Proceedings of the Twenty-sixth (2016) International Ocean and Polar Engineering Conference **www.isope.org**
Rhodes, Greece, June 26-July 1, 2016

ISBN 978-1-880653-88-3; ISSN 1098-6189

Numerical Prediction of KCS Self-Propulsion in Shallow Water

Jianhua Wang[1] , Xiaojian Liu[1,2] , Decheng Wan[1] , Gang Chen[1,2]*

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, China
2 Marine Design & Research Institute of China (MARIC), Shanghai, China
*Corresponding author

ABSTRACT

The present work is focused on the numerical prediction of ship self-propulsion in different shallow water conditions: H/T=2.0 and H/T=1.2 (H is the depth of water and T is draft of ship). The KRISO Container Ship (KCS) model is used in the present simulation. Numerical computations are carried out by our own 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. A proportional-integral (PI) controller is applied to adjust the rotational speed of the propeller to achieve the desired ship speed. The simulated results, i.e. the rate of revolution of propeller *n*, propulsion coefficients, are compared to the experimental data provided by Flanders Hydraulics Research (FHR) in SIMMAN 2014. Good agreements show that the present approach is applicable for self-propulsive prediction in shallow water.

KEY WORDS: self-propulsion; overset grid; shallow water; KCS; naoe-FOAM-SJTU solver

INTRODUCTION

Self-propulsion, which is closely bound up with energy consumption, is one of the most important characters in ship performance. However, due to the economic slump, the world shipping industry has now been experiencing difficulties. And the coming out of EEDI proposed by IMO also makes it essential to accurately estimate ship self-propulsion in its design stage. Up to the present, the main method for predicting ship self-propulsion is still model scale experiments in a conventional towing tank.

With the performance of computers boosting in the past few decades, tremendous advances have been made in the development of Computational Fluid Dynamics (CFD) on ship hydrodynamics. Thus, direct simulation of ship resistance and seakeeping becomes feasible. However, great challenges remain in the area of ship hydrodynamics, especially for fully appended model with rotating propellers and rudders, i.e. self-propulsion and free maneuvering computations. The dynamic overset grid method, including a hierarchy of bodies that enable computation of full 6DoF and moving components (rudders, propellers), makes it possible for computing complex motions, including problems like ship motion in large-amplitude waves, self-propulsion and free maneuvering with rotating propellers and moving rudders.

Nowadays, overset grid method has been applied to the computations of ship hydrodynamics, especially for the direct simulation of hull-propeller-rudder interaction. The KRISO Container Ship (KCS) self-propulsion test was one of the benchmark cases in the CFD Workshops of Tokyo 2005 (Hino, 2005), Gothenburg 2010 (Larsson et al., 2010), and even in Tokyo 2015. Lübke (2005) first preformed the simulation with discretized propeller model at fixed propeller revolution rate using the commercial code CFX in the Tokyo 2005 CFD Workshop. Later, more and more people applied the discretized method for the self-propulsive prediction on different platforms. Carrica et al. (2010), who used dynamic overset grid to handle with the rotating propeller, computed the KCS self-propulsion free to trim and sinkage. Castro et al. (2011) carried out full scale self-propulsion computations for KCS using discretized propeller. Shen et al (2015) implemented dynamic overset grid module to OpenFOAM and applied to the KCS self-propulsion simulation.

Although there are many works on KCS self-propulsion computation, the numerical calculation of self-propulsion in shallow water is rarely done. In the present work, we applied dynamic overset grid method to numerically calculate the KCS self-propulsion in two shallow water conditions (H/T=2.0 and H/T=1.2). The KCS model from Potsdam Model Basin (SVA) is used (experiment is done at FHR, model borrowed from SVA) and the ship model is free to trim and sinkage with rotating propeller and static rudder. A proportional-integral (PI) controller is applied to adjust the rotational speed of the propeller to achieve the desired ship speed.

The objective of the present work is to investigate the capability of our in-house code naoe-FOAM-SJTU (Shen and Wan, 2012) in predicting self-propulsion in shallow water. In this paper, the numerical results of the ship motions as well as the propulsion coefficients are presented and compared with physical experiments performed at FHR (data published in SIMMAN 2014). The existing solver naoe-FOAM-SJTU is developed by implementing overset grid technique into the open source platform OpenFOAM. An external software SUGGAR++

(Noack et al., 2009) is used to generate the domain connectivity information (DCI) to connect the solutions among multiple overset component grids. The naoe-FOAM-SJTU solver has been extensively validated in the field of ship hydrodynamics (Cao and Wan, 2014; Shen and Wan, 2013; Shen et al., 2014a, 2015; Wang and Wan, 2015; Wang et al., 2015a). With overset grid technique handling moving components and full viscous-flow calculating URANS equation, the numerical results are promising and the detailed information of flow field at wake region will be given.

In this paper, the numerical methods including governing equations, solver and algorithm, overset grid method will be introduced first. Then the second section will be the computational overviews, where the geometry model, computational domain, mesh configuration, PI controller for rotating components, test conditions and calculation procedure will be described in detail. Following this is the numerical results and analysis, where the numerical results of self-propulsion test including ship motions, hydrodynamic force coefficients for ship hull and moving components will be presented. Furthermore, the ship wake profile and pressure distribution around propeller and rudder will be presented for further analysis. Finally, a summary of the paper is drawn.

NUMERICAL METHODS

Governing Equations

Generally, Navier-Stokes equations are used to describe the motion of fluid continuum. In terms of the unsteady incompressible two-phase 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:

$$\nabla \cdot \mathbf{U} = 0 \tag{1}$$

$$\frac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot \left[(\rho \boldsymbol{U} - \boldsymbol{U}_g) \boldsymbol{U} \right] = -\nabla p_d - \boldsymbol{g} \cdot \boldsymbol{x} \nabla \rho + \nabla \cdot (\mu_{eff} \nabla \boldsymbol{U}) + (\nabla \boldsymbol{U}) \cdot \nabla \mu_{eff} + f_\sigma + f_s \tag{2}$$

where $\mathbf{U}$ is the fluid velocity field and $\mathbf{U}_g$ the velocity of mesh points; $p_d = p - \rho \mathbf{g} \cdot \mathbf{x}$ is the dynamic pressure, obtained by subtracting the hydrostatic component from the total pressure; ρ is the mixture density of the two-phase fluid; $\mathbf{g}$ is the gravity acceleration; $\mu_{\text{eff}} = \rho(\nu + \nu_t)$ is the effective dynamic viscosity, in which ν and ν_t are the kinematic viscosity and kinematic eddy viscosity respectively, the latter one is obtained by the SST $k-\omega$ turbulence model (Menter, 1994); f_σ is a source term due to surface tension 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 (α), 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:

$$\begin{aligned} \rho &= \alpha \rho_1 + (1-\alpha) \rho_2 \\ \mu &= \alpha \mu_1 + (1-\alpha) \mu_2 \end{aligned} \tag{3}$$

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

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \left[\left(\boldsymbol{U} - \boldsymbol{U}_g \right) \alpha \right] + \nabla \cdot \left[\boldsymbol{U}_r \left(1 - \alpha \right) \alpha \right] = 0 \tag{4}$$

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

Solver and Algorithm

The CFD solver applied in this paper is naoe-FOAM-SJTU(Shen et al., 2014b), which is based on the open source CFD platform OpenFOAM. It is developed to deal with complex motion problems such as large-amplitude ship motion in waves, moving rudder and rotating propeller in ship self-propulsion and free maneuvering. During the calculation, the URANS equations and VOF transport equation are discretized by the finite volume method (FVM), and for the discretized URANS 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 (2012) and Issa (1986). Additionally, several built-in numerical schemes in OpenFOAM are used in solving the PDEs. The convection terms are discretized 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 Euler scheme is applied for temporal discretization

Overset Grid Method

Overset grid is a grid system that 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 connective information is built and exchanged between these grids via interpolation of the fluid variables at appropriate cells or points. 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, especially for self-propulsion and free maneuvering.

The overset grid module is implemented into OpenFOAM depending on its numerical methods including the cell-centered scheme and unstructured grids. SUGGAR++ (Noack et al., 2009) 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 is the inertial system (earth-fixed system $o'x'y'z'$) and the other is non-inertial system (ship-fixed system $oxyz$). The inertial system can be fixed to earth or move at a constant speed respect to the ship (here we only apply the horizontal motion for the moving of inertial system). The non-inertial system is fixed to the ship and can translate or rotate according to the ship motions. Details of the 6DoF module with overset grid method can be found in Shen (2014b).

COMPUTATIONAL OVERVIEWS

Geometry Model

The geometry ship model with propeller and rudder in the present study is shown in Fig. 1 and the principal dimensions of the ship hull and its appendages are listed in Table 1.

Fig. 1 KRISO Container Ship (KCS) Model

Table 1 Principal dimensions of ship hull and its appendages

Dimension	Full Scale	SVA
Scale	1.0	52.667
Lpp (m)	230.0	4.3671
T (m)	10.8	0.2051
Displacement(m^3)	52030	0.3562
S of Rudder (m^2)	115	0.0415
D of Propeller (m)	7.9	0.150
Rotation	Right hand	Right hand

Computational Domain

With dynamic overset grid technique, here we have four part of the computational grids, i.e. grid around ship hull, grids for both propeller and rudder and the last one is the background grid. The four part grids have overlapping areas, which can move independently without restrictions and build connections among them by interpolation at appropriate cells or points. The computational domain arrangement is shown in Fig. 2.

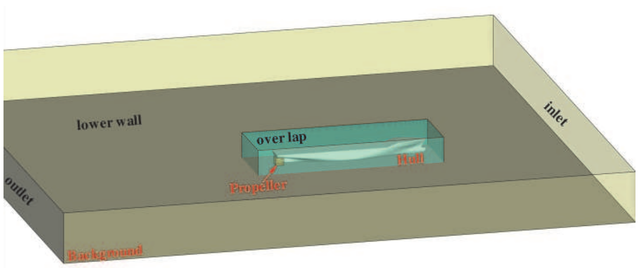


Fig. 2 Computational Domain

In the present study, we have chosen the shallow water condition for self-propulsion simulation and the boundary condition is shown in Fig. 2. The background grid move forward with the ship speed (only advancing in X direction without flow velocity).

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 4.62 M and the detail information for each part is show in Table 2. The local mesh distribution around ship hull is shown in Fig. 4 and the local mesh arrangement around propeller and rudder is shown in Fig. 4.

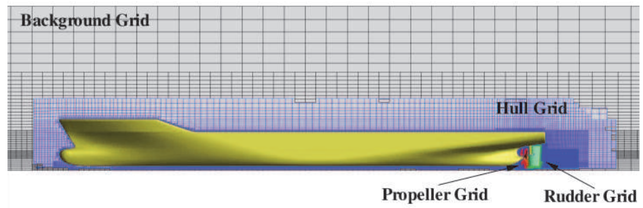
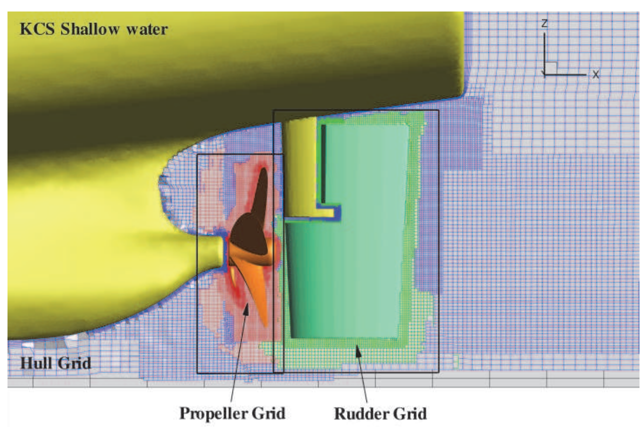


Fig. 3 Mesh distribution around ship hull

a) Propeller and rudder model (SVA)



b) Overset grid distribution

Fig. 4 Mesh arrangement around propeller and rudder

Table 2 Grid Information in Each Part

Grid	Number	Motion Level
Background	0.90M	Highest
Hull	2.49M	Parent
Propeller	0.76M	Children
Rudder	0.47M	Children

PI Controller

For the self-propulsion case, KCS is equipped with a rotating propeller, which provide the thrust for the ship to advance, and a static rudder. The computations are carried out at the model point following the experimental procedure.

A proportional-integral (PI) controller is applied to adjust the rotational rate of the propeller to achieve the target ship speed. The instantaneous RPS of the propeller is calculated as:

$$n = Pe + I\int_0^t edt \tag{5}$$

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

$$e = U_{\text{target}} - U_{\text{ship}} \tag{6}$$

The PI controller is activated at the beginning of the calculation and updates the RPS at the end of each time step. The rate of revolutions of the propeller n is to be adjusted to obtain force equilibrium in the longitudinal direction:

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

where T is the computed thrust, $R_{T(SP)}$ is the total resistance of the self-propulsion KCS model in calm water.

Test Conditions

The present work is for self-propulsion prediction of KCS model in shallow water. According to the experimental procedure, the ship model is free to trim and sinkage at model point (no friction deduction force applied) in calm water with rotating propeller and static rudder. The KCS ship model is advancing in two different water depth, i.e. H/T=1.2 and H/T=2.0. The approaching speed of KCS model is $V_A = 0.62$ m/s ($F_r = 0.095$). In the experiment setup, the rotational speed of the propeller is set to 330 RPM (speed chosen to match free model tests at BSHC).

The boundary conditions are identical with zero velocity and zero gradient of pressure imposed on inlet and far-field boundaries. No-slip condition is applied for the bottom moving wall. The boundary condition of interface between each two grids is set to overlap for later information interpolation. The propeller grids are fitted behind the ship hull and rotating in right hand. Another thing we should care about is that there is a physical gap between the propeller and the hull, as well as the rudder and hull to provide enough grids for the interpolation between different parts of regions. When dealing with self-propulsion calculations, 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 0.62 m/s and the rate of propeller was zero. 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 could also cause overshoots in propeller RPS and ship velocity.

Calculation Procedure

The coupled calculation procedure with naoe-FOAM-SJTU solver is shown in Fig. 5.

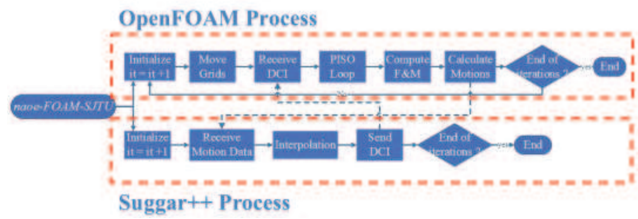


Fig. 5 naoe-FOAM-SJTU calculation procedure

Shen, et al (2015) applied this solver in calculating the KCS self-propulsion in deep water and grid convergence study is also presented for both the towed condition and propeller open water calculation. Wang et al (2015b) also investigated the fully appended ONR Tumblehome ship with free running condition (benchmark case in Tokyo 2015 CFD Workshop). Good agreements show that the existing solver is applicable for the self-propulsion simulation. In this paper, we focus more on the self-propulsion performance in shallow water condition.

NUMERICAL RESULTS AND ANAYLSIS

Numerical Results

All CFD predicted coefficients and motions 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:

$$C_T = \frac{R_T}{\frac{1}{2}\rho U^2 S_0} \tag{8}$$

$$K_T = \frac{T}{\rho n^2 D_P{}^4} \tag{9}$$

$$K_Q = \frac{T}{\rho n^2 D_P{}^5} \tag{10}$$

Table 3 Numerical results for H/T=1.2

Parameter	CFD	Experiment	Error
C_T	7.39e-3	7.17e-3	3.12%
K_T	0.319	0.363	-12.22%
K_Q	5.49e-2	5.84e-2	-5.92%
RPM(r/min)	329.92	331	-0.33%
Sinkage(mm)	6.68	6.26	6.71%
Trim(mm/m)*	0.351	0.40	-12.25%

Table 4 Numerical results for H/T=2.0

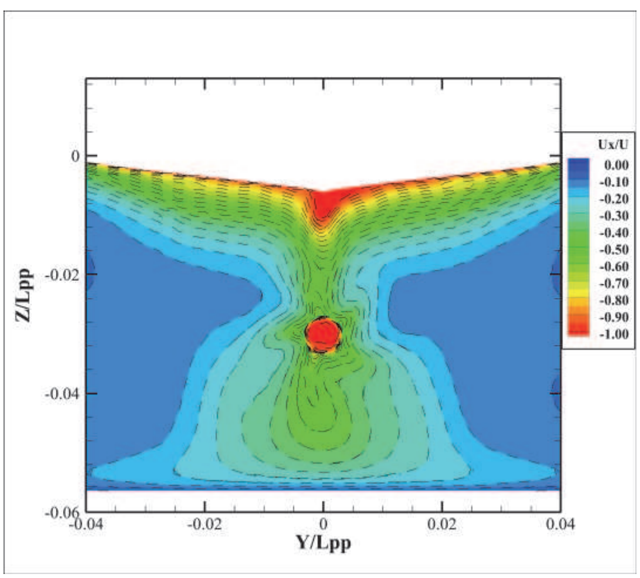
Parameter	CFD	Experiment	Error
C_T	6.07e-3	6.76e-3	-10.21%
K_T	0.271	0.307	-11.51%
K_Q	4.11e-2	5.15e-2	-20.19
RPM(r/min)	324.12	331	-2.08%
Sinkage(mm)	3.03	2.96	2.36%
Trim(mm/m)*	0.202	0.26	-22.29%

*Trim is calculated using the position of sinkage fore. and sinkage aft. divided by ship length.

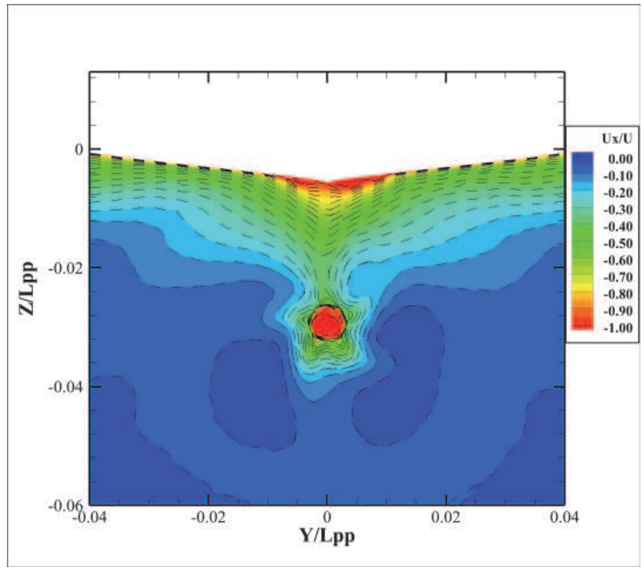
Numerical predicted results for shallow water H/T=1.2 is listed in Table 3 and results for H/T=2.0 is shown in Table 4. We can see that the predicted rotational speed of the propeller agrees very well to the experimental result, with the error up to 2%. As for the force coefficients, the simulation results also show good agreement with the experimental data. We will give analysis for the numerical data here with the detailed flow field information in the next section. Furthermore the computed ship motions, especially for the sinkage, are consistent with the experimental ones.

Wake Distribution

All the figures in this section is in non-dimensional scale by ship length L_{pp} and velocity magnitude U, including the position (x, y, z) and the flow field (u, v, w). From the Fig. 6 we can see that the flow velocity before the propeller disk in shallow water H/T=1.2 is larger than that of the H/T=2.0. This can perfectly explain that the equivalent advance coefficient J in H/T=1.2 is larger and as a result the propulsion thrust is bigger at the approximately rotational rate of propeller. This can explain the difference between the computed thrust shown in Table 3 and Table 4. Also, the figure can show that the velocity contour at the bottom has a big difference, and that is why the flow velocity differs a lot before the propeller disk.



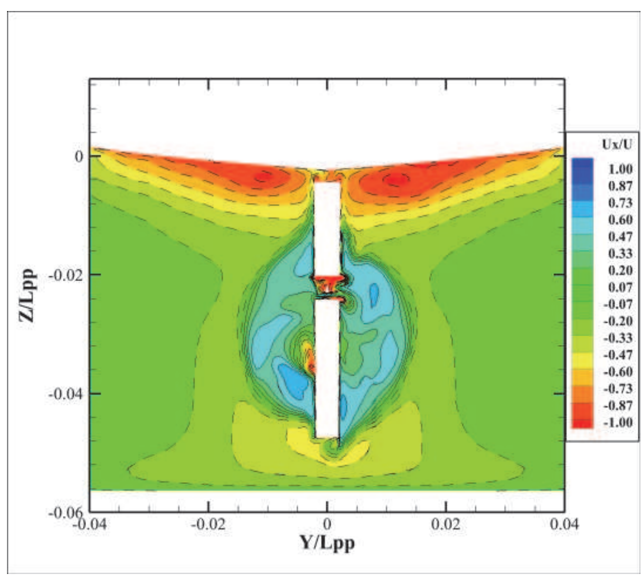
a) H/T=1.2



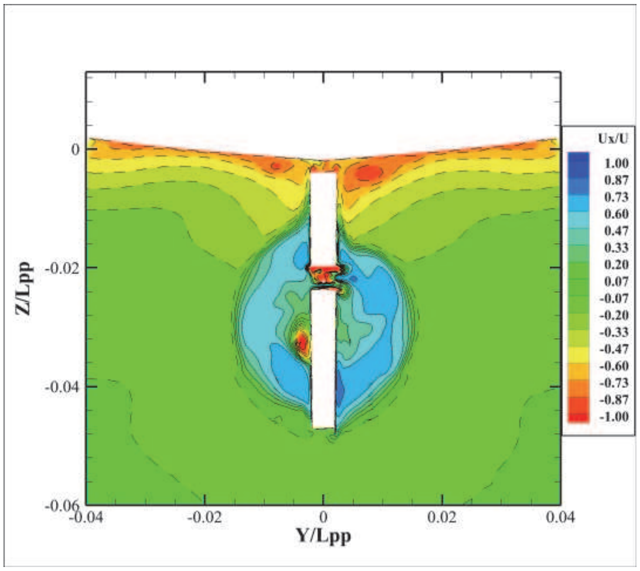
b) H/T=2.0

Fig. 6 Wake distribution before the propeller disk

Fig. 7 shows the wake distribution at slice X/L_{pp}=1.0, which is around the rudder. In both shallow water condition H/T=1.2 and H/T=2.0, the flow velocity show big difference with that in Fig. 6. Negative velocity appears at both conditions around the rudder, which is due to the propeller wake region. The water depth can affect the wake a lot according to this figure. The velocity field in shallow water condition H/T=1.2 around ship hull extends to a wider region at same value than that of H/T=2.0. The wake distribution is more complex in the H/T=1.2 condition, which illustrate that the shallow water effect is of significant importance for ship advancing with rotating propeller and moving rudder.



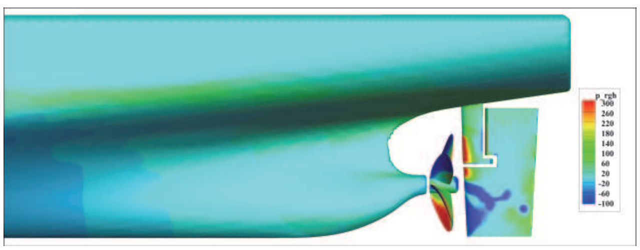
a) H/T=1.2



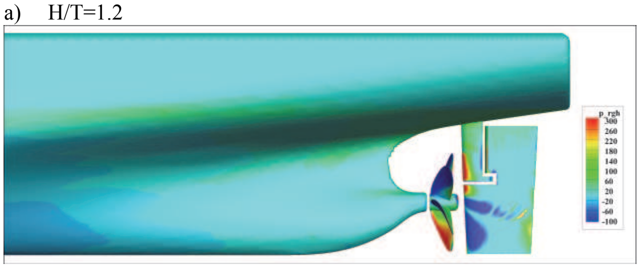
b) H/T=2.0

Fig. 7 Wake distribution at X/L_{pp}=1.0

Pressure Distribution



a) H/T=1.2



b) H/T=2.0

Fig. 8 Pressure distribution around propeller and rudder

The above figure shows the pressure distribution around ship hull, propeller and rudder. In Fig. 8, the pressure distribution around rudder is strongly affected by the rotating propeller. From wake profile section we can see that the equivalent flow velocity is larger at H/T=1.2, and as a result the pressure in H/T=1.2 is smaller. Furthermore, due to the high shallow water effect in case H/T=1.2, the difference of pressure distribution in the ship hull is very obvious.

Vortical Structure

Fig. 9 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 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. 9 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. Another effect occurs when the vortices of blades pass the rudder, where the vortices separated rapidly at both the top side and bottom side. The vortices can give a detailed view of the propeller, rudder and hull interaction.

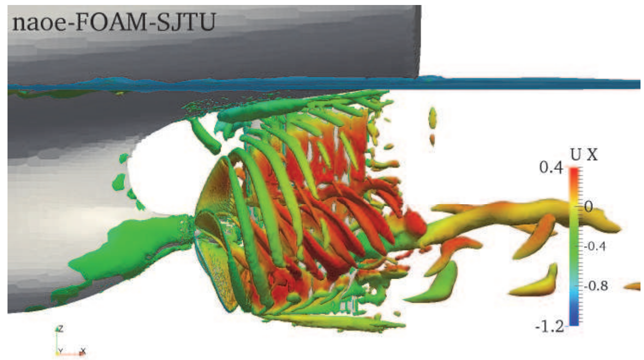


Fig. 9 Q contour around propeller and rudder

CONCLUSIONS

The self-propulsion computations are carried out at model point for KCS in shallow water conditions (H/T=1.2 and H/T=2.0). All the simulations are calculated by our own CFD solver naoe-FOAM-SJTU. During the process, the moving components, i.e. propeller and rudder, are handled by the dynamic overset grid method. A proportional-integral (PI) controller is employed to adjust the rotational rate of the propeller to achieve the desired ship speed.

The rate of revolutions of the propeller RPM is obtained and the computed results agrees well to the measured data with error up to 2%. Ship motions and propulsion coefficients are given in both two shallow conditions. The thrust in H/T=1.2 is larger than that of H/T=2.0 and is explained by the detailed wake distribution. Both wake distribution before the propeller disk and around rudder is presented and shows that shallow water can affect the equivalent advancing coefficients J. Additionally, pressure distribution and vortical structures around propeller and rudder are also presented to illustrate the detailed flow information for self-propulsion in shallow water conditions. Finally we can draw the conclusion that the shallow water condition shows significant effect for the ship advancing with rotating propeller. And the present approach with dynamic overset grid method directly simulating propeller is applicable for the self-propulsion prediction in shallow water.

Future work will focus on the free maneuvering simulation including course keeping, turning circle and zigzag with rotating propellers and moving rudders. More emphasis will be placed on the large-amplitude motion to investigate the ship-propeller-rudder interaction in complex conditions.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of

China (51379125, 51490675, 11432009, 51579145, 11272120), Chang Jiang Scholars Program (T2014099), Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (2013022), Innovative Special Project of Numerical Tank of Ministry of Industry and Information Technology of China (2016-23) and Foundation of State Ley Laboratory of Ocean Engineering (GKZD010065), to which the authors are most grateful.

REFERENCES

Cao, H, and Wan, DC (2014). "Development of Multidirectional Nonlinear Numerical Wave Tank by naoe-FOAM-SJTU Solver," *Int J Ocean Sys. Eng*, 4(1), 52–59.

Carrica, PM, Castro, AM, and Stern, F (2010). "Self-propulsion computations using a speed controller and a discretized propeller with dynamic overset grids," *J Mar Sci Technol*, 15(4), 316–330.

Castro, AM, Carrica, PM, and Stern, F (2011). "Full scale self-propulsion computations using discretized propeller for the KRISO container ship KCS," *Comput Fluids*, 51(1), 35–47.

Ferziger, JH, and Peric, M (2012). *Computational methods for fluid dynamics* Springer Science & Business Media.

Hino, T (2005). *The Proceedings of CFD Workshop, Tokyo, 2005* National Maritime Research Institute.

Issa, RI (1986). "Solution of the implicitly discretised fluid flow equations by operator-splitting," *J Comput Phys*, 62(1), 40–65.

Larsson, L, Stern, F, and Visonneau, M (2010). "Gothenburg 2010, A Workshop on Numerical Ship Hydrodynamics."

van Leer, B (1979). "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *J Comput Phys*, 32(1), 101–136.

Lübke, LO (2005). "Numerical simulation of the flow around the propelled KCS," *Proc CFD Workshop 2005*, Tokyo, Japan, 9–11.

Noack, RW, Boger, DA, Kunz, RF, and Carrica, PM (2009). "Suggar++: An improved general overset grid assembly capability," *Proc 47th AIAA Aerosp Sci Exhib*, 22–25.

Shen, Z, and Wan, DC (2013). "RANS computations of added resistance and motions of a ship in head waves," *Int J Offshore Polar Eng*, ISOPE, 23(04), 264–271.

Shen, Z, Ye, H, and Wan, DC (2014a). "URANS simulations of ship motion responses in long-crest irregular waves," *J Hydrodyn Ser B*, 26(3), 436–446.

Shen, Z and Wan, DC (2012). "Manual of CFD solver for ship and ocean engineering flows: naoe-FOAM-SJTU." Technical Report No. 2012SR118110, Shanghai Jiao Tong University.y.

Shen, Z, Wan, DC and Carrica, PM (2015). "Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering," *Ocean Eng*, 108, 287–306.

Wang, J, and Wan, DC (2015). "Dynamic coupling analysis of the mooring system and floating pier in the South China Sea," *Chin J Hydrodyn*, 30(2), 180–186.

Wang, J, Liu, X, and Wan, DC (2015a). "Numerical Simulation of an Oblique Towed Ship by naoe-FOAM-SJTU Solver," *Proceedings of the Twenty-fifth (2015) International Ocean and Polar Engineering Conference (ISOPE)*, Kona, Big Island, Hawaii, USA, pp.432-438.

Wang, J, Liu, X, and Wan, DC (2015b). "Numerical prediction of free runing at model point for ONR Tumblehome using overset grid method," *Proc CFD Workshop 2015*, Tokyo, Japan, pp. 383–388.