Ocean Engineering 108 (2015) 287–306

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Dynamic overset grids in OpenFOAM with application to KCS self-propulsion and maneuvering

Zhirong Shen [a], Decheng Wan [a,*], Pablo M. Carrica [b]

[a] State Key Laboratory of Ocean Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Collaborative Innovation Center of Advanced Ship and Deep-Sea Exploration, 800 Dongchuan Road, Shanghai 200240, China
[b] IIHR-Hydroscience and Engineering, The University of Iowa, 300 South Riverside Dr., Iowa City, IA 52242, USA

ARTICLE INFO

Article history:
Received 2 June 2014
Accepted 21 July 2015
Available online 29 August 2015

Keywords:
OpenFOAM
naoe-FOAM-SJTU solver
Overset grid
Self-propulsion
Zig-zag maneuver
KCS

ABSTRACT

An implementation of the dynamic overset grid technique into naoe-FOAM-SJTU solver developed by using the open source code OpenFOAM is presented. OpenFOAM is attractive for ship hydrodynamics applications because of its high quality free surface solver and other capabilities, but it lacks the ability to perform large-amplitude motions needed for maneuvering and seakeeping problems. The implementation relies on the code Suggar to compute the domain connectivity information (DCI) dynamically at run time. Several Suggar groups can be used in multiple lagged execution mode, allowing simultaneous evaluation of several DCI sets to reduce execution time and optimize the exchange of data between OpenFOAM and Suggar processors. A towed condition of the KRISO Container Ship (KCS) are used for static overset tests, while open-water curves of the KP505 propeller and self-propulsion and zig-zag maneuvers of the KCS model are exercised to validate the dynamic implementation. For self-propulsion the ship model is fitted with the KP505 propeller, achieving self-propulsion at $Fr=0.26$. All self-propulsion factors are obtained using CFD results only, including those from open-water curves, towed and self-propulsion conditions. Computational results compare well with experimental data of resistance, free-surface elevation, wake flow and self-propulsion factors. Free maneuvering simulations of the HSVA KCS model appended with the HSVA propeller and a semi-balanced horn rudder are performed at constant self-propulsion propeller rotational speed. Results for a standard 10/10 zig-zag maneuver and a modified 15/1 zig-zag maneuver show good agreement with experimental data, even though relatively coarse grids are used. Grid convergence studies are performed for the open-water propeller test and bare hull KCS model to further validate the implementation of the overset grid approach.

1. Introduction

As computers and numerical methods advance, direct computations of ship maneuvers and seakeeping are becoming feasible. The open source computational fluid dynamics (CFD) code OpenFOAM is attractive as a tool for computational ship hydrodynamics since it has good flow and free surface capabilities and is free. OpenFOAM is developed and released by OpenCFD Ltd at ESI Group and is one of the most popular open source CFD packages. However, OpenFOAM lacks the ability to perform arbitrary motions needed to move ships, propellers, rudders and other appendages simultaneously as required for free-model ship simulations.

Although CFD has evolved significantly in the past decade, great challenges remain in the area of ship hydrodynamics. Computations of seakeeping, self-propulsion, dynamic stability and maneuvering problems are especially difficult, mostly due to limitations of traditional meshing methodologies to handle moving geometries. Using dynamically deforming meshes ship motions are usually restricted to small amplitudes to prevent divergence due to excessive grid deformation. Shen and Wan (2013) predicted pitch and heave motions of the Wigley Hull and the model DTMB5512 in head waves using the deforming mesh technique with OpenFOAM. Although good agreement with experiments was obtained, motions were still limited to small amplitudes. The sliding mesh approach can handle large-amplitude motions, but simultaneous motion of close moving objects like rudders and propellers is extremely challenging, leading to frequent failure or poor performance.

The dynamic overset grid technology, including a hierarchy of objects that enable computation of 6DoF and control surfaces (rudders, stabilizers), opens the possibility of computation of complex motions, including problems large-amplitude waves,

* Corresponding author. Tel.: +86 21 3420 5697; fax: +86 21 3420 5685.
E-mail address: dcwan@sjtu.edu.cn (D. Wan).

http://dx.doi.org/10.1016/j.oceaneng.2015.07.035
0029-8018/© 2015 Elsevier Ltd. All rights reserved.

moving rudders and rotating propellers, encompassing all traditional areas of naval architecture hydrodynamics.

The overset grid technique has been successfully applied to the field of computational ship hydrodynamics. Chen and Yu (2009) investigated the effects of green water and wet deck slamming, using large eddy simulation (LES) with the level-set method. The results demonstrated the capability of the overset grid method to deal with violent flows and large amplitude motions. Broglia et al. (2013) computed turning circle maneuvers of a tanker-like ship model, evaluating the performance of several propeller models in strong oblique flows. Carrica et al. (2007) predicted the heave and pitch motions of the DTMB 5512 model in head waves using CFDShip-Iowa V4 using overset grids. The computations were carried out at two different Froude numbers and two different wavelengths. The methodology was later expanded and used for other applications, including self-propulsion (Carrica et al., 2011, 2010a; Castro et al., 2011), maneuvering (Carrica et al., 2013; Mofidi and Carrica, 2014) and dynamic stability (Carrica et al., 2012). Most of the literature uses structured grids, with the consequent grid generation expense when handling complex geometries such as appendages, propellers and rudders.

Boger et al. (2010) first developed the FoamedOver library to implement the overset capability in OpenFOAM. This library was developed based on Suggar++ (Noack et al., 2009) and DiRTlib libraries (Noack, 2005a). Suggar++, an improved version of Suggar (Noack, 2005b), was used to generate the domain connectivity information (DCI) to connect the solutions among multiple overset component grids. DiRTlib is a solver-neutral library that simplifies the addition of an overset capability to a flow solver. The applications shown, however, were limited to simple geometries with structured grids.

In this paper, the dynamic overset technique is implemented into the OpenFOAM-based solver naoe-FOAM-SJTU (Shen et al., 2012, Shen and Wan, 2013). It is derived from interDyMFoam (a standard solver in OpenFOAM) and adds a 6DoF motion solver based on Euler angles and a wave generation and damping module for various types of waves common in marine and ocean engineering. Suggar is used to obtain the DCI. Compared with the newer Suggar++, Suggar lacks the capability for running in parallel using domain decomposition, though it has some level of parallelism using threaded execution, and the interface for OpenFOAM grids. The version of OpenFOAM used in this paper is OpenFOAM-2.0.1.

In dynamic motion situations the relative positions between overset grids change every time step, which requires Suggar to compute the DCI dynamically at run time. Suggar re-computes the DCI based on the new grid positions and sends the new DCI back to OpenFOAM. The procedures for data exchange between OpenFOAM and Suggar must be optimized to minimize this cost. The DCI is decomposed to match the CFD domain decomposition before sent to OpenFOAM processors. Each domain is partitioned in particular for one OpenFOAM processor and the size of each domain is chosen so that the communication time is minimized. In addition, a lagged mode (Carrica et al., 2010b) is used to allow OpenFOAM and Suggar to run in parallel so that the waiting time for Suggar to run is negligible.

To validate the implementation and demonstrate the potential of the overset grid approach, simulations are carried out for open-water tests of the KP505 propeller, self-propulsion of the MOERI KCS model and zig-zag maneuvers of the HSVA KCS model (with a different scale factor and propeller than the MOERI model). For the open-water tests, solutions with overset and non-overset grids are compared. A single-run approach (Xing et al., 2008) is used to obtain the whole open-water curve in a single computation.

The KCS self-propulsion test was one of the benchmark cases in the CFD Workshops of Tokyo 2005 (Hino, 2005) and Gothenburg 2010 (Larsson et al., 2014). Lübke (2005) first preformed the simulation with discretized propeller at fixed propeller revolution rate with the commercial code CFX in the CFD Workshop of Tokyo in 2005. Five years later, other researchers (Bugalski and Hoffmann, 2010; Jin et al., 2010; Lee and Rhee, 2010, Wu et al., 2010, Zhang et al., 2010) carried out the same case for the Gothenburg 2010 Workshop. However, these computations used the sliding grid approach to model the rotating propeller. Carrica et al. (2010a) used an overset grid approach to perform self-propulsion computations of KCS, allowing the model to sink and trim in calm water. Self-propulsion with discretized propellers has also been studied for KCS in full scale (Castro et al., 2011), and for a variety of geometries in model scale (Carrica et al., 2010a, 2011, 2012).

Fully predictive simulation of maneuvers requires the additional capability of handling discretized moving rudders and propellers, which can be only done for general geometries with overset grids. Literature of free model ship maneuvers with discretized moving rudders and propellers is very limited. Possibly the first such computation involved turn and zig-zag maneuvers of the KVLCC1 tanker (Carrica and Stern, 2008). In the present work the KCS model from HSVA (Steinwand, 2006) is used, and standard 10/10 and modified 15/1 zig-zag maneuvers are performed. Simulations of zig-zag maneuvers for this geometry with discretized propeller and rudder have been performed by Mofidi and Carrica (2014).

To gain more confidence on the methodology and implementation, two grid convergence studies are performed using four sets of overset grids with $\sqrt{2}$ refinement ratios. The first one is the open-water tests of KP505 propeller at an advance coefficient $J=0.7$, evaluating convergence of thrust and torque coefficients. The second involves the bare hull KCS in calm water at $Fr=0.26$, where the total resistance coefficient is analyzed.

2. Implementation of dynamic overset grids

The overset grid technique allows separate overlapping grids to move independently without restrictions and builds connection among them by interpolation at appropriate cells or points. The process is described in detail in Noack (2005b) and Carrica et al. (2010b) and only an overview is provided herein.

Cells located outside the domain or of no interest, such as inside a body, are marked as holes and excluded from the computation. Cells around hole cells are called fringe or receptor cells, and are treated as boundaries in each overset grid. Every fringe cell has a stencil consisting of several donor cells that provide information to the fringe cell from the donor grid. The value of a variable ϕ of the fringe cell is obtained by interpolation from the donor cells

$$\phi = \sum_{i=1}^{n} \omega_i \cdot \phi_i \tag{1}$$

where ω_i is the weight coefficient and ϕ_i is the donor cell value for each donor i. If a fringe cell cannot find a valid donor stencil, it is marked as an orphan. An orphan cannot receive data from any donor grid. Orphans usually occur in regions where there is insufficient overlap between the overset grids.

The information mentioned above is contained in the DCI produced by Suggar. Suggar is capable of providing DCI for node-centered solvers, and cell-centered solvers, and can handle both structured and unstructured grids. Since OpenFOAM stores values at cell centers and is discretized with unstructured grids, these two features of Suggar makes it a good candidate for the implementation of overset grids. Notice that only values on cell centers are interpolated with Suggar. Suggar cannot handle the interpolation of OpenFOAM's boundary values.

For static situations, the domain connectivity information (DCI) is computed only once as a pre-processing step. For dynamic motion situations, the relative positions between overset grids change every time step, requiring re-computation of the DCI repeatedly. This forces exchange of information between OpenFOAM and Suggar at every time step, a costly process that has to be optimized to achieve reasonable performance. In the implementation herein, Suggar runs in separate processors and the communication between OpenFOAM and Suggar is handled with MPI. In the simplest implementation, OpenFOAM and Suggar would be running serially. For each time step, OpenFOAM computes the flow solution first, then integrates the forces and moments and predicts the motions for the next time step. Once Suggar receives the predicted motions it starts computing the DCI, and OpenFOAM needs to wait until Suggar completes the computation of the DCI to start computation of the next time step or nonlinear iteration. In such a serial mode OpenFOAM and Suggar wait for each other, resulting in considerable idle time. The problem is magnified because Suggar takes longer to compute the DCI on unstructured grids than on structured grids.

A lagged mode as described in Carrica et al. (2010b) is implemented to allow OpenFOAM and Suggar to perform computations in parallel, and updated into the naoe-FOAM-SJTU solver. The detailed procedure is shown in Fig. 1. OpenFOAM and Suggar start computating simultaneously at the beginning of each time step. OpenFOAM sends motion data to Suggar and Suggar transfers the DCI back to OpenFOAM at the end of each time step. At the beginning of time step n, Suggar starts computation of the DCI for time step $n+1$ using motions data predicted by OpenFOAM by extrapolating from the available motion data in time steps $n-1$ and n. Although the extrapolation of motions will cause a discrepancy in grid positions between OpenFOAM and Suggar, in a problem with a final steady state the motions will converge to the same position and the lag will be inconsequential.

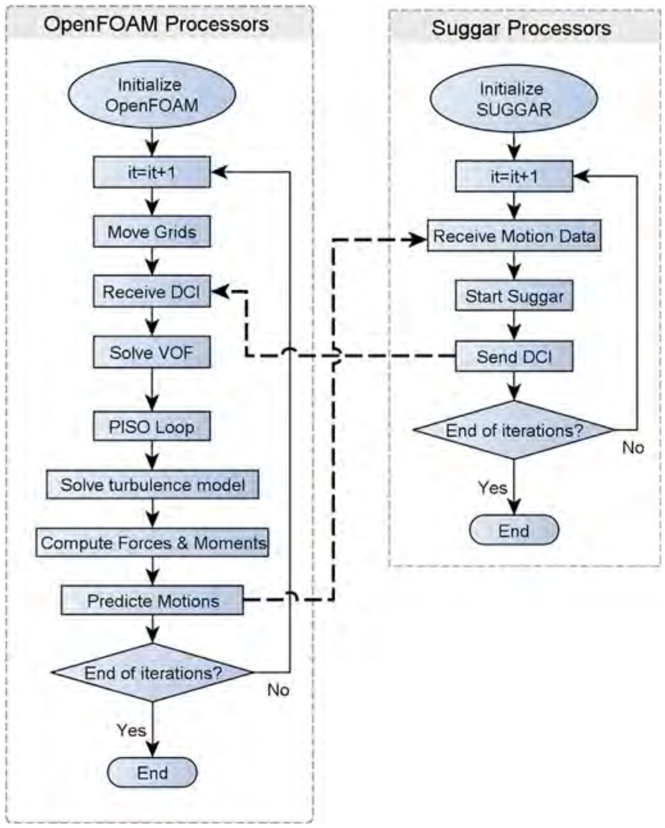


Fig. 1. Flow chart depicting exchanges between OpenFOAM and Suggar in lagged mode.

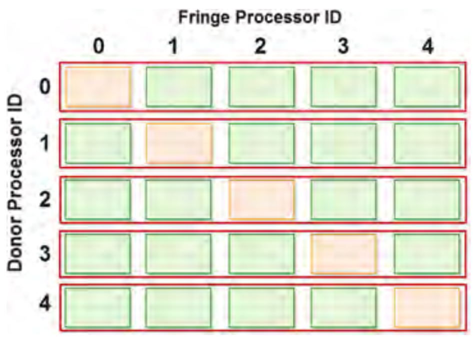


Fig. 2. Decomposition of DCI data.

Similarly, motions that are linear (with constant velocity) are exactly predicted by a linear extrapolation and thus have no error in position. Unsteady solutions in which time step or accelerations are small result negligible position errors. This is the case for most ship hydrodynamics problems, and is particularly true in OpenFOAM where time steps are typically very small. In addition, if Suggar takes longer than OpenFOAM to complete a time step, more than one Suggar processor can be run in multiple-lagged mode (Carrica et al., 2010b).

For parallel computations, the DCI from Suggar is decomposed to match the CFD domain decomposition before being sent to OpenFOAM processors. For example, if OpenFOAM is using 5 processors, then the DCI is split into 5^2 blocks. As shown in Fig. 2, Suggar sends each row of blocks, wrapped in red boxes, to each OpenFOAM processor. Each row has 5 sub-blocks sent respectively to the 5 fringe processors from the corresponding donor processors. Notice that the blocks located on the diagonal, colored with light orange, are not transferred between processors because donors and fringes are in the same processor.

Fig. 3 shows parallel timelines for an example performance analysis for case with 40 processors (37 for OpenFOAM and 3 for Suggar). Processor IDs from 0 to 36 are OpenFOAM processors and the rest are Suggar processors. As shown in Fig. 3, OpenFOAM processors spend most of the time in computation (blue region) with a small amount of time spent in communication or waiting (red region). Notice that the time one Suggar processor spends in computing the DCI for one time step is about two times larger than the time OpenFOAM processors take to finish the computation of the same time step. Selecting 3 Suggar processes guarantees that the computation of the DCI will take less time than CFD, reducing the cost of communication to the lowest level possible even though Suggar processors have considerable idle time. More than 3 Suggar processes would only add to idle time to all Suggar processors, while less would result in idle time on CFD processors resulting in an increased overall wall clock time.

3. Computational methods

3.1. Governing equations

The fluid motion is represented by the incompressible unsteady Reynolds-Averaged Navier–Stokes (URANS) equations. The Volume of Fluid (VOF) method with artificial compression is used to capture the air/water interface. Details of the solution procedure as implemented in OpenFOAM are described in Jasak (1996) and Rusche (2002), and only a brief introduction is presented here.

The incompressible URANS equations for two-phase flow are written as

$$\nabla \cdot \mathbf{U} = 0 \tag{2}$$

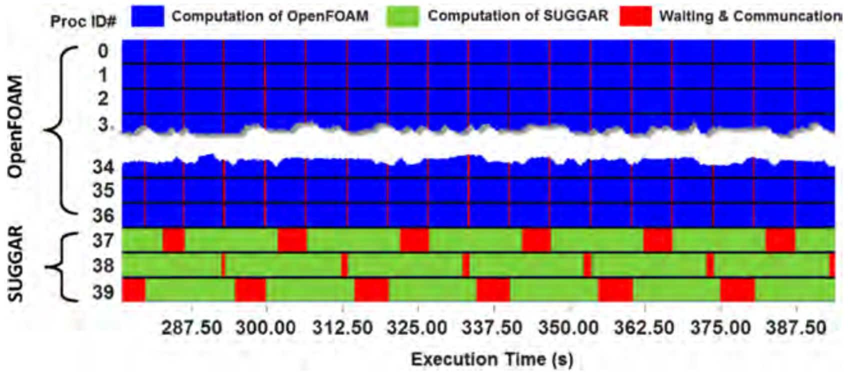


Fig. 3. Timeline for parallel execution of OpenFOAM and Suggar (37 OpenFOAM processors and 3 Suggar processors).

$$\frac{\partial \rho \mathbf{U}}{\partial t}+\nabla\cdot\left(\rho(\mathbf{U}-\mathbf{U}_g)\mathbf{U}\right)=-\nabla p_d-\mathbf{g}\cdot\mathbf{x}\nabla\rho+\nabla\cdot\left(\mu_{eff}\nabla\mathbf{U}\right)+(\nabla\mathbf{U})\cdot\nabla\mu_{eff}+f_\sigma \tag{3}$$

where $\mathbf{U}$ is fluid velocity field and $\mathbf{U}_g$ is the grid velocity; $p_d=p-\rho\mathbf{g}\cdot\mathbf{x}$ is the dynamic pressure, obtained subtracting the hydrostatic component from the total pressure; ρ is the mixture density; $\mathbf{g}=(0,0,-9.81)$ is the gravity acceleration; $\mu_{\text{eff}}=\rho(\nu+\nu_t)$ is the effective dynamic viscosity, in which ν_t and ν_t are the kinematic and eddy viscosity, respectively, and ν_t is obtained from the turbulence model. f_σ is a source term due to surface tension.

A VOF method with bounded compression technique is applied to capture free surface interface. The transport equation is expressed as

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

where α is volume of fraction, indicating the relative proportion of fluid in each cell and its value is always between zero and one:

$$\begin{cases}\alpha=0 & \text{air}\\ \alpha=1 & \text{water}\\ 0<\alpha<1 & \text{interface}\end{cases} \tag{5}$$

$\mathbf{U}_r$ in Eq. (4) is the velocity field used to compress the interface and it takes effect only on the surface interface due to the term $(1-\alpha)\alpha$. The expression of this term can be found in Berberović et al. (2009). The surface tension term in Eq. (3) is defined as $f_\sigma=\sigma\kappa\nabla\alpha$, where σ is the surface tension coefficient (0.07 kg/s^2 in water). κ is the curvature of surface interface, determined from the volume of fraction by $\kappa=-\nabla\cdot(\nabla\alpha/|\nabla\alpha|)$. The mixture density ρ and dynamic viscosity μ from the governing equations are defined as

$$\begin{cases}\rho=\alpha\rho_l+(1-\alpha)\rho_g\\ \mu=\alpha\mu_l+(1-\alpha)\mu_g\end{cases} \tag{6}$$

In addition to the momentum equations and VOF equation, the two-equation shear stress transport (SST) model (Menter 2009) is employed for turbulent closure.

3.2. 6DOF module and coordinate systems

A 6DOF motion module for ship hydrodynamics applications was implemented in OpenFOAM by Shen and Wan (2013). In this work, the motion module is redesigned for the implementation of dynamic overset grid technique with a hierarchy of objects (Carrica et al., 2010a). Two coordinate systems are used to solve the 6DOF equations. One system is inertial (Earth system $o'x'y'z'$) and the other is non-inertial (ship system $oxyz$), as shown in Fig. 4. The inertial system can be fixed to Earth or move at a constant speed respect to the Earth. The non-inertial system is fixed to the ship and translates and rotates according to the ship motions. Details of the 6DOF module can be found in Carrica et al. (2010a).

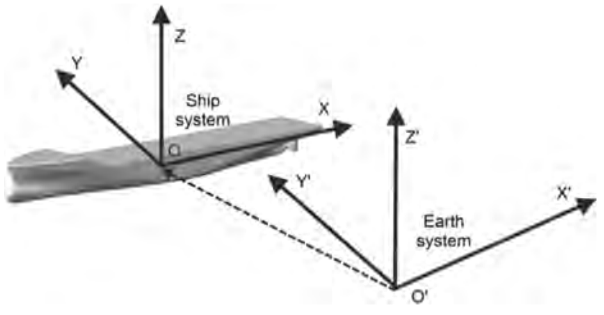


Fig. 4. Description of ship and earth systems.

As shown in Fig. 4, the two coordinate systems at the beginning of the computation are aligned, with the longitudinal x-axis pointing fore to aft, the transversal y-axis pointing from port to starboard, and the vertical z-axis pointing upward. The Earth system is fixed throughout the computation. At the beginning of the computation the origin of the Earth coordinate system is located at the intersection of the waterline and the ship's bow, while the origin of the Ship system is always fixed on the rotation center of the ship, located in a free ship at the center of gravity. A positive deflection angle of the rudder occurs when the rudder executes to starboard.

All the computations in this paper are based on the Earth system fixed with zero velocity. All objects (ship, propeller, rudder) move with their corresponding speeds and in the far field the liquid velocity is zero.

In the 6DOF module, $\boldsymbol{\eta}=(\boldsymbol{\eta}_1,\boldsymbol{\eta}_2)=(x,y,z,\phi,\theta,\psi)$ are the translation displacements and rotation angles of the object in the Earth system, representing motions of surge, sway, heave, roll, pitch and yaw, respectively. $\mathbf{v}=(\mathbf{v}_1,\mathbf{v}_2)=(u,v,w,p,q,r)$ are the corresponding linear and angular velocities in the Ship system. The velocities in the Ship system can be transformed to the Earth system and vice versa with

$$\mathbf{v}_1=\mathbf{J}_1^{-1}\cdot\dot{\boldsymbol{\eta}}_1,\quad \mathbf{v}_2=\mathbf{J}_2^{-1}\cdot\dot{\boldsymbol{\eta}}_2 \tag{7}$$

$$\dot{\mathbf{x}}_1=\mathbf{J}_1\cdot\mathbf{v}_1,\quad \dot{\mathbf{x}}_2=\mathbf{J}_2\cdot\mathbf{v}_2 \tag{8}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are 3×3 transformation matrices based on Euler angles (Fossen, 1994):

$$\mathbf{J}_1=\begin{bmatrix}\cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi\\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi\\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta\end{bmatrix} \tag{9}$$

$$\mathbf{J}_2 = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{10}$$

Since all CFD computations occur on the Earth system, the total forces and moments are computed in the Earth system first and then projected into the Ship system

$$\begin{cases} \mathbf{F}_s = (X_s, Y_s, Z_s) = \mathbf{J}_1^{-1}\cdot\mathbf{F}_e \\ \mathbf{M}_s = (K_s, M_s, N_s) = \mathbf{J}_1^{-1}\cdot\mathbf{M}_e \end{cases} \tag{11}$$

where the subscript e and s represent the forces and moments in the Earth and Ship systems, respectively. $\mathbf{F}_e$ and $\mathbf{M}_e$ are the total forces and moments acting on the solid surfaces of each components including hull, rudder and propeller, with the moments computed with respect to the center of rotation in the Earth system. X, Y, Z, K, M and N, obtained from Eq. (11), are surge, sway and heave forces and roll, pitch and yaw moments, respectively.

The linear and angular accelerations in the Ship system are computed from

$$\begin{cases} \dot{u} = X_s/m + vr - wq + x_g(q^2+r^2) - y_g(pq-\dot{r}) - z_g(pr+\dot{q}) \\ \dot{v} = Y_s/m + wp - ur + y_g(r^2+p^2) - z_g(qr-\dot{p}) - x_g(qp+\dot{r}) \\ \dot{w} = Z_s/m + uq - vp + z_g(p^2+q^2) - x_g(rp-\dot{q}) - y_g(rp+\dot{p}) \\ \dot{p} = \frac{1}{I_{xx}}\left\{K_s - (I_z-I_y)qr - m[y_g(\dot{w}-uq+vp) - z_g(\dot{v}-wp+ur)]\right\} \\ \dot{q} = \frac{1}{I_{yy}}\left\{M_s - (I_x-I_z)rp - m[z_g(\dot{u}-vr+wq) - x_g(\dot{w}-uq+vp)]\right\} \\ \dot{r} = \frac{1}{I_{zz}}\left\{N_s - (I_y-I_x)pq - m[x_g(\dot{v}-wp+ur) - y_g(\dot{u}-vr+wq)]\right\} \end{cases} \tag{12}$$

where m is the mass of the object. $\mathbf{x}_G = (x_g, y_g, z_g)$ is the vector pointing from the center of rotation $\mathbf{x}_{rot}$ to the center of gravity $\mathbf{x}_{cog}$ in the Ship system

$$\mathbf{x}_G = \mathbf{x}_{cog} - \mathbf{x}_{rot} = (x_g, y_g, z_g) = (x_{cog}, y_{cog}, z_{cog}) - (x_{rot}, y_{rot}, z_{rot}) \tag{13}$$

and I_{xx}, I_{yy} and I_{zz} are the moments of inertia around center of rotation, obtained from

$$\begin{cases} I_{xx} = I_{xcg} + m\left({y_g}^2 + {z_g}^2\right) \\ I_{yy} = I_{ycg} + m\left({x_g}^2 + {z_g}^2\right) \\ I_{zz} = I_{zcg} + m\left({x_g}^2 + {y_g}^2\right) \end{cases} \tag{14}$$

in which I_{xcg}, I_{ycg} and I_{zcg} are measured respect to the center of gravity. The center of rotation coincides with center of gravity for free running ships.

The velocities $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$ are obtained by integration of the accelerations from Eq. (12) and then they are transformed back to the Earth system by Eq. (8). The integration is performed in the Earth system to obtain the translations and rotations of the ship.

Before applying the motions to the ship grids, the appendages (propeller and rudder) need to be moved based on the propeller and rudder rotation angles. The grids for the propeller or rudder rotate about an axis fixed on the ship using the following equation:

$$\mathbf{x}_{ap}^n = \begin{bmatrix} c+u_1^2(1-c) & u_1u_2(1-c)-u_3s & u_1u_3(1-c)+u_2s \\ u_1u_2(1-c)+u_3s & c+u_2^2(1-c) & u_2u_3(1-c)-u_1s \\ u_1u_3(1-c)-u_2s & u_2u_3(1-c)+u_1s & c+u_3^2(1-c) \end{bmatrix}\left(\mathbf{x}_{ap}^0 - \mathbf{p}_a\right) + \mathbf{p}_a \tag{15}$$

where $c = \cos\alpha$, $s = \sin\alpha$ and α is the rotation angle of the appendage (propeller or rudder), $\mathbf{u} = (u_1, u_2, u_3)$ is a unit vector pointing on the rotation axis direction, $\mathbf{p}_a$ is the location of an arbitrary point located on the rotation axis, $\mathbf{x}_{ap}^0$ is the initial position of a mesh point in the appendage, and $\mathbf{x}_{ap}^n$ is the position of the mesh point after the rotation. All variables are computed in the Ship system. The strategy to compute the rotation angles of the appendages depends on the type of problem to solve. For self-propulsion the rotation angle of the propeller is obtained using a PI controller that seeks the target ship speed by acting on the propeller rotational speed (in RPS or revolutions per second). The rudder rotation angle is fixed at zero. In the zigzag maneuvers, the propeller rotation angle is calculated from a constant RPS condition while the rudder angle is obtained by a zigzag controller, which executes the rudder according to the heading angle of ship and the rudder angle history.

After the grids for the appendages are transformed, the ship grids, including the appendage grids, are translated and rotated as

$$\mathbf{x}_{ship}^n = \mathbf{J}_1\cdot\left(\mathbf{x}_{ship}^0 - \mathbf{x}_{rot}\right) + \mathbf{x}_{rot} + \boldsymbol{\eta}_1. \tag{16}$$

where $\mathbf{x}_{ship}^0$ are the initial positions of all grids points belonging to the ship and $\mathbf{x}_{ship}^n$ is the new position after the motions are applied.

The background grids, used to capture the free-surface and impose the far-field boundary conditions, moves with the ship but are restricted only to surge, sway and yaw motions.

4. Ship hydrodynamics applications

4.1. Open-water curves for KP505 propeller

The first application case is prediction of the open-water curves for the KP505 propeller. This case is selected for two purposes. First, it is used to validate the basic dynamic overset grid strategy and implementation by comparison with non-overset grid results and experimental data. Second, the open water propeller performance is needed for a full CFD prediction of the self-propulsion factors. The principal particulars of the KP505 propeller are listed in Table 1.

An earth-fixed frame of reference is used where the propeller is moving with constant forward and rotational speeds. During the runs the RPS is fixed and the advance speed is chosen to achieve the desired advance coefficient J. Thrust and torque coefficients K_T and K_Q, and efficiency η_0 for each advance coefficient are obtained from the thrust and torque, all defined as

$$K_T = \frac{T}{\rho n^2 D^4} \tag{17}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \tag{18}$$

$$\eta_0 = \frac{JK_T}{2\pi K_Q} \tag{19}$$

$$J = \frac{V_A}{nD} \tag{20}$$

where T and Q are the thrust and torque, D is the diameter of propeller, n is the RPS and V_A is the advance speed. n=9.5 RPS is chosen based on experimental data for self-propulsion. The experimental data are available from the Tokyo 2005 CFD Workshop (Hino, 2005).

Table 1
Main particulars of propeller KP505.

Main particulars	Symbol	Value
Diameter	D (mm)	250
Mean pitch ratio	P_{mean}/D	0.950
Area ratio	A_e/A_o	0.800
Hub ratio	d_h/D	0.180
Number of blades	Z	5
Section profile		NACA66

To evaluate the performance with respect to traditional approaches, both non-overset and overset grid systems are used. For the non-overset approach, all cells rotate with the propeller as a rigid body, as shown in Fig. 5(a). Fig. 5(b) shows the overset strategy, where the boundary layer grid wraps the propeller and resolves the local flow, and is embedded in the background grid that covers the computational domain and is used to impose the far-field boundary conditions. The boundary layer grid rotates with the propeller while the background grid is fixed. The computational domain extends 5 propeller diameters upstream, 20 diameters downstream and 10 diameters laterally. All grids are generated with SnappyHexMesh, an automatic grid generation tool available in the OpenFOAM package. The grids consist of hexahedral cells with octree topology, as shown in Fig. 6 that displays the overset grids colored in different colors. The overset and non-overset grid systems share the same size of computational domain and same local refinements to emphasize differences between overset and non-overset results and minimize other effects. The grid sizes for non-overset and overset grids are listed in Table 2. Though the overset grid has more cells than the non-overset grid, the difference is mostly due to the overlapping area that is covered on fringe or hole points, with the number of active cells essentially the same for both grids.

Both non-overset and overset grids have the same boundary conditions as shown in Fig. 7. The boundary condition at the inlet imposes zero velocity because the computational domain moves forward with the advance velocity and the boundary is assumed to be far enough from the propeller to result in undisturbed fluid.

The computations use the single-run procedure described in Xing et al. (2008), in which the propeller is towed with a small acceleration to cover a wide range of advance velocities in a single run. Additional static runs are performed at three constant advance ratios, $J=0.6$, 0.7 and 0.8, to validate the single-run procedure.

The resulting open-water curves are shown in Fig. 8. In the single-run procedure, used for both overset and non-overset approaches, the propeller accelerated from $J=0.05$ to 1.05 in 5 s. The results using overset and non-overset grids are almost identical. The single-run procedure predicts the open-water

Table 2
Grid sizes for KP505 propeller computations.

	Overset	Non-overset
Grid size	2.47 M	1.86 M

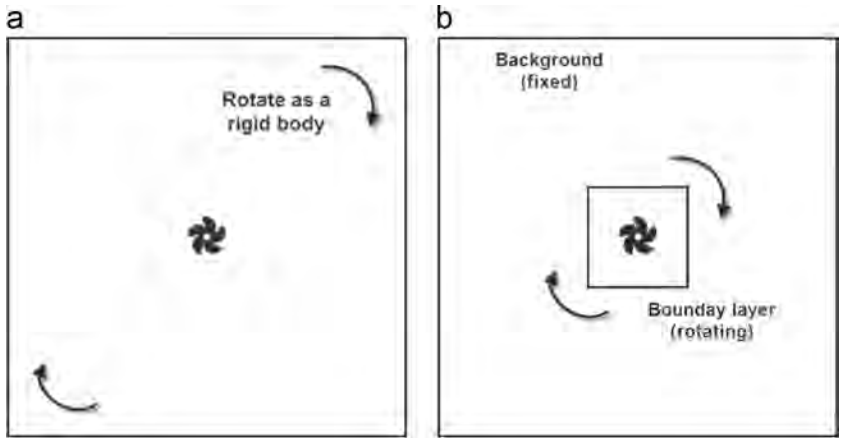


Fig. 5. Two different grid strategies to compute open water curves: (a) non-overset and (b) overset.

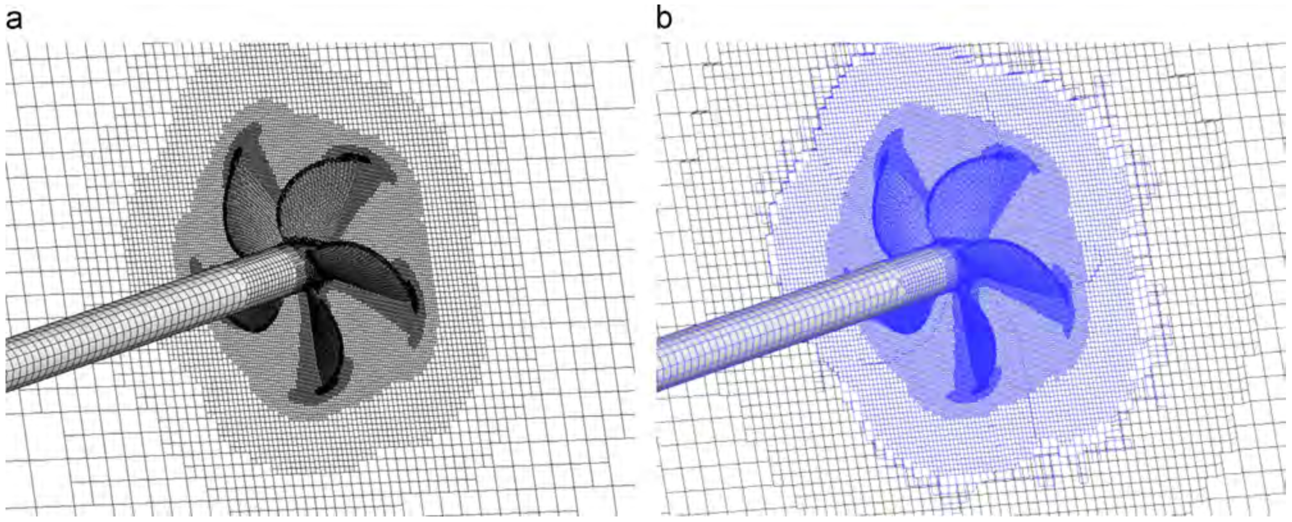
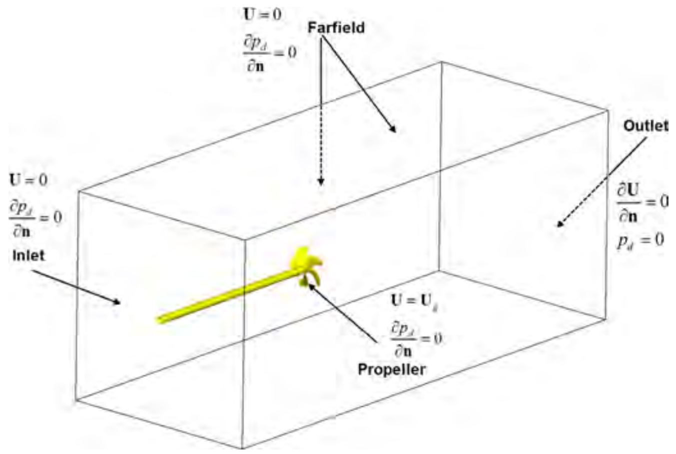


Fig. 6. Grids used for open water computations: : (a) non-overset and (b) overset.



$U=0$
$\frac{\partial p_d}{\partial n}=0$
Farfield
Outlet
$\frac{\partial U}{\partial n}=0$
$p_d=0$
$U=0$
$\frac{\partial p_d}{\partial n}=0$
Inlet
$U=U_b$
$\frac{\partial p_d}{\partial n}=0$
Propeller

Fig. 7. Boundary conditions for open water computations with KP505.

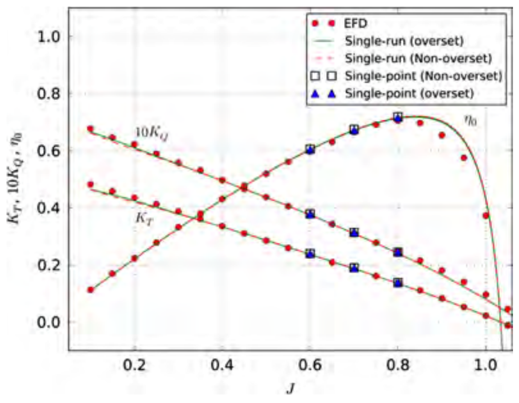


Fig. 8. Open water curves for experiments (circles), single-run (solid lines) and single-point (triangles) procedures.

curves very well, and shows good agreement with experimental data, except for $J>0.8$, where η_0 is overpredicted. The results of single-point procedure ($J=0.6$, 0.7 and 0.8) closely match the curves predicted by the single-run procedure, indicating that the single-run procedure is capable of predicting the open-water curves with similar accuracy as the single-point procedure.

The computations of two single-run tests are performed with 72 processors (Xeon 5650, 2.67 GHz) at the Helium HPC cluster of the University of Iowa. 70 processors are assigned to OpenFOAM and 2 are used by Suggar for DCI computation. For non-overset computations, all 72 processors are assigned to OpenFOAM. The time step for both cases is set to $\Delta t = 1.5\times10^{-4}$ s. The wall clock times are 45 and 58 h for the non-overset and overset computations, respectively. There are two reasons causing the difference in computation time. First, overset grids have more grid cells than the non-overset grid. The additional cells are mainly hole and fringe cells, which require additional time and memory to be discretized and included in the matrices. Second, two processors are assigned to Suggar in the overset case, and thus OpenFOAM runs with two processors less than the non-overset computation.

Fig. 9 shows vortical structures using isosurfaces of $Q=500$, the second invariant of the velocity gradient tensor, for solutions computed with the single-point approach. The tip vortices are clearly resolved, showing that as the advance coefficient increases the angle of attack decreases with the consequent decrease in propeller load and strength of the tip vortices. The vortex-pairing effect mentioned in Carrica et al. (2010a) and Castro et al. (2011) is not observed. This is likely due to the use of RANS, which produces significant turbulent viscosity that prevents full resolution of separated vortices, and the use of a relatively coarse grid. Carrica et al. (2010a) and Castro et al. (2011) employed detached eddy simulation (DES), resolving more turbulence features than RANS. Despite of the limitations of RANS method, the integral values K_T, K_Q and η_0 are well predicted by the dynamic overset grid approach.

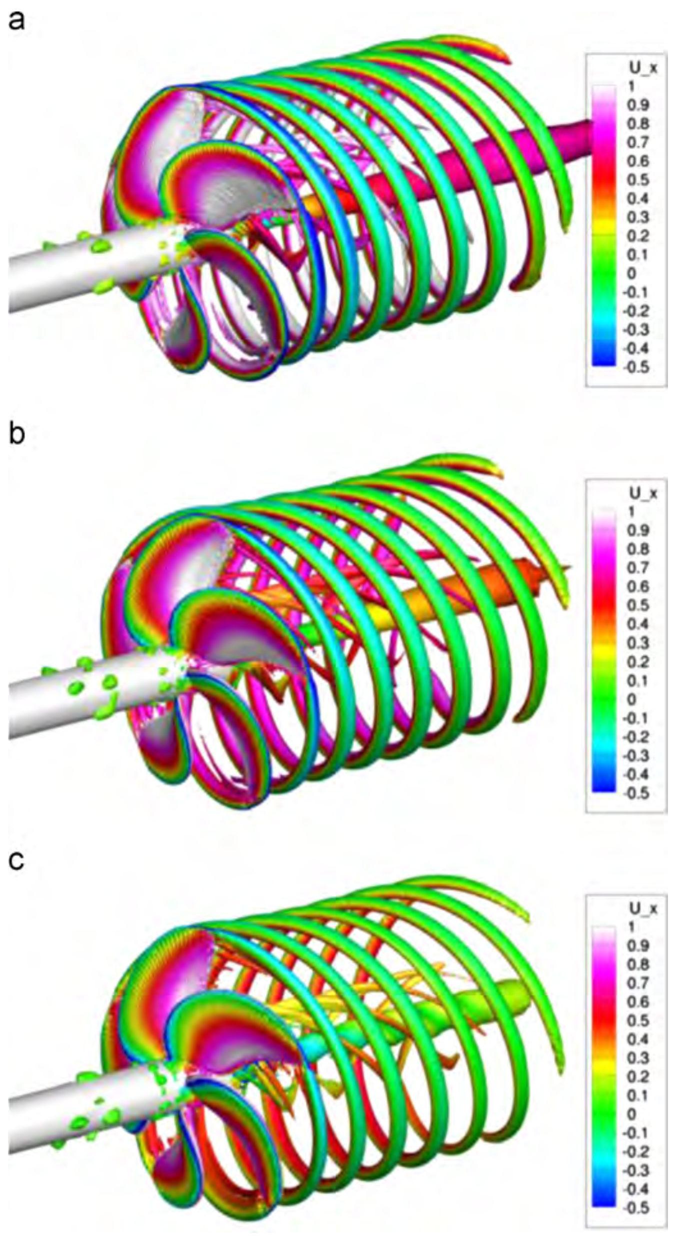


Fig. 9. Iso-surfaces of $Q=500$ for different advance coefficients obtained using the single-point procedure with overset grids: (a) $J=0.6$, (b) $J=0.7$ and (c) $J=0.8$.

4.2. *Towed KCS without propeller*

The second application case involved the KCS model without appendages. The goal of this case is to validate the ability of VOF method to capture the free surface when overset grids are used. There are strong discontinuities in pressure and density on the free surface interface that are challenging to the overset methodology. This case is also needed to obtain the hull resistance $R_{T(Tow)}$ and nominal wake W_n for prediction of the self-propulsion factors in the next subsection. The converged solution of the towed

Table 3
Geometrical properties of KCS model.

Main particulars	Symbol	Model scale	Full scale
Scale factor	λ	31.6	
Length between perpendiculars	L_{PP} (m)	7.2786	230
Length of waterline	L_{WL} (m)	7.3576	232.5
Maximum beam of waterline	B_{WL} (m)	1.019	32.2
Draft	T (m)	0.342	10.8
Displacement	Δ (m^3)	1.649	52,030
Wetted area without rudder	A_W (m^2)	9.4376	9424
Block coefficient	CB (m)	0.6505	0.6505
Longitudinal center of buoyancy, fwd+	LCB (%L_{PP})	−1.48	−1.48
Vertical center of gravity (from keel)	KG(m)	0.230	7.28
Moment of inertia	K_{yy}/L_{PP}	0.25	0.25

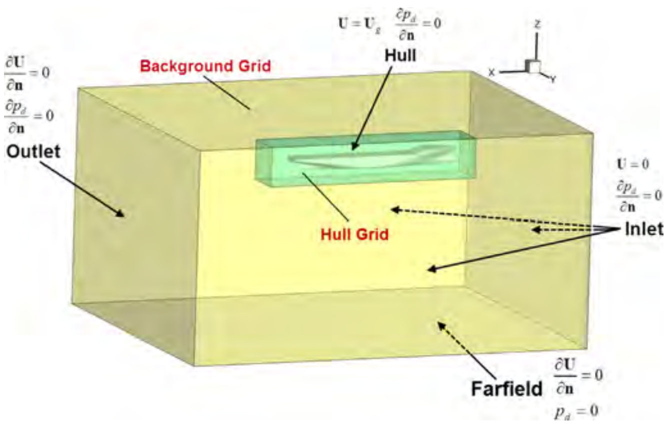

$U=U_g$ $\frac{\partial p_d}{\partial n}=0$
Hull
Background Grid
$\frac{\partial U}{\partial n}=0$
$\frac{\partial p_d}{\partial n}=0$
Outlet
$U=0$
$\frac{\partial p_d}{\partial n}=0$
Inlet
Hull Grid
Farfield
$\frac{\partial U}{\partial n}=0$
$p_d=0$

Fig. 10. Design of the overset grid system for the bare hull KCS.

Table 4
Details of the overset grids for bare hull KCS.

	Hull	Background	Total
Mesh size	959,401	716,064	1,675,465

Table 5
Resistance coefficients and nominal wake.

	Experiment	Present Work	% Error	CFDShip-Iowa (DES)
C_T	3.55×10^{-3}	3.52×10^{-3}	−0.958	3.58×10^{-3}
C_P	7.18×10^{-4}[a]	6.99×10^{-4}	−2.674	7.37×10^{-4}
C_F	2.83×10^{-3}[b]	2.82×10^{-3}	−0.530	2.84×10^{-3}
W_n	0.686	0.742	8.120	0.723

[a] Computed by $C_P=C_T-C_F$.
[b] By ITTC 1957 friction line $C_F=0.075/(\log_{10}Re-2)^2$.

condition was also used as initial condition of the self-propulsion to achieve faster convergence.

The KCS model, developed by the Korean Maritime and Ocean Engineering Research Institute (MOERI, formerly KRISO), was conceived to provide data for both study of flow physics and CFD validation for a modern container ship. Detailed geometrical properties of the KCS model are listed in Table 3. This KCS model is fitted with the five-bladed propeller KP505; Table 4 shows the main particulars. The ship model is fixed at even-keel condition with a service speed of 2.196 m/s, corresponding to $Fr=0.26$ and $Re=1.4\times10^7$. This was one of the benchmark cases in the CFD Workshops of Tokyo 2005 and Gothenburg 2010, and high-quality data is available for comparison.

The design of the overset grid system for the bare hull KCS is shown in Fig. 10. Two overset grids are used: the hull grid resolves the near flow around the ship and the background grid accommodates the far-field boundary conditions and is refined at the free-surface. The sizes of these grids are listed in Table 4. The full geometry without symmetry is used so that the solution of the towed condition can provide the initial condition for the self-propulsion computation. The boundary conditions are also presented in Fig. 10, showing that the computations are performed in the Earth reference system.

The computation is performed with 24 processors (Xeon 5650, 2.67 GHz) at the Helium HPC cluster. Since the KCS model is fixed during the simulation, the DCI is computed once as a pre-processing step and no processors are assigned to Suggar. The time step is $\Delta t=5\times10^{-3}$ s and the simulation is performed for 26 s model scale time. The total wall clock time per run is 6.9 h.

Table 5 shows the results of the computations in towed condition. Results are compared against experimental measurements from NMRI available from the Tokyo 2005 CFD Workshop (Hino, 2005) and the computational results from CFDShip-Iowa v4.5 with DES (Carrica et al., 2010a). The comparisons of resistance show excellent agreement between CFD and EFD, but the nominal wake W_n is significantly overpredicted by 8.12%, though close to the result of CFDShip-Iowa. Fig. 11 depicts the wave patterns. The surface is very well resolved, showing excellent agreement between CFD and EFD. This is further stressed by CFD and EFD comparisons of wave elevation at three different lateral sections

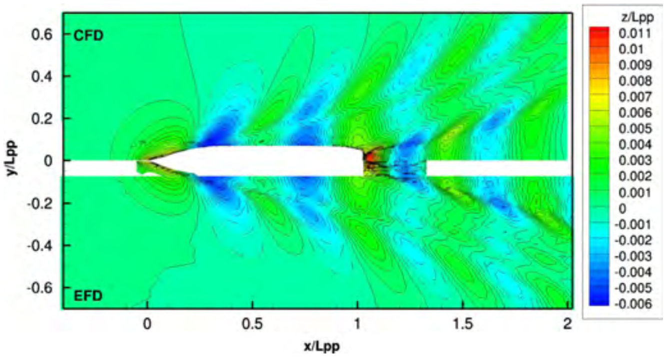


Fig. 11. Wave-elevation at $Fr=0.26$ for towed condition.

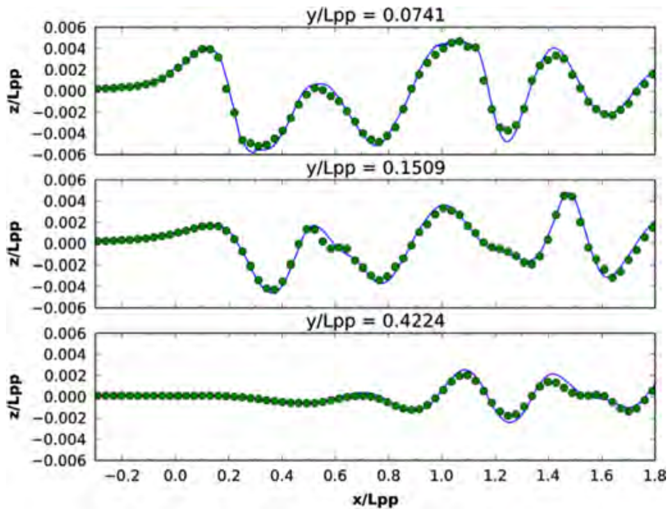


Fig. 12. Comparison of free-surface cuts at different lateral positions between experiments (circles) and computational results (line).

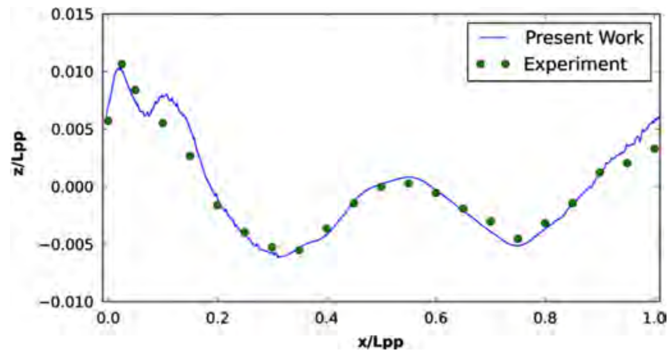


Fig. 13. Wave profile on the hull surface.

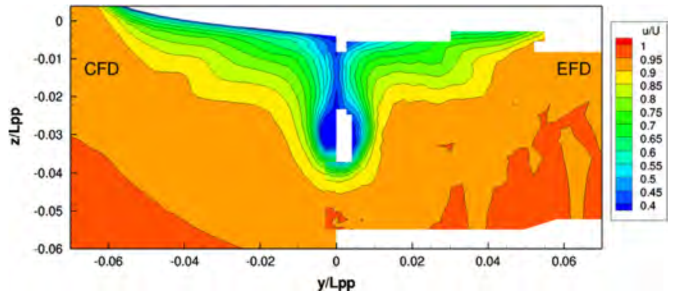


Fig. 14. Axial velocity contours on the propeller plane (x/L_{pp}=0.9825).

(y/L_{pp}=0.0741, 0.1509 and 0.4224) as shown in Fig. 12. There is no noticeable discontinuity caused by the overset interpolation on the surface interface. Fig. 13 shows the wave profile on the hull, which is also well predicted with exception of small discrepancies at the bow shoulder and stern. Fig. 14 shows axial velocity contours at the propeller plane, again exhibiting good agreement with experimental measurements.

4.3. Self-propelled KCS

For self-propulsion, KCS is equipped with a directly rotating propeller, which provides the thrust for the ship to advance. The self-propulsion is performed at ship point by adding a skin friction correction force (SFC) of the form

$$\mathrm{SFC} = \{(1+k)(C_{F0M} - C_{F0S}) - \Delta C_F\} \times \tfrac{1}{2}\rho U_0^2 A_W \tag{21}$$

where $C_{F0M} = 2.832 \times 10^{-3}$ and $C_{F0S} = 1.378 \times 10^{-3}$ are frictional resistance coefficients at model and ship scale, respectively, obtained from ITTC 1957 frictional line $C_{F0} = 0.075/(\log_{10} Re - 2)^2$; $\Delta C_{F0} = 0.00027$; $1+k=1.1$ from experimental data, $U_0 = 2.196$ m/s is the reference velocity and A_W is the reference area. The resulting skin correction force is SFC=30.25 N.

Following Carrica et al. (2010a), 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

$$n = Pe + I\int_0^t e\,dt \tag{22}$$

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

$$e = U_{target} - U_{ship} \tag{23}$$

The PI controller is activated at the beginning of the computation. It updates the RPS of the propeller at the end of each time step until the longitudinal force reaches the final balance:

$$T = R_{T(SP)} - SFC \tag{24}$$

where $R_{T(SP)}$ is total resistance of the self-propelled KCS model.

The computation is performed with 72 processors in the Helium HPC cluster. Four of the 72 processors are assigned to Suggar for DCI computation. The time step is $\Delta t = 1.5 \times 10^{-4}$ s. The simulation extends for 16 s in model scale and the overall wall clock time is 225 h.

Fig. 15(a) illustrates the basic layout of the overset grid strategy and boundary conditions. Compared with the design shown in Fig. 10 for the bare hull, one more overset grid is used to add the propeller. The boundary conditions are identical with zero velocity and zero gradient of pressure imposed on inlet and far-field boundaries. The propeller grid is fitted behind the ship hull and rotating about the propeller shaft, as shown in Fig. 15(b). The overset grids, generated with SnappyHexMesh, are depicted in Fig. 16, including grids on solid surfaces, a transversal section crossing the propeller plane and the center-longitudinal section. Local refinement is used to capture the propeller vortices. The size of each component grid is shown in Table 6 for a total of 3.21 M cells. Notice that there is a physical gap between the propeller and the hull, to prevent occurrence of orphans in this region.

The initial condition for the self-propulsion computation was interpolated from the final solution of the towed condition with the tool *mapFields* in OpenFOAM. This pre-processing step saves computational time by starting with a developed boundary layer. The initial ship speed was set to the target cruise speed of 2.196 m/s and the propeller was static. The proportional and integral constants of the PI controller were set to P=2000 RPS s/m and I=2000 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. The chosen PI constants were obtained with a simple simulator code before the self-propulsion computation was carried out. The basic idea of the simulator code is predict the time histories of ship speed and propeller RPS with given thrust coefficient (K_T) and hull resistance (C_T). K_T and C_T can be obtained from experiments or empirical formulae. Suitable P and I constants are found by running the code multiple times.

The time histories of RPS and ship speed are shown in Fig. 17(a). The computation was run for 16 s of model scale time. Fig. 17(b) shows a close up of the convergence history of ship speed and RPS during the first 6 s. The RPS starts at 0 and increases rapidly. The ship speed starts at the target speed (2.196 m/s) and initially drops because the propeller rate is accelerating and initially does not provide enough thrust to maintain the speed. As the RPS increases, the ship speed reaches the lowest speed and returns to the target slowly. After 6 s, the ship speed has reached the target speed and the RPS converged to the self-propulsion point.

Table 7 lists the self-propulsion factors as obtained by the CFD computations following ITTC procedures (ITTC, 2002). The self-propulsion factors are calculated using the open-water curves in Fig. 8 computed with overset grids and single-run procedures using the thrust identity method. The results are also compared with results obtained with CFDShip-Iowa v4.5 in Carrica et al. (2010a), which used the experimental open-water curves for KP505 measured at NMRI. The largest error occurs for the relative rotative efficiency (η_R), underpredicted by 2.955%, but it is close to the one predicted by CFDShip-Iowa. All factors match well the experimental data, with errors within 3.0%.

Axial velocity contours (normalized by ship speed) at a plane downstream of the propeller are shown in Fig. 18. The CFD predictions compare very well with experiments, even for the relatively coarse grid used in the computations. The maximum velocity occurs on the starboard side, in which the momentum

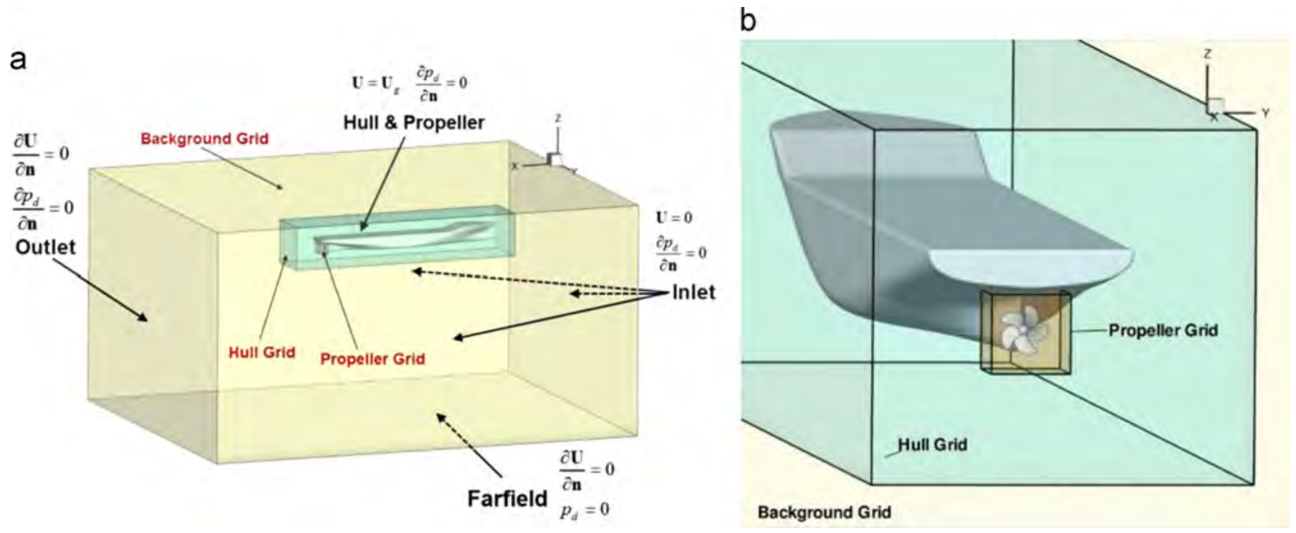

a
$\mathbf{U}=\mathbf{U}_g$ $\frac{\partial p_d}{\partial \mathbf{n}}=0$
Hull & Propeller
Background Grid
$\frac{\partial \mathbf{U}}{\partial \mathbf{n}}=0$
$\frac{\partial p_d}{\partial \mathbf{n}}=0$
Outlet
$\mathbf{U}=0$
$\frac{\partial p_d}{\partial \mathbf{n}}=0$
Inlet
Hull Grid
Propeller Grid
Farfield
$\frac{\partial \mathbf{U}}{\partial \mathbf{n}}=0$
$p_d=0$
b
Propeller Grid
Hull Grid
Background Grid

Fig. 15. Overset grid system and boundary conditions for self-propelled KCS: (a) global view and (b) close view.

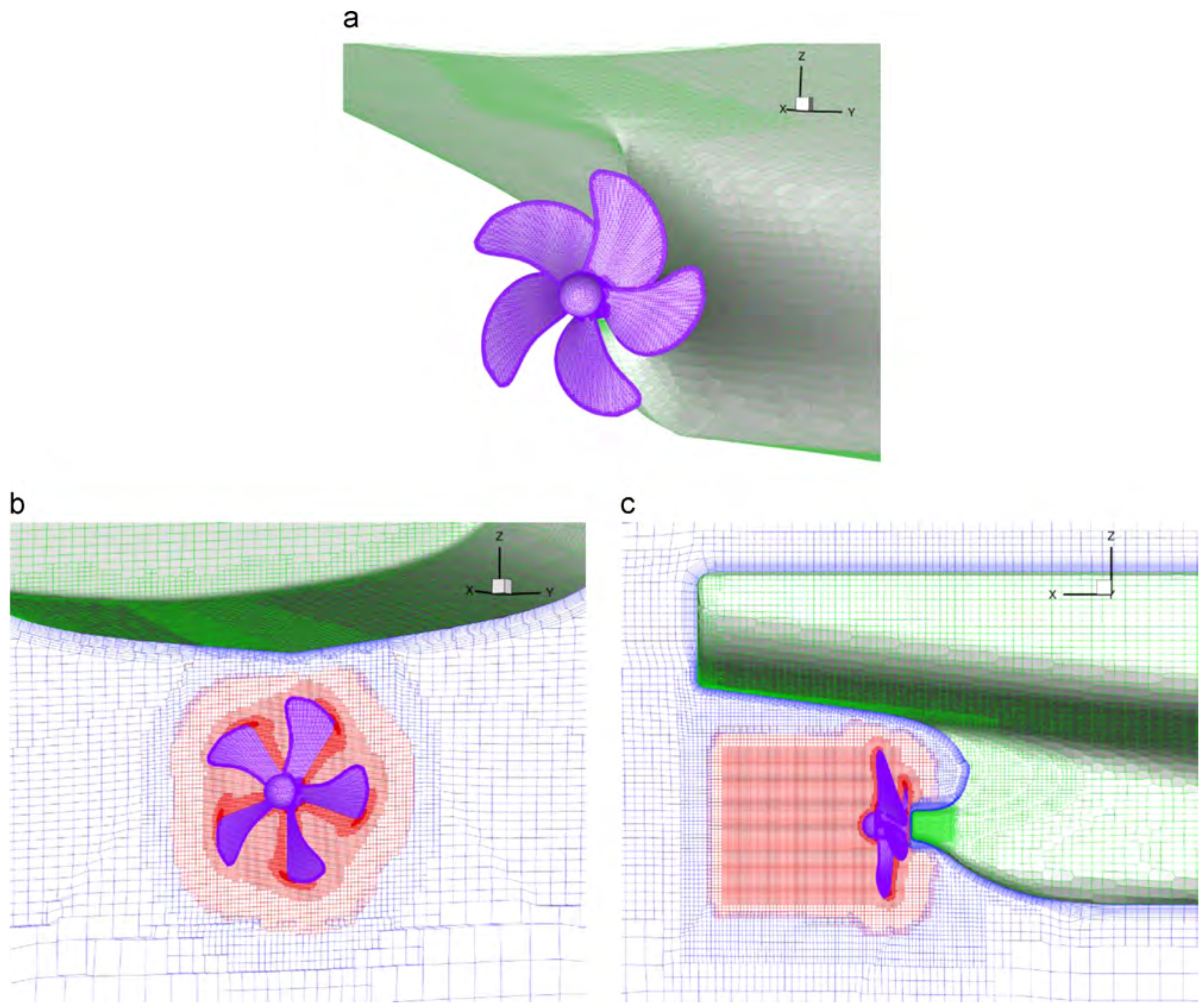


Fig. 16. Mesh views for self-propelled KCS: (a) Surface mesh of hull and propeller, (b) transversal cross section and (c) longitudinal section.

imparted by the propeller is maximum, while the momentum loss downstream of the propeller hub is properly predicted. In this area a strong hub vortex can observed, as discussed next.

Fig. 19 shows vortical structures displayed as isosurfaces of $Q=250$ colored by axial velocity. In Fig. 19(a), 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 and strength, as expected, so that it is still somewhat resolved by the coarser grid downstream of the refinement. Fig. 19(b) shows a stern view at self-propulsion, corroborating the asymmetric nature of the propeller wake flow. An interesting effect occurs when the blades pass the top position. At this location the wake velocity is lowest and the angle of attack the blades experience is largest. The maximum load on blades occurs right after the blades pass the top position, causing maximum strength in the blade tip vortices, suggesting a reason for the vortex breakdown to occur at the top right side of the propeller plane. The strength of the tip vortices then rapidly decreases as the blade moves to higher wake velocities (and higher local J) causing a breakdown and separation of the higher strength vortex which is then transported downstream and to the hub.

4.4. Zig-zag maneuvers

Zig-zag maneuver simulations are performed for the HSVA KCS model. This model shares the same hull shape of the model used in the previous self-propulsion computation, but has a different scale factor ($\lambda=52.667$) and a different propeller (SVP 1193). A semi-balanced horn rudder is installed at the stern of ship, mimicking the experimental conditions. Details of the propeller and rudder are summarized in Tables 8 and 9, respectively. The experiments were performed by HSVA in Germany in 2006, providing data of ship motion, propeller forces and moments for CFD validation (Steinwand, 2006). Two zig-zag maneuvers are performed in this work, a 10/10 standard maneuver and a 15/1 modified maneuver. In both cases the ship speed is 1.701 m/s, corresponding to $Fr=0.26$. The conditions for the two cases are listed in Table 10. Notice that the two cases in the experiment were performed at two different metacentric heights (GM), with the vertical location of center of gravity Z_g (defined herein as the vertical distance above water line, positive upwards) listed in Table 11.

Table 6
Details of the overset grids for KCS with propeller.

	Hull	Background	Propeller	Total
Mesh size	1,129,476	716,064	1,368,100	3,213,640

The overset grid layout is depicted in Fig. 20. The layout is similar to the one used for self-propulsion (see Fig. 15), but it has additional rudder grids as shown in Fig. 21. The propeller grid is regenerated according to geometry of the SVP 1193 propeller, while the hull grid is adjusted to provide more refinement around the rudder. The physical gaps existing in the model between rudder and horn are slightly enlarged to allow for an overset grid system without orphans for the coarse grid used herein. Notice that Mofidi and Carrica (2014) required over 6 times more grid points to resolve the actual rudder/horn gap. The final grids used are presented in Figs. 21 and 22, in which details of different grids and overlapping regions are shown. A total 3.82 M cells are used, and details of the grid overset grid system are listed in Table 12.

Mimicking the experiments in model scale, self-propulsion is achieved and the propeller rotational speed frozen before staring the maneuver. Since the simulations and experiments are performed in model scale, no SFC is needed. The ship is free to sink and trim during self-propulsion computations, and the same PI controller is employed to adjust the propeller speed until the ship target speed is reached. The final predicted RPM transformed to full scale is 117.3 per minutes, 1.04% larger than the experimental value (116.5 RPM).

The zig-zag maneuvers are restarted from the converged self-propulsion solution, freezing the propeller speed, and the rudder is executed according to the conditions listed in Table 10. To match the experimental data, results are reported in full scale, but based on the CFD coordinate systems used in this study.

The computations of the two zig-zag tests are performed with 96 processors (E5-2650 v2, 2.60 GHz) at Neon HPC cluster of the University of Iowa. Four of the 96 processors are assigned to Suggar for DCI computation. The time step is set to $\Delta t=5\times10^{-4}$ s. The simulation extends 48 s and 26 s model scale time for the 15/1 and 10/10 zig-zag maneuvers, respectively, taking about 263 h of wall clock time for the 15/1 zig-zag maneuver and 123 h for the 10/10 case.

Electronic Annexes I and II show two animations in real time at model scale of the 10/10 zig-zag maneuver. In Annex I the ship is colored with pressure and the free surface with velocity magnitude. The instantaneous time histories of rudder, yaw and roll motions are plotted to help observe the ship motions and rudder execution points. Unsteady pressure on the rudder and propeller and pressure oscillation at the stern induced by the propeller rotation can be observed. In Annex II the boundary layer is shown at four transversal sections colored with axial velocity in the ship coordinate system. The free surface is colored by wave elevation.

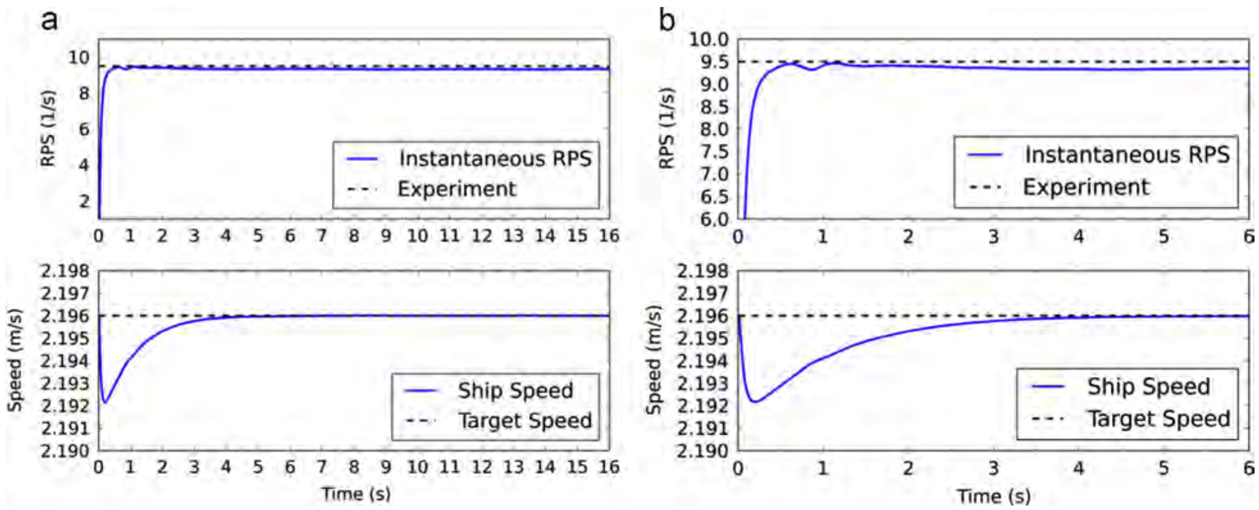


Fig. 17. Time histories of RPS and ship speed for the self-propulsion case: (a) global and (b) close up for $t=0$–6 s.

Table 7
Resistance coefficients and propulsion factors for self-propelled KCS.

	Experiment	Present Work	Error (%)	CFDShip-Iowa (DES)
Resistance coefficient, C_T	3.942×10^{-3}	3.840×10^{-3}	−2.586	4.011×10^{-3}
Thrust coefficient, K_T	0.17	0.1682	−1.061	0.1689
Torque coefficient, K_Q	0.0288	0.0290	0.863	0.02961
Thrust deduction, $1-t$	0.853	0.8857	2.237	0.8725
Effective wake coefficient, $1-W_t$	0.792	0.8721	−1.326	0.803
Open water efficiency, η_o	0.682	0.6785	−0.507	0.683
Relative rotative efficiency, η_R	1.011	0.9811	−2.955	0.976
Advance ratio, J	0.728	0.7363	1.142	0.733
Rate of revolution, n	9.5	9.3231	−1.862	9.62
Propulsive efficiency, η	0.74	0.7429	0.392	0.724

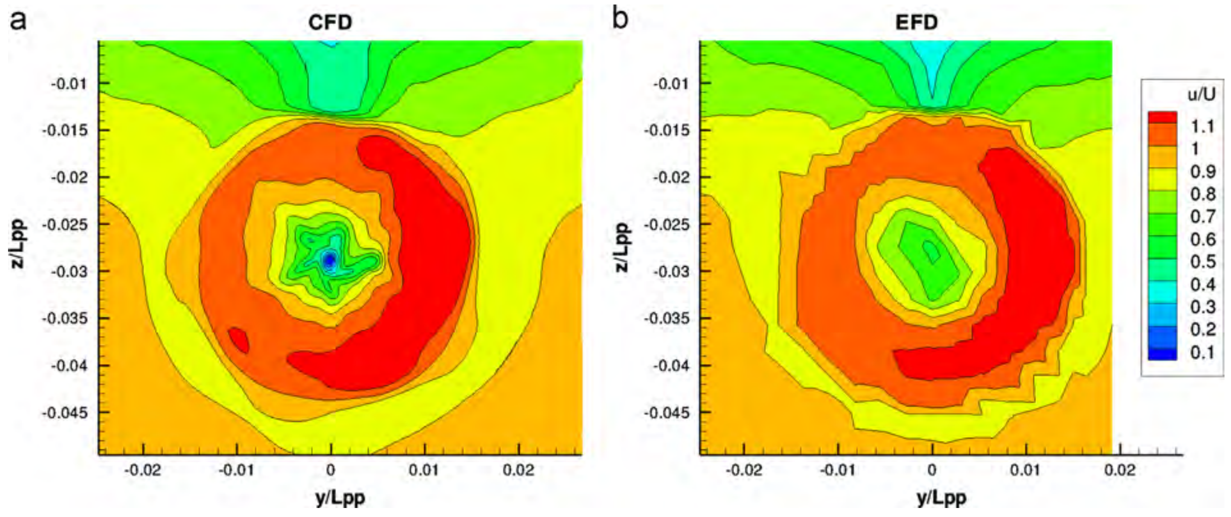


Fig. 18. Axial velocity contours downstream of the propeller plane ($x/L_{pp}=0.9941$).

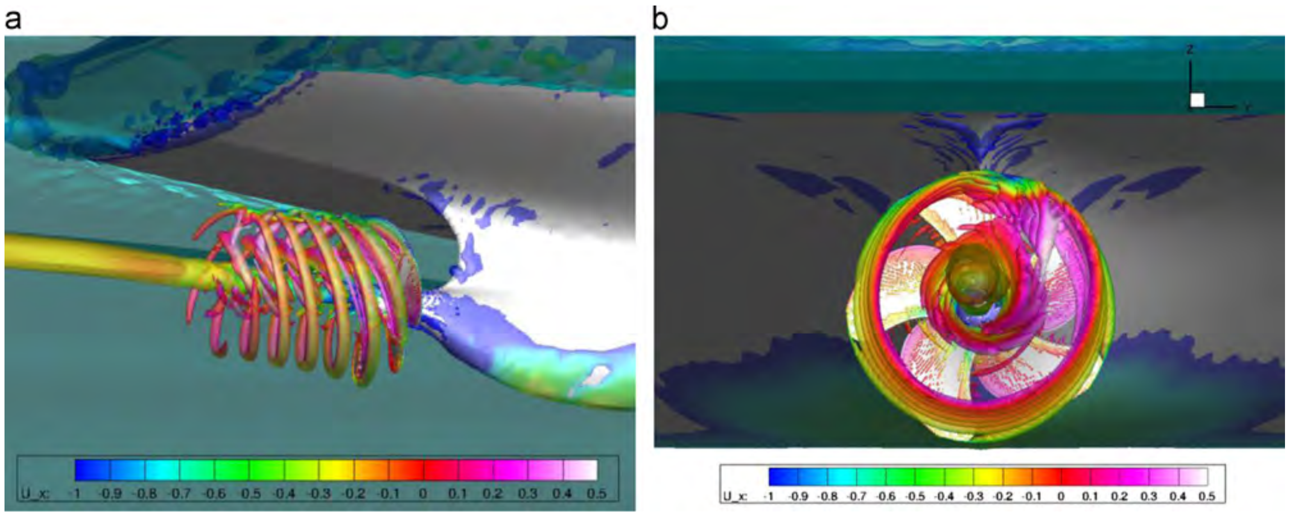

a
b
U_x: -1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5

Fig. 19. Propeller vortices represented by isosurfaces of $Q=250$ colored by axial velocity for self-propelled KCS: (a) side view and (b) stern view.

Table 8
Main particulars of SVP-1193 propeller (model scale).

Main particulars	Symbol	Value
Diameter	D (mm)	150.0
Pitch ratio	$P_{0.7}/D$	1.000
Area ratio	A_e/A_o	0.700
Hub ratio	d_h/D	0.227
Number of blades	Z	5
Direction of rotation		Right-handed

The evolution of the boundary layer as the drift angle changes are evident. An insert showing the axial velocity downstream of the rudder ($x/L_{pp}=1.018$) is used to depict the evolution of the complex flow downstream of the rudder during the zig-zag maneuver.

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.oceaneng.2015.07.035.

The results of ship motions and rudder angle for the modified 15/1 zig-zag maneuver are shown in Fig. 23. Fig. 23(a) displays time histories of yaw motion (heading angle) and rudder angle, showing

Table 9
Main particulars of rudder (model scale).

Main particulars	Value
Type	Semi-balanced horn rudder
Rudder area (m^2)	0.0415
Lateral Area (m^2)	0.0196
Turn rate (°/s)	16.8

Table 10
Conditions for zig-zag maneuvers (model scale).

Case name	Speed (m/s)	Heading change (deg)	Max rudder angle (deg)	Turn rate (°/s)	First execute rudder to
15/1	1.701	1	15	16.8	Starboard
10/10	1.701	10	10	16.8	Port

Table 11
Vertical locations of center of gravity with different GMs.

Case name	GM (full scale/m)	GM (model scale/m)	Zg (model scale/m)
15/1	0.6	0.011392	0.0636
10/10	1.1	0.020886	0.0530

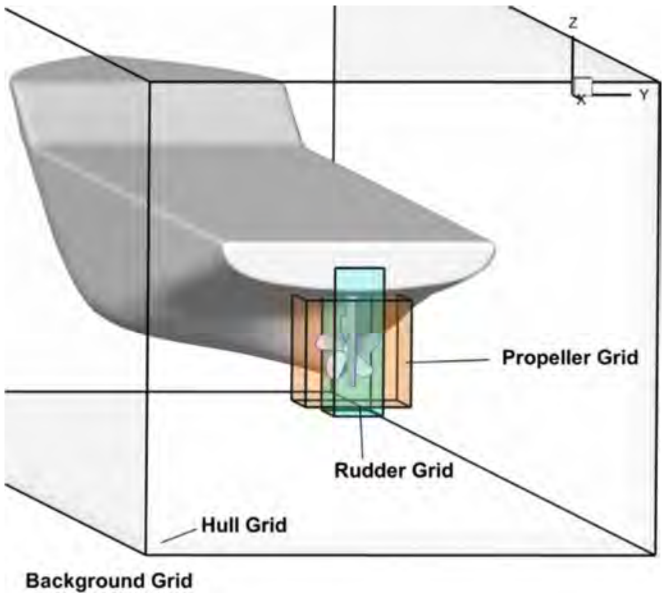


Fig. 20. Overset grid design for HSVA KCS model with rudder and propeller.

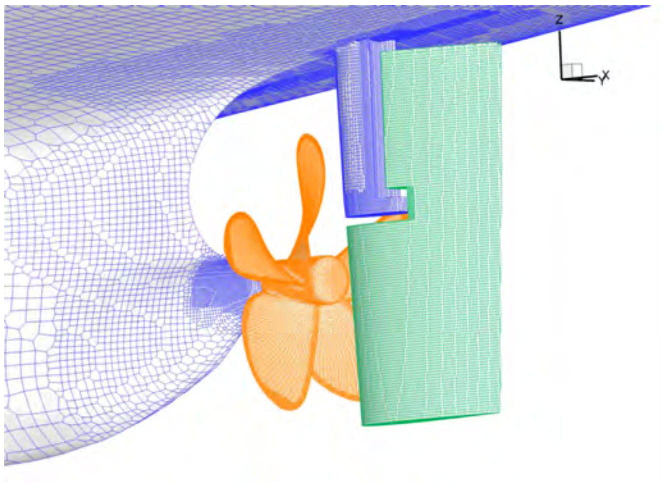


Fig. 21. Surface mesh for HSVA KCS model including propeller and semi-balanced horn rudder.

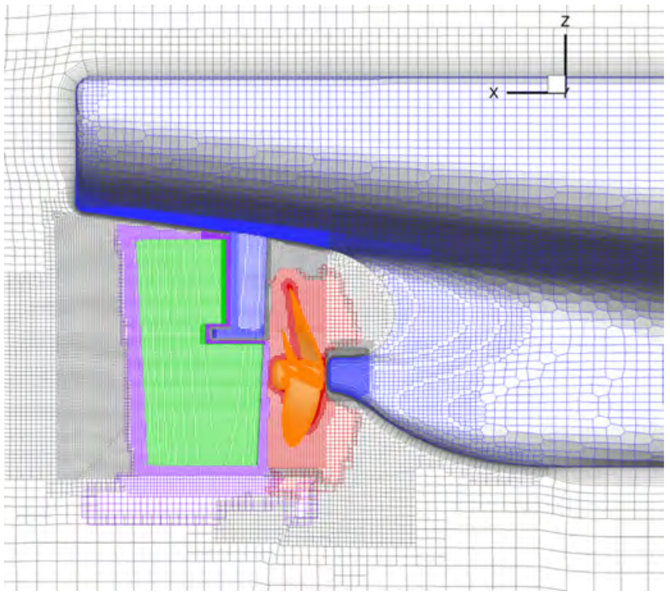


Fig. 22. Longitudinal section of HSVA KCS model.

Table 12
Grid sizes for overset grids of HSVA KCS model.

Grid name	Hull	Propeller	Rudder	Background	Total
Size	1,309,867	1,324,869	473,732	716,064	3,824,532

that the computational results agree well with the experiments. The positive maximum yaw angle (ship turns to port) is under-predicted compared with measurements. As the maneuver progresses the error increases for the next two zig-zag periods, but the maximum yaw angle on the starboard side (negative value) is much closer to experiments. The time history of yaw rate is shown in Fig. 23(b), showing good agreement with measurements in particular for the leading phase, but with amplitude slightly smaller than EFD, as expected from the under-prediction of heading angle in Fig. 23(a). The results of drift angle have a very similar trend with the time histories of yaw rate but in the opposite direction as shown in Fig. 23(c). The drift angle peaks at the points of maximum yaw rate, $t=112$, 178, 252 and 322 s, right after the ship reaches the check angle (1 degree for the 15/1 maneuver) and the rudder executes to the opposite side. As with the yaw rate, the predicted drift angle matches well the experimental data, but with slightly underestimated amplitude. As shown in Fig. 23(d), the ship starts at the service speed of 24 knots and drops to 20.8 knots at the end of the simulation due to the increase of resistance. CFD and EFD agree well in the first half of simulation, but show some discrepancies in the second half. Fig. 23(e) depicts the time evolution of roll motion, showing that a roll excursion occurs due to rudder-induced moments every time the rudder executes. The roll motion essentially follows the yaw rate with the same roll period of the maneuver, plus a higher frequency component due to the natural frequency of the ship, as can be seen in the roll rate shown in Fig. 23 (f). The roll oscillation at the natural frequency is excited every time the rudder is executed.

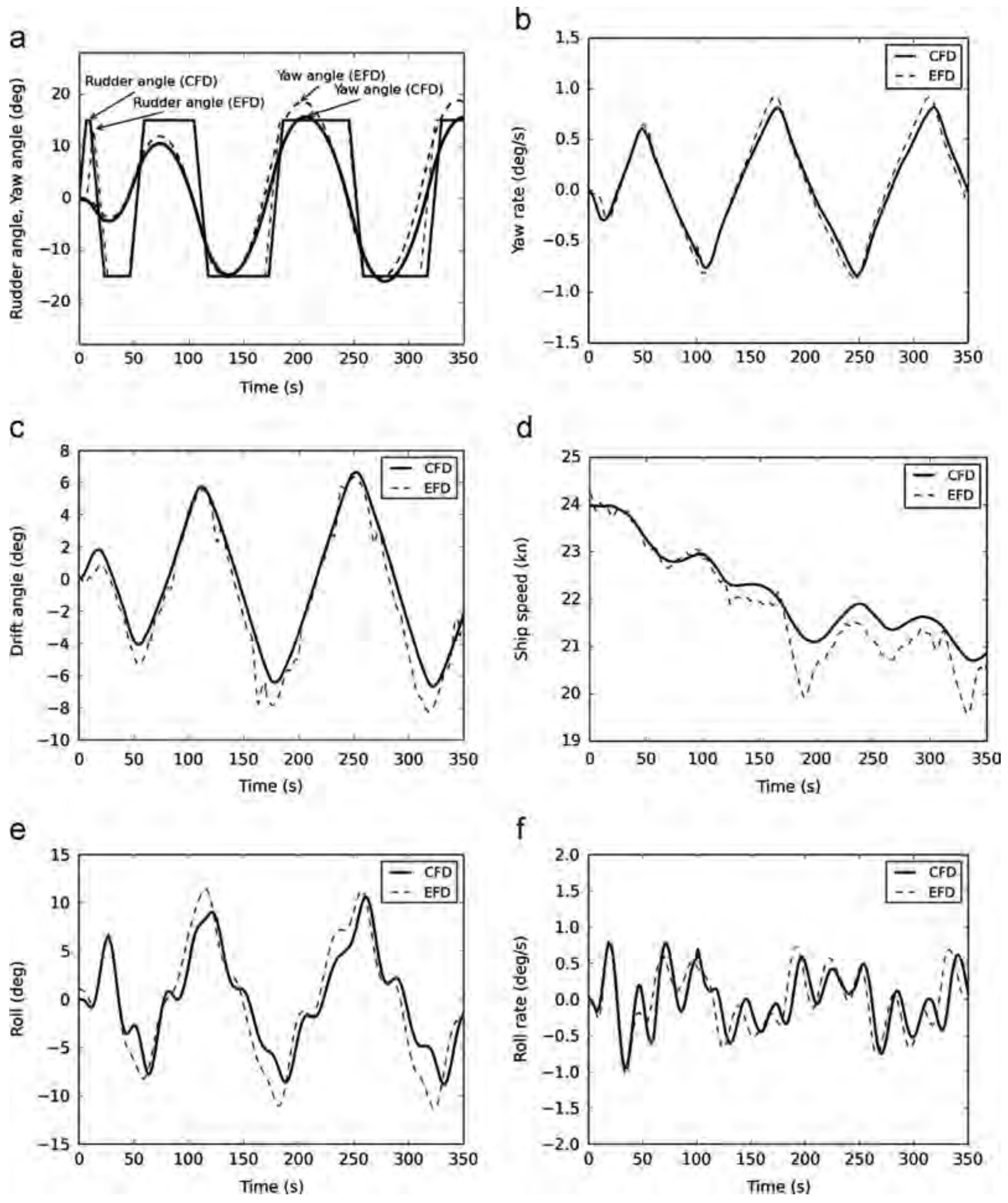


Fig. 23. Time histories of ship motions and rudder angles for 15/1 modified zig-zag maneuver: (a) rudder and yaw angles, (b) yaw rate, (c) drift angle, (d) ship spend and propeller RPM, (e) roll motion and (f) roll rate.

Fig. 24 illustrates the predicted ship motions and rudder angle for the 10/10 standard zig-zag maneuver. In this case, the experiments were limited to one full zig-zag period. Because the maximum rudder angle is smaller and the check change angle is larger than for the 15/1 zig-zag maneuver, it takes longer for the ship to finish one zig-zag period, resulting in more computational cost. The comparisons between CFD and EFD show similar trends as observed for the 15/1 zig-zag maneuver. The predicted maximum yaw angle is under-predicted compared with EFD, as shown in Fig. 24(a). The time history of yaw rate shows good agreement with data with a slightly under-predicted peak value, causing an under estimation of the heading overshoot angle in Fig. 24(a). The results of drift angle also show fair agreement with EFD, as shown in Fig. 24(c). The experimental data shows significant oscillations and the drift angle at the start of the maneuver is approximately -2 degrees instead of the expected zero degrees, indicating relatively large experimental uncertainties for the drift angle. The ship speed, shown in Fig. 24(d), shows considerably high fluctuations in the experiment, starting the maneuver with an approach speed of 25 knots instead of the nominal 24 knots, causing discrepancies between CFD and EFD. The maximum roll angle, at about 6.1°, is smaller than the maximum roll angle of 10.6° in the 15/1 maneuver, because the maximum rudder angle deflects to only 10°, showing that the rudder roll moments are significant in driving the roll motions of the ship. Though trends are good, the discrepancies seen in roll angle in Fig. 24(e) are likely due to the experimental initial roll angle of 2°, which provides roll energy absent in CFD. Fig. 24(f) depicts the time history of roll rate, showing large discrepancies during the period between 0 and 50 s, mostly due to the differences in initial ship speed and roll angle at the start of the maneuver. From 50 to 140 s the comparison shows similar trends between CFD and EFD, with leading phase errors. Beyond 140 s the agreement between simulation and experiment is good, indicating the initial errors of roll motions disappear

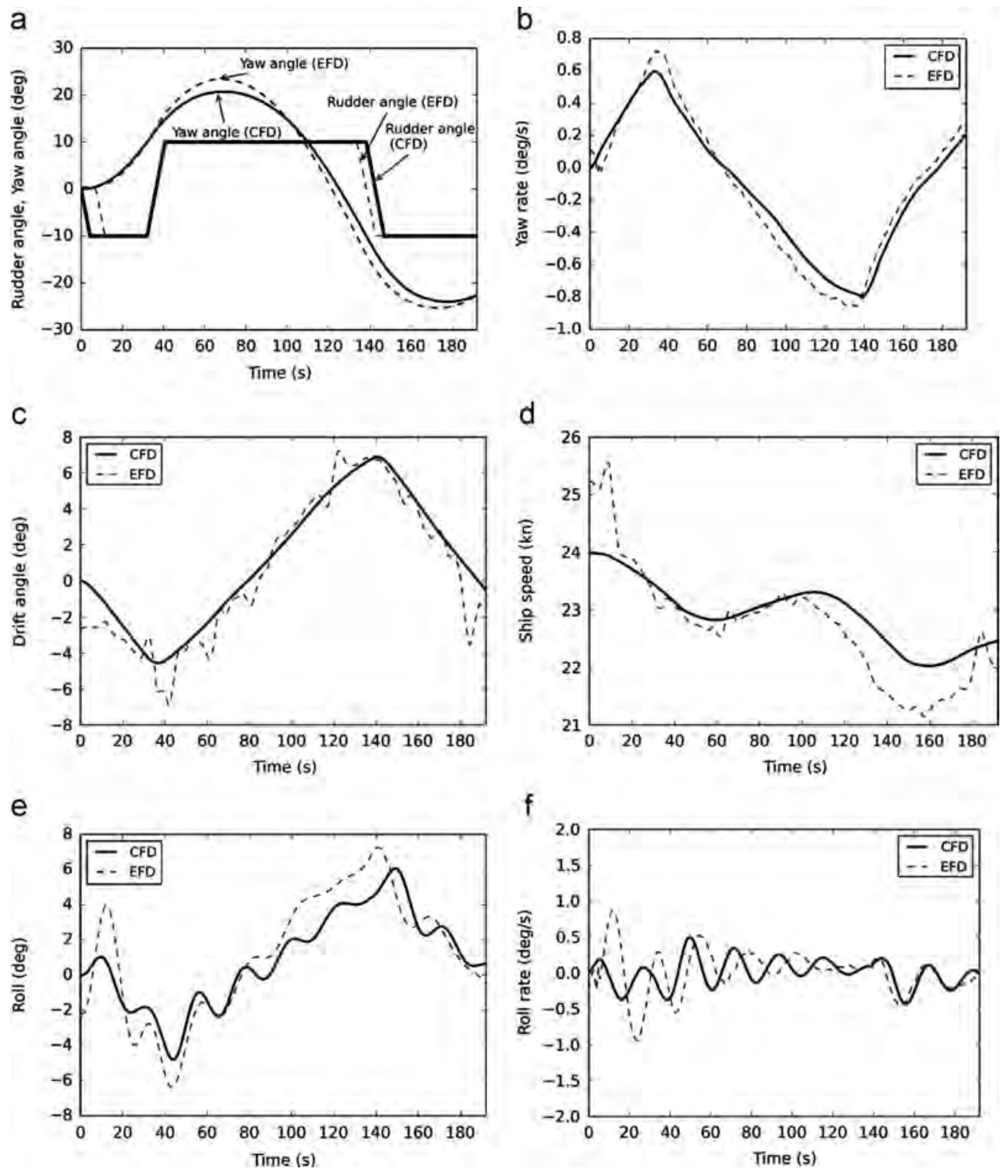


Fig. 24. Time histories of ship motions and rudder angle for 10/10 standard zig-zag maneuver: (a) rudder and yaw angles, (b) yaw rate, (c) drift angle, (d) ship speed, (e) roll motion and (f) roll rate.

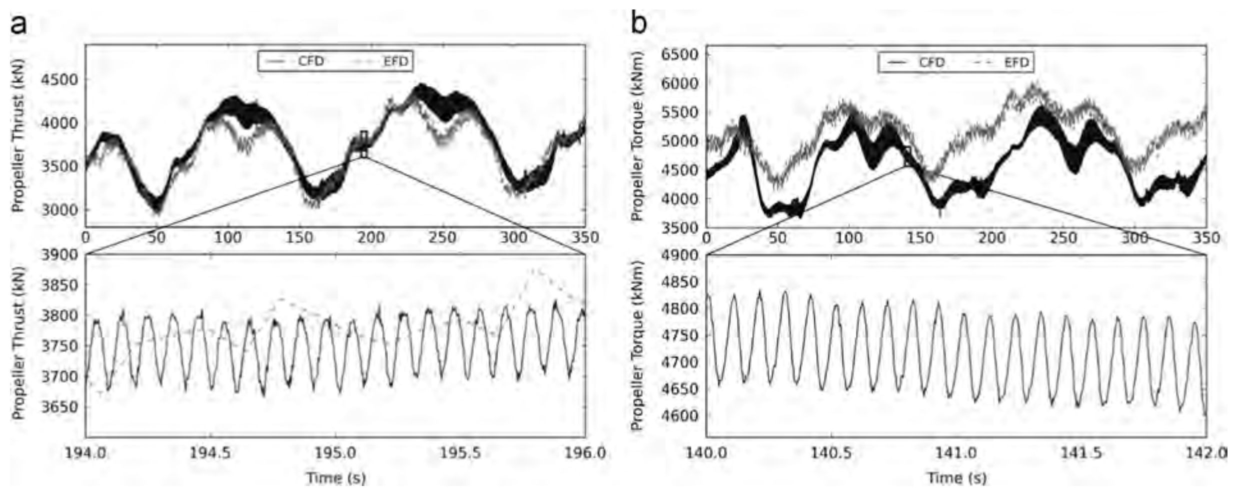


Fig. 25. Time histories of propeller thrust and moment for 15/1 modified zig-zag maneuver: (a) thrust and (b) torque.

as the test evolves to a periodic solution and the initial conditions decay.

Fig. 25 shows propeller thrust and torque for the 15/1 zig-zag maneuver. Fig. 25(a) depicts the time history of the propeller thrust, with the bottom section showing a detail of a shorter period of time. The predicted thrust closely agrees with the measurements, with thrust oscillations at five times the rotational frequency, due to the blade passage through the non-uniform wake flow at the propeller plane. The propeller thrust exhibits trends similar to the drift angle, indicating a strong relationship between the two variables. When the drift angle is positive the drift velocity points from port to starboard, increasing the angle of attack and load of the lower blade and decreasing the angle of attack and load of upper blades (the propeller is right handed, rotating clockwise looking from stern to bow). Because the advance velocity is much higher in the lower part of the propeller (due to the wake velocity, see Fig. 14), changes in the lower blades affect more the total force, resulting in an increase of overall propeller thrust. When the drift angle is negative, the drift velocity decreases the load on the lower blades, leading to a decrease in propeller thrust. Fig. 25(b) shows the time history of the propeller torque. The torque is computed with respect to the propeller axis, parallel to the *x*-axis in the ship system. The computational results show the same trends as the experiment, but the torque is underpredicted by about 10%. The reasons for this discrepancy have not been investigated. Fig. 26 shows the time histories of propeller thrust and torque for the 10/10 standard zig-zag maneuver. As in the case of the 15/1 maneuver, the propeller thrust agrees well between CFD and experiments, but the propeller torque is also underpredicetd by about 10%.

Fig. 27 shows instantaneous views of the wave patterns during the zig-zag maneuvers. Fig. 27(a) presents the start of the zig-zag maneuvers, with the ship in self-propulsion condition. Fig. 27(b) and (c) depict two instants of the 15/1 and 10/10 maneuvers, respectively, when the rudder just finished an execution from starboard to port side and reached the maximum angle. In both cases, the roll angle is close to maximum. For self-propulsion (Fig. 27(a)), good symmetry of the free surface in the farfield is obtained and the Kelvin waves, including diverging and transverse waves, are adequately resolved. In Fig. 27(b) and (c), a strong asymmetry in the wave patterns is observed due to the drift velocity and rotating trajectory. The bow waves at the port side are steeper and deeper than on the starboard side. The angle between diverging wave and ship centerline is larger for leeward side than windward side. Comparison between 15/1 and 10/10 zig-zag maneuvers shows little difference since the drift angles are close to each other, 5.6° for 15/1 and 6.1° for 10/10 maneuver. The larger roll angle in the 15/1 maneuver is evident in Fig. 27.

Fig. 28 illustrates the vortical structures represented by iso-surfaces of Q in the stern/propeller/rudder region for the 10/10 zig-zag maneuver. The iso-surfaces are colored with axial velocity in model scale. Fig. 28(a) shows the self-propulsion condition right before the zig-zag maneuver begins. The propeller tip vortices are clearly resolved, and interaction with the rudder causes the tip vortices to rise up on the port side of rudder. The tip vortices are quickly lost when the coarser region downstream of the refinement is reached, as the coarser grid is unable to resolve them. The propeller hub vortex also interacts with the rudder and propeller tip vortices, resulting in complex vortical interactions and structures. The hub vortex and resulting bigger structures can still be resolved by the coarser grids downstream of the rudder. Two small vortices shedding from the root and tip of the rudder are also observed. Fig. 28(b) depicts the instant when the rudder completes the execution from port to starboard side at $t = 45.7$ s. At this instant, the drift angle is $-3.93°$ and the rudder is deflected 10° to starboard. The propeller tip vortices are severely affected by the rudder deflection and interaction with the rudder is stronger. The higher angle of attack results in higher load and a stronger rudder tip vortex that interacts with the propeller hub vortex downstream of the rudder. The bilge vortex on the port side strengthens and is transported into the propeller flow. In Fig. 28(c) the rudder finishes an execution from starboard to port at $t = 142.2$ s. At this instant, the drift angle is 6.8° and the rudder angle is $-10°$ (deflected to port). This condition is similar with that Fig. 28(b) but the rudder deflects to the opposite direction and the drift angle increases. The bilge vortex, on the starboard side this time, is stronger due to increase of drift angle and maintains coherence after crossing the propeller wake flow into the ship wake. The vortices generated by the propeller and the rudder shown in Fig. 28(b) and (c) exhibit different characteristics, mostly due to two reasons. The first reason is that the drift angle in Fig. 28(c) is larger in magnitude than that in Fig. 28(b), causing stronger bilge vortices and cross flow. The second reason is the asymmetry caused by the rotation of the propeller. When the ruder deflects to starboard, the propeller-induced velocity increases the angle of attack in the lower part of the rudder, causing strong vorex shedding from the tip of the rudder as shown in Fig. 28(b). On the other hand, when the rudder turns to port side, the induced velocity increases the angle of attach in the upper part of the rudder, resulting in strip-like vortices shedding from the trailing edge of rudder. An animation of the vortical structures in real time at model scale is shown in Electronic Annex III. A cross section at $x/L_{pp} = 0.96$ is included

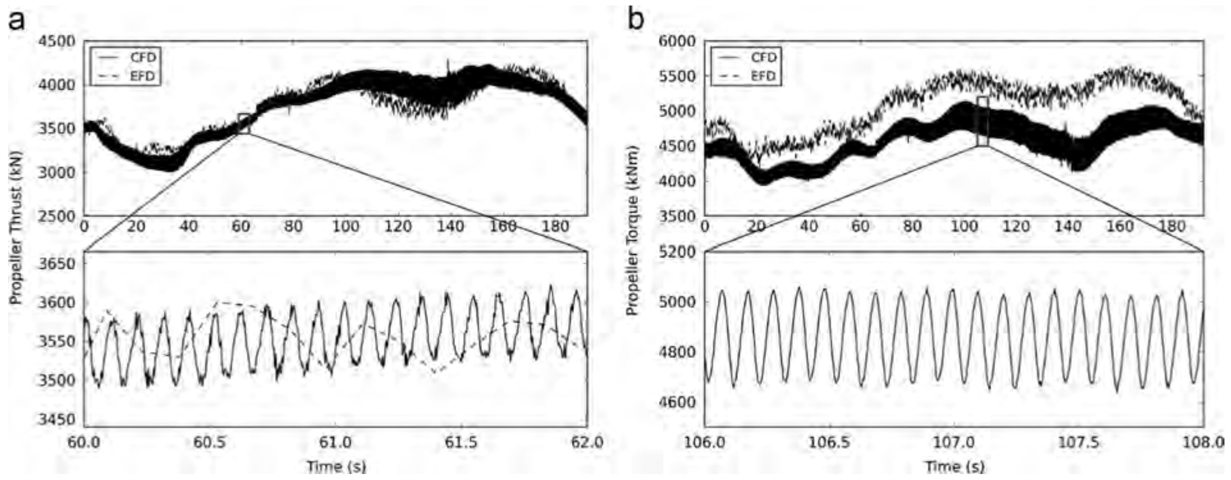


Fig. 26. Time histories of propeller thrust and moment for 10/10 standard Zig-zag maneuver (top: complete maneuver; bottom: close-up of local time history): (a) thrust and (b) torque.

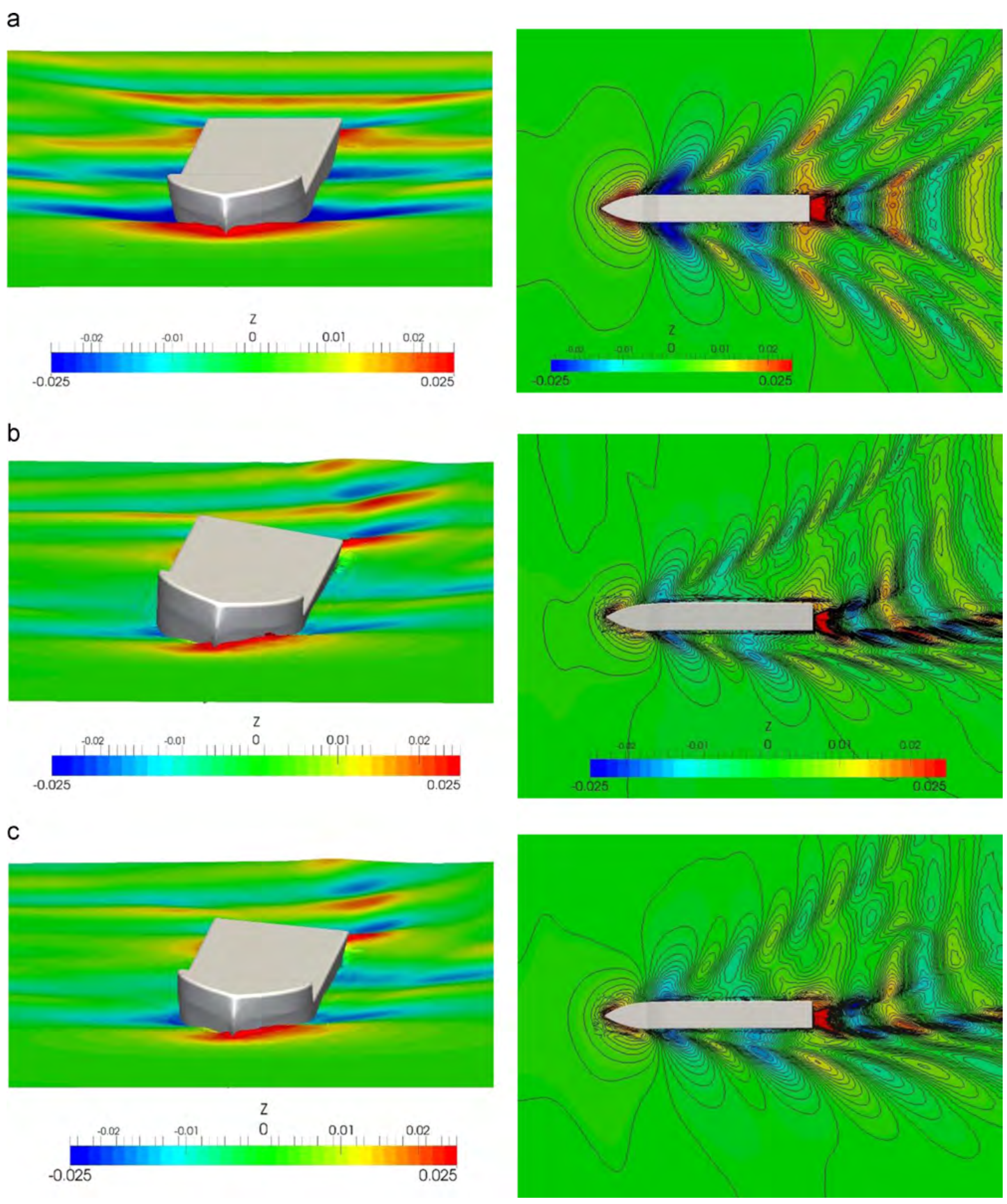


Fig. 27. Free surface waves for KCS zig-zag maneuvers: (a) self-propulsion, (b) 15/1 zig-zag maneuver and (c) 10/10 zig-zag maneuver.

to present the flow upstream of the propeller and rudder. The evolution of the propeller tip, root and hub vortices and the vortices shedding from the rudder root and tip can be clearly observed in the animation. As discussed above, every time the rudder is executed the vortical structures change dramatically.

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.oceaneng.2015.07.035.

5. Grid convergence discussion

Grid convergence studies are performed for the cases of open-water flow for the KP505 propeller and the towed condition for KCS. These cases are selected for the grid studies due to the simplicity of the overset grid layout (only two component grids for each case) and the fact that it would be too expensive to conduct a

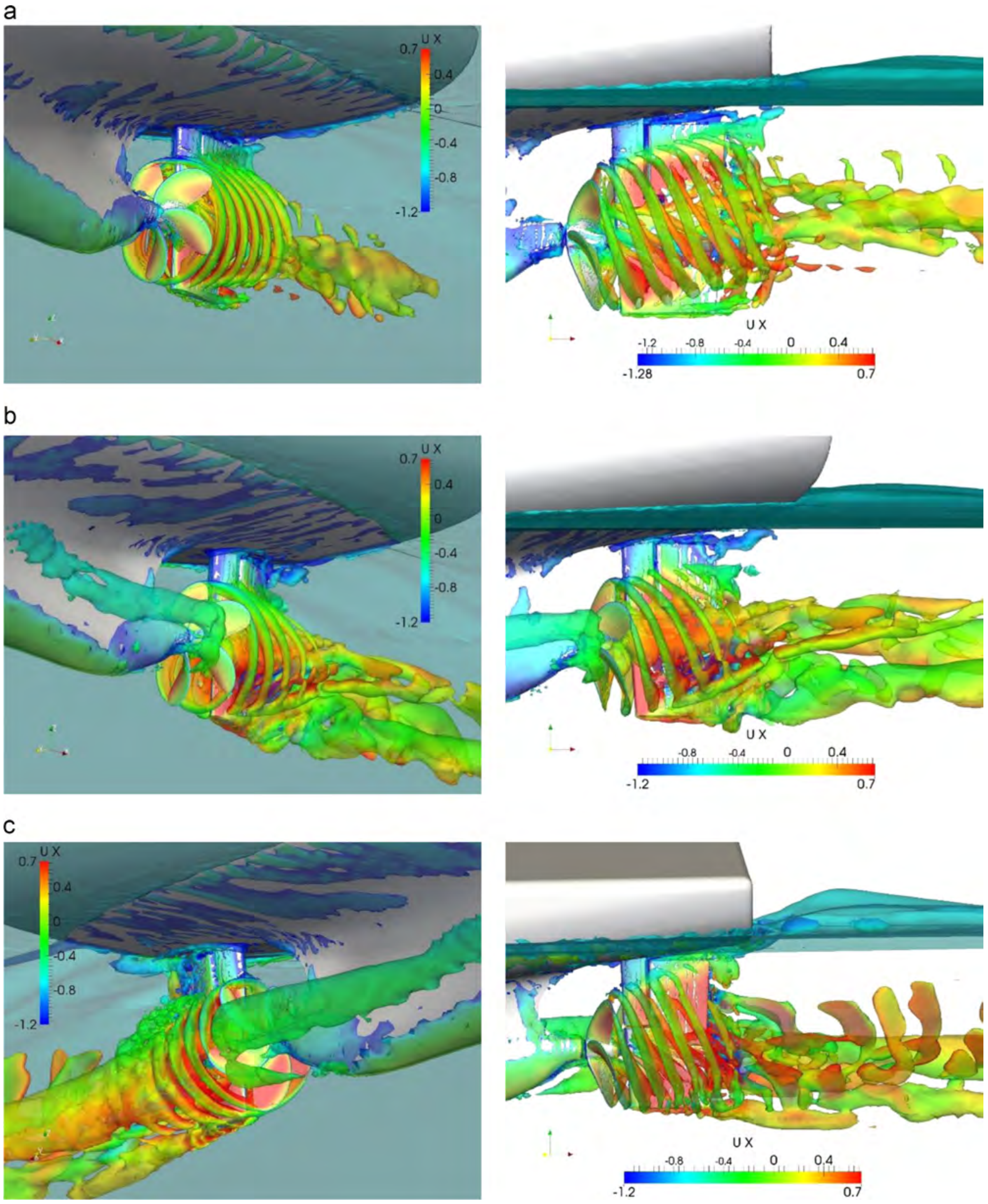


Fig. 28. Vortical structures represented by isosurfaces of Q visualizing vortices produced by propeller, rudder and hull interaction for the 10/10 zig-zag maneuver: (a) final solution of self-propulsion and initial condition for maneuver ($T=0$), (b) after a rudder execution from port to starboard ($T=45.7$ s) and (c) after a rudder execution from starboard to port side ($T=142.2$ s).

grid convergence study for self-propulsion and zig-zag maneuvers in terms of computational resources.

Notice that all grids used in this paper are unstructured grids generated with SnappyHexMesh. As opposed to structured grids, systematic refinement in three directions is difficult. To achieve consistent refinement in all directions, the following approach is applied. SnappyHexMesh requires a preliminary Cartesian grid and splits cells from it to get an unstructured mesh. By creating

Table 13
Results of grid convergence study at $J=0.7$.

Grid name	ID	Grid size (M)	K_T	Error (%)	$.K_Q.$	Error (%)	η	Error (%)
EFD			0.185		0.0311		0.665	
Fine	S_1	4.802	0.18487	−0.071	0.03075	−1.123	0.66977	0.717
Medium	S_2	1.611	0.18537	0.202	0.03072	−1.213	0.67221	1.084
Coarse	S_3	0.580	0.19009	2.752	0.03185	2.411	0.66492	−0.012
R_G			0.1059		−0.0265		−0.3347	
P_G			6.4804					
$U_{GC}(\%S_2)$			0.2379					
U_G $(\%S_2)$			0.5078		1.8392		0.5422	
Convergence type			Monotonic		Oscillatory		Oscillatory	

Table 14
Convergence study of towed condition of KCS at $Fr=0.26$.

Grid name	ID	Mesh size (M)	C_p (10^{-3})	C_v (10^{-3})	C_t (10^{-3})	Error (%)
EFD					3.55	
Fine	S_1	10.58	0.6611	2.865	3.526	−0.663
Medium	S_2	4.26	0.6684	2.859	3.528	−0.631
Coarse	S_3	1.68	0.6988	2.817	3.516	−0.959
R_G			0.2401	0.1429	−0.1667	
P_G			4.1180	5.6172		
$U_{GC}(\%S_2)$			0.7477	0.1750		
U_G $(\%S_2)$			1.8405	0.3850	0.1701	
Convergence type			Monotonic	Monotonic	Oscillatory	

systematically refined Cartesian grids with a specific refinement ratio and running SnappyHexMesh over these grids, the resulting unstructured grids are approximately (but not exactly) systematically refined as required for a grid study. In this study, a refinement ratio of $\sqrt{2}$ in each direction is selected.

The methodology discussed in Stern et al. (2001) and Wilson et al. (2004) were used to perform a verification study, with results shown in Tables 13 and 14. The convergence of the solution (R_G) is defined by the solutions (S) of the three grids:

$$R_G=\frac{S_2-S_1}{S_3-S_2} \tag{25}$$

where S_i ($i=1,2,3$) are the solutions of the Fine, Medium and Coarse grids, respectively. There are three possible convergence conditions:

$$\begin{cases} 0<R_G<1 & \text{Monotonicconvergence} \\ R_G<0 & \text{Oscillatoryconvergence} \\ R_G>1 & \text{Divergence} \end{cases} \tag{26}$$

The grid uncertainty (U_G) was estimated for each grid convergence study. Notice that the grid uncertainty cannot be estimated for divergent condition.

5.1. KP505 propeller

In this case, overset grids are used but no free surface is present, comprising a test for the overset implementation in single phase. The grid convergence study is conducted at $J=0.7$ with three sets of overset grids ranging from 0.58 million to 4.8 million points. Table 13 summarizes the results, including the grid uncertainty analysis. The unrealistically high number obtained for P_G (order of accuracy) may indicate that the grids are not in the asymptotic range, likely because they are not truly systematically refined. We, however, proceed and estimate grid uncertainty. U_{GC} and U_G are the corrected and uncorrected grid uncertainties respectively. From Table 13, the thrust coefficient (K_T) shows monotonic convergence with $R_G=0.1059$ while the torque coefficient (K_Q) and the efficiency (η) show oscillatory convergence with R_G of -0.0265 and -0.3347 respectively. The grid uncertainty (U_G) ranges from 0.5078% to 1.8392%. The averaged grid uncertainty is $U_G=1.396\%$. The grid used for the self-propulsion computations (Section 4.4) using this propeller was finer than the medium grid but coarser than the fine grid. It must be stressed that small changes in the grid design slightly change the results for computations with this propeller. This may be due to the sharp edges on the blade tip and trailing edge that are hard to capture well with SnappyHexMesh, resulting in slightly different geometries, even as refined grids are used since the grid generation based on Cartesian and octree refinements does not conform fully to the very small features of a propeller trailing edge. Other grid generation software may perform better on this regard.

5.2. Towed condition of KCS

The towed condition computation of KCS discussed in Section 4.2 adds the computation of the free surface to test the performance of overset grids. Three grids with refinement ratio of $\sqrt{2}$ were used, with the coarsest grid comprising 1.68 million points and the finest 10.58 million points. The results show good convergence, as summarized in Table 14. Again the order of accuracy obtained is higher than the approximately second order expected. Both C_v and C_p show monotonic convergence with R_G of 0.2401 and 0.1429 respectively. The total resistance coefficients (C_t) presents oscillatory convergence with $R_G=-0.1667$. The C_p shows the maximum grid uncertainty with $U_G=1.8405\%$. The grid uncertainty of C_t is only 0.1701%, suggesting that the grid density has limited effect on C_t in the selected range of grid size.

6. Conclusions and future work

This paper discusses the implementation of overset grids in the open source code OpenFOAM with respect to the naoe-FOAM-SJTU solver. The DCI, used to connect the multiple overset grids, is dynamically computed with Suggar. The implementation is fully parallelized to improve runtime efficiency. The DCI is decomposed based on the CFD domain decomposition before sent to each CFD processor. A lagged mode is implemented to run OpenFOAM and Suggar simultaneously, preventing the CFD code from waiting while Suggar computes the DCI. The implementation allows the use of grids generated by SnappyHexMesh directly, simplifying the task of mesh generation for complex geometries.

Computations of the towed bare hull KCS were performed as needed to obtain self-propulsion factors. Free-surface, resistance and nominal wake were compared against experimental data showing good results. Open-water computations for the KP505 propeller were performed using overset and non-overset grids, resulting in no noticeable discrepancies. The propeller open-water curves were obtained using the single run methodology, achieving good agreement with experimental data. Self-propulsion computations were carried out adding the KP505 propeller to the bare hull and using a PI controller to achieve the desired speed by changing the propeller RPS. The propeller RPS and ship speed converged after 6 s (model scale), with the final results for RPS and self-propulsion factors in close agreement with experiments and comparable with other results presented in the Computational Ship Hydrodynamics Workshop Gothenburg 2010. Simulations of zig-zag maneuvers for the HSVA KCS model fitted with a five-bladed propeller and a semi-balanced horn rudder were then conducted, with direct simulation of the rotating propeller and the moving rudder. 15/1 and 10/10 zig-zag maneuvers were performed. The results show good agreement with measurements, including ship motions, rudder angle and propeller forces, but the overshoot angle is underpredicted, possibly due to excessive rudder turning moment caused by inability to predict separation in the rudder. In addition, two grid convergence studies were performed for the KP505 propeller at $J=0.7$ and the bare hull KCS model in towed condition, achieving good convergence.

The main conclusion of the work presented in this paper is that the naoe-FOAM-SJTU solver can be effectively modified to handle overset grids and perform complex computations with moving objects. Of particular interest is the ability to perform ship hydrodynamics free model computations once overset grids and appropriate motion controllers are implemented.

Future work includes more validation for other ship hydrodynamics problems and inclusion of detached eddy simulation (DES) turbulence models.

Acknowledgments

The authors thank IIHR-Hydroscience and Engineering for hosting Dr. Zhirong Shen's visit to The University of Iowa and provide the research environment for the development of this work. The visit of Dr. Zhirong Shen to The University of Iowa was supported by China Scholarship Council. The work is also supported by the National Natural Science Foundation of China (Grant nos. 51379125, 51490675, 11432009, and 51411130131), National Key Basic Research Development Plan (973 Plan) Project of China (Grant no. 2013CB036103), Chang Jiang Scholars Program (Grant no. T2014099), Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning (Grant no. 2013022), Center for HPC at Shanghai Jiao Tong University and Lloyd's Register LRF Foundation.

References

Berberović, E., van Hinsberg, N., Jakirlić, S., Roisman, I., Tropea, C., 2009. Drop impact onto a liquid layer of finite thickness: dynamics of the cavity evolution. Phys. Rev. E 79 (3), 36306.
Boger, D., Paterson, E., Noack, R.W., 2010. FoamedOver: a dynamic overset grid implementation in OpenFOAM. In: Proceedings of the 10th Symposium on Overset Composite Grids and Solution Technology. NASA Ames Research Center, Moffet Field, CA, USA.
Broglia, R., Dubbioso, G., Durante, D., Mascio, A.D., 2013. Simulation of turning circle by CFD: analysis of different propeller models and their effect on manoeuvring prediction. Appl. Ocean Res. 39, 1–10.
Bugalski, T., Hoffmann, P., 2010. Numerical simulation of the interaction between ship hull and rotating propeller. In: Proceedings of the Gothenburg 2010 Workshop, Gothenburg, Sweden, pp. 441–446.
Carrica, P.M., Castro, A.M., Stern, F., 2010a. Self-propulsion computations using a speed controller and a discretized propeller with dynamic overset grids. J. Mar. Sci. Technol. 15 (4), 316–330.
Carrica, P.M., Fu, H., Stern, F., 2011. Computations of self-propulsion free to sink and trim and of motions in head waves of the KRISO Container Ship (KCS) model. Appl. Ocean Res. 33 (4), 309–320.
Carrica, P.M., Huang, J., Noack, R.W., Kaushik, D., Smith, B., Stern, F., 2010b. Large-scale DES computations of the forward speed diffraction and pitch and heave problems for a surface combatant. Comput. Fluids 39 (7), 1095–1111.
Carrica, P.M., Ismail, F., Hyman, M., Bhushan, S., Stern, F., 2013. Turn and zigzag maneuvers of a surface combatant using a URANS approach with dynamic overset grids. J. Mar. Sci. Technol. 18 (2), 166–181.
Carrica, P.M., Sadat-Hosseini, H., Stern, F., 2012. CFD analysis of broaching for a model surface combatant with explicit simulation of moving rudders and rotating propellers. Comput. Fluids 53, 117–132.
Carrica, P.M., Stern, F., 2008. DES simulations of KVLCC1 in turn and zigzag maneuvers with moving propeller and rudder. In: Proceedings of the SIMMAN 2008 — Workshop on Verification and Validation of Ship Manoeuvering Simulation Methods, Copenhagen, Denmark.
Carrica, P.M., Wilson, R.V., Noack, R.W., Stern, F., 2007. Ship motions using single-phase level set with dynamic overset grids. Comput. Fluids 36 (9), 1415–1433.
Castro, A.M., Carrica, P.M., Stern, F., 2011. Full scale self-propulsion computations using discretized propeller for the KRISO container ship KCS. Comput. Fluids 51 (1), 35–47.
Chen, H.C., Yu, K., 2009. CFD simulations of wave–current-body interactions including greenwater and wet deck slamming. Comput. Fluids 38 (5), 970–980.
Fossen, T.I., 1994. Guidance and Control of Ocean Vehicles. John Wiley & Sons Inc, New York.
Hino, T. (Ed.), 2005. Proceedings of CFD Workshop Tokyo 2005. National Maritime Research Institute.
ITTC, 2002. Testing and Extrapolation Methods Propulsion, Performance Propulsion Test. Recommended Procedures and Guidelines, No. 7.5-02-03-01.1.
Jasak, H., 1996. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. Imperial College, London, UK (Ph.D. thesis).
Jin, W., Gao, Q., Vassalos, D., 2010. The prediction of KCS resistance and self-propulsion by RANSE. In: Proceedings of the Gothenburg 2010 Workshop, Gothenburg, Sweden, pp. 615–620.
Larsson, L., Stern, F., Visonneau, M. (Eds.), 2014. Numerical Ship Hydrodynamics – An assessment of the Gothenburg 2010 Workshop. Springer, Netherlands.
Lee, J.H., Rhee, S.H., 2010. Flexible CFD Meshing Strategy for Prediction of Ship Resistance and Propulsion Performance. In: Proceedings of the Gothenburg 2010 Workshop, Gothenburg, Sweden, pp. 595–600.
Lübke, L.O., 2005. Numerical simulation of the flow around the propelled KCS. In: Proceedings of the CFD Workshop Tokyo 2005, Tokyo Japan, pp. 587–592.
Menter, F.R., 2009. Review of the shear-stress transport turbulence model experience from an industrial perspective. Int. J. Comput. Fluid Dyn. 23 (4), 305–316.
Mofidi, A., Carrica, P.M., 2014. Simulations of zigzag maneuvers for a container ship with direct moving rudder and propeller. Comput. Fluids 96, 191–203.
Noack, R.W., 2005a. DiRTlib: a library to add an overset capability to your flow solver. In: Proceedings of the 17th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, Toronto, Ontario, Canada.
Noack, R.W., 2005b. SUGGAR: a general capability for moving body overset grid assembly. In: Proceedings of the 17th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, Toronto, Ontario, Canada.
Noack, R.W., Boger, D.A., Kunz, R.F., Carrica, P.M., 2009. Suggar++: an improved general overset grid assembly capability. In: Proceedings of the 19th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, San Antonio, TX, USA.
Rusche, H., 2002. Computational Fluid Dynamics of Dispersed Two-phase Flows at High Phase Fractions. Imperial College, London, UK (Ph.D. thesis).
Shen, Z.R., Wan, D.C., 2013. RANS computations of added resistance and motions of a ship in head waves. Int. J. Offshore Polar Eng. 23 (4), 263–271.
Shen, Z.R., Ye, H.X., Wan, D.C., 2012. Motion response and added resistance of ship in head waves based on RANS simulations. Chin. J. Hydrodyn. 27 (6), 621–633 (in Chinese).
Steinwand, M., 2006. Manövrierversuche mit einem Modell des Kriso Container Ship (KCS), No. Bericht No. 3293. Schiffbau-Versuchsanstalt Potsdam GmbH, Potsdam, Germany (in German).
Stern, F., Wilson, R.V., Coleman, H.W., Paterson, E.G., 2001. Comprehensive approach to verification and validation of CFD simulations—Part 1: methodology and procedures. J. Fluids Eng. 123 (4), 793–802.
Wilson, R., Shao, J., Stern, F., 2004. Discussion: criticisms of the "correction factor" verification method. J. Fluids Eng. 126 (4), 704–706.
Wu, Q., Yu, H., Wang, J.B., Cai, R.Q., Chen, X.L., 2010. Prediction of ship resistance and propulsion performance using multi-block structural grid. In: Proceedings of the Gothenburg 2010 Workshop, Gothenburg, Sweden, pp. 483–488.
Xing, T., Carrica, P.M., Stern, F., 2008. Computational towing tank procedures for single run curves of resistance and propulsion. J. Fluids Eng. 130 (10), 101102.
Zhang, Z., Wu, C., Huang, S., Zhao, F., Weng, Z., 2010. Numerical simulation of free surface flow of KCS and the interaction between propeller and hull. In: Gothenburg 2010 Workshop, Gothenburg, Sweden, pp. 435–440.