based ship motions are also in the non-dimensional format by the following equation:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} / V_A \quad (14)$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \times \frac{L_{WL}}{V_A} \quad (15)$$

where  $\phi$  is roll angle,  $\theta$  is pitch angle,  $\psi$  is yaw angle.

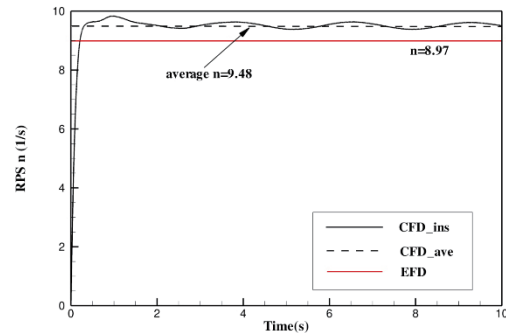


Fig. 5 Time History of RPS

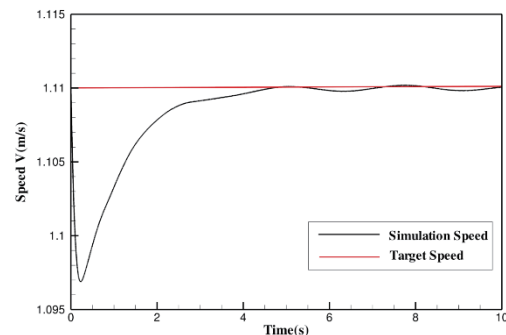


Fig. 6 Time History of Ship Speed



The time histories of RPS and ship speed are shown in Fig. 5 and Fig. 6 respectively. The computation was run for 10 s of model scale time. Both figures show a close up of the convergence history of ship speed and RPS during the first 5 s. The RPS starts at 0 and increases significantly. The ship speed starts at the target speed (1.11 m/s) and initially drops because the propeller rate is accelerating and at the beginning there is not enough thrust to maintain the speed. As the RPS increases, the ship speed reaches the lowest speed and returns to the target slowly. After 2 s, the ship speed has reached the target speed and the RPS converged to the self-propulsion point.

Table 3 lists the ship motions of self-propulsion obtained by the CFD computations and experimental data.

Table 3 Comparison of Ship Motions

Parameters		EFD	CFD
u	Value	1.01E+00	1.00
	E%D	--	1.5
sinkage $\sigma \times 10^2$ (m)	Value	2.26E-01	2.43E-01
	E%D	--	-7.4
trim $\tau$ (deg)	Value	-3.86E-02	-6.76E-02
	E%D	--	-75.3

From Table 3 we can see that CFD based method can precisely achieve the desired speed and the computational result of sinkage is 7.4% bigger than the experimental data while the comparison error of trim is up to 75% since the absolute value is very small.

As with the motions of other degrees of freedom, the amplitude is very small from our simulated results, especially for the yaw motion, which indicates that there is no need to move the rudders in the 10s simulated time to meet its coursekeeping demand.

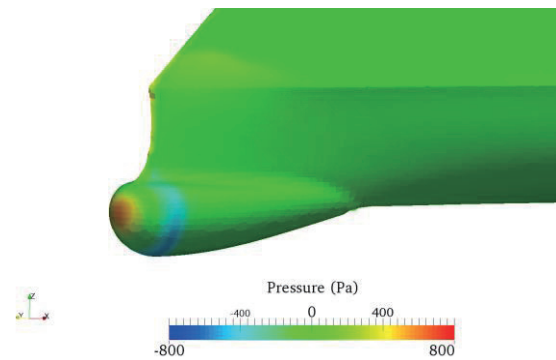
Table 4 shows the propulsive coefficients of the free running simulation. Since the computation is carried out to predict the self propulsion model point, all the coefficients except  $n$  cannot be compared with the measured data. The rate of revolutions of the propeller  $n$  computed by our own solver naoe-FOAM-SJTU is about 9.48, which is 5.7% larger than that of the experimental data.

Table 4 Propulsive Coefficients

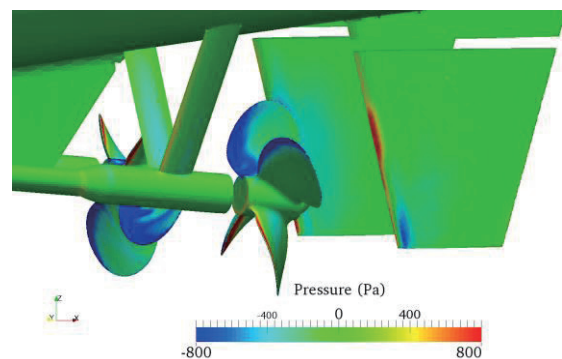
Parameters		EFD	CFD
$C_T \times 10^3$	Value	--	4.94

$C_F \times 10^3$	Value	--	3.53
$C_P \times 10^3$	Value	--	1.41
$K_T$	Value	--	0.196
$K_Q$	Value	--	0.061
n (rps) (for given SFC)	Value	8.97	9.48
	E%D	--	-5.7

Since the rate of revolutions of the propeller  $n$  is to be adjusted to obtain force equilibrium in the longitudinal direction and the simulated ship resistance is more trustable, the computed thrust is relatively lower than that of experiment at the same rate of revolution  $n$ . In our previous study (Shen 2015) on numerically predicting the propeller open water curves by overset grid method, the error becomes larger when the advance coefficient  $J$  exceeds 1.0 and in this case  $J$  is about 1.16.  $K_T$  and  $K_Q$  are obtained by the computed thrust and torque.



(a) Pressure Distribution around Bow



(b) Pressure Distribution around Propeller and Rudder

Fig. 7 Pressure Distribution

The above figure shows the pressure distribution around ship bow, propellers and rudders. In Fig. 7(b), the pressure distribution around rudders is strongly affected by the rotating propellers. While the pressure distribution in the bow side is approximately the same with that of the towing condition.

Fig. 8 shows a stern view of vortical structures displayed as isosurfaces of  $Q=200$  colored by axial velocity. In the figure, the propeller tip vortices are clearly resolved where the grid was refined, but dissipate quickly within the coarser mesh downstream. The strong hub vortex observed has a much larger size so that it is still somewhat resolved by the coarser grid downstream of the refinement. From Fig. 8 we can also see the vortices after the rudder root, which is caused by the gap between the rudder and rudder root, and this will not come out in the real test. An interesting effect occurs when the vortices of blades pass the rudders, where the vortices separated rapidly at the top side.

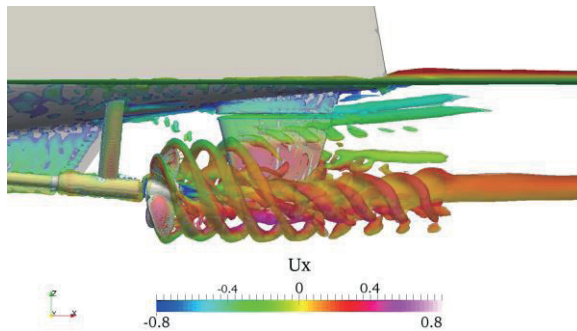


Fig. 8 Vortical Structures around Propellers and Rudders

## CONCLUSIONS

The free running computations are carried out at model point for ONR Tumblehome by our in-house solver naoe-FOAM-SJTU. During the calculation, the moving components, i.e. propellers and rudders, are simulated by the overset grid method. A proportional-integral (PI) controller is employed to adjust the rotational speed of the propeller to achieve the desired ship speed.

The rate of revolutions of the propeller  $n$  is obtained and the computed result agrees well to the measured data. The error may be due to the inaccuracy prediction of propellers at large advancing coefficient  $J$ . Simulated propulsion coefficients and ship motions are also presented. Additionally, pressure distribution and vortical structures around ship hull are also presented to illustrate the detailed flow information in self-propulsion conditions.

## ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (Grant Nos. 51379125, 51490675, 11432009, 51411130131), The National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2013CB036103), High Technology of Marine Research Project of The Ministry of Industry and Information Technology of China, Chang Jiang Scholars

Program (Grant No. T2014099) and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (Grant No. 2013022), to which the authors are most grateful.

## REFERENCES

- Shen Z. R., Wan D. C. and Carrica P. M. (2015). "Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering." *Ocean Engineering*, 108: 287-306.
- Cao, H. J. and Wan, D. C. (2014). "Development of multidirectional nonlinear numerical wave tank by naoe-FOAM-SJTU solver." *International Journal of Ocean System Engineering*, 4(1), 52-59.
- Carrica, P. M., Ismail F., Hyman, M., et al. (2013). "Turn and zigzag maneuvers of a surface combatant using a URANS approach with dynamic overset grids." *Journal of Marine Science and Technology*, 18(2), 166-181.
- Ferziger, J. H. and Peric, M. (1999). "Computational methods for fluid dynamics." Second ed. Springer, Berlin.
- Hirt, C. W., and Nichols, B. D. (1981). "Volume of fluid (VOF) method for the dynamic of free boundaries." *Journal of Computational Physics*, 39, 201-225.
- Issa, R. I. (1986). "Solution of the implicitly discretized fluid flow equations by operator-splitting." *Journal of Computational Physics*, 62(1), 40-65.
- Menter, F. R. (1994). "Two-equation eddy-viscosity turbulence models for engineering applications." *AIAA Journal*, 32(8), 1598-1605.
- Van Leer, B. (1979). "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method." *Journal of Computational Physics*, 32(1), 101-136.
- Shen, Z. R., Carrica, P. M. and Wan, D. C. (2014) "Ship motions of KCS in head waves with rotating propeller using overset grid method." *ASME 33rd International Conference on Ocean, Offshore and Arctic Engineering*.
- Shen, Z. R., Zhao, W. W., Wang, J. H. and Wan, D. C. (2014). "Manual of CFD solver for ship and ocean engineering flows: naoe-FOAM-SJTU." *Technical Report for Solver Manual*, Shanghai Jiao Tong University.
- Wan, D. C. and Shen, Z. R. (2012) "Overset-RANS computations of two surface ships moving in viscous fluids." *International journal of Computational Methods*.