Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Applied Mathematical Modelling 29 (2005) 1196–1210
APPLIED MATHEMATICAL MODELLING
www.elsevier.com/locate/apm

Computation of flow-induced motion of floating bodies

I. Hadžić [a], J. Hennig [b], M. Perić [a], Y. Xing-Kaeding [c,*]

[a] *CD-adapco Group, Dürrenhofstraße 4, D-90402 Nürnberg, Germany*
[b] *Institute of Naval Architecture and Ocean Engineering, Department of Transportation and Applied Mechanics, Berlin University of Technology, D-10587 Berlin, Germany*
[c] *Fluid Dynamics and Ship Theory Section, Technical University of Hamburg-Harburg, Lämmersieth 90, D-22305 Hamburg, Germany*

Received 1 March 2003; received in revised form 1 January 2005; accepted 1 February 2005
Available online 14 April 2005

Abstract

A computational procedure for the prediction of motion of rigid bodies floating in viscous fluids and subjected to currents and waves is presented. The procedure is based on a coupled iterative solution of equations of motion of a rigid body with up to six degrees of freedom and the Reynolds-averaged Navier–Stokes equations describing the two- or three-dimensional fluid flow. The fluid flow is predicted using a commercial CFD package which can use moving grids made of arbitrary polyhedral cells and allows sliding interfaces between fixed and moving grid blocks. The computation of body motion is coupled to the CFD code via user-coding interfaces. The method is used to compute the 2D motion of floating bodies subjected to large waves and the results are compared to available experimental data, showing favorable agreement.

Keywords: Viscous flow; Floating body; Free surface deformation

* Corresponding author. Fax: +49 40 42804 5589.
E-mail addresses: milovan.peric@de.cd-adapco.com (M. Perić), yan.xing@tu-harburg.de (Y. Xing-Kaeding).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2005.02.014

1. Introduction

Computational fluid dynamics has been mainly used in shipbuilding to predict flows around bodies moving with a prescribed velocity and vertical position, i.e. with the forces acting on the body not influencing its motion. However, the motion of a floating body is a direct consequence of the flow-induced forces acting on it, while at the same time these forces are a function of the body movement itself. Therefore, the prediction of flow-induced body motion is a challenging task and requires a coupled solution of both fluid flow and body motion. It is relevant for many applications in ship and ocean engineering (sea-keeping, slamming, maneuvering etc.).

Ship hydrodynamics computations based on solving the Reynolds-averaged Navier–Stokes equations (RANSE) were initiated in the 1980s, and since then a number of research groups have developed methods for solving viscous flow problems. In the last few years, methods for solving RANSE together with the free-surface deformation have been presented and shown great potential in ship design and other practical applications. However, ship motions and associated loads on a ship hull have so far been mostly predicted using methods based on the potential theory (see Ref. [1] for a review of some of the most widely used methods.). For several practically important cases like ship motions in large-amplitude waves, ship response under impact wave loads (slamming), or ship capsizing, large errors can be introduced by the potential theory assumptions. The need for a numerical tool that can predict the motions and loads in large waves, taking into account viscous effects, turbulence, flow separation and wave-breaking phenomena, is thus obvious.

The methods that compute turbulent flow and allow arbitrary free-surface deformation and body motion are less numerous; however, this is an area of active research, as can be concluded from topics of presentations at the annual international workshops on Water Waves and Floating Bodies (20th to be held in 2005) and the Numerical Towing Tank Symposium (8th to be held in 2005). In most cases, special application areas and specialized methods for a particular purpose are presented, see e.g. Refs. [2,3].

A combination of methods solving RANSE in the near-body region and assuming potential flow elsewhere have been developed to overcome some of the problems of methods based on the potential theory while keeping the computing cost lower than when full RANSE are used in the whole domain (see Ref. [4]). For moderate body motions, solvers based on full RANSE with free-surface modeling using the volume-of-fluid approach have been used, where the grid is attached to the moving body (see Ref. [5]). Another class of methods uses the so-called ''immersed boundary approach'' (see Ref. [6]) or ''cut-cell approach'' (see Ref. [7]). In these methods the numerical grid is not fitted to the body surface, which makes the computation of turbulent shear stress at walls of slender bodies—where shear forces may be dominant—difficult.

This paper reports some results obtained in a research project aimed at developing and validating a computational technique for the coupled analysis of viscous flow and flow-induced body motion in large irregular waves. In the present study the fluid flow is predicted using the commercial CFD code Comet [8]. It is based on the finite volume method and can accommodate any type of grid, thus being applicable to complex geometry problems. SIMPLE algorithm [9] is used to couple the pressure field to the velocity field. HRIC (high-resolution interface capturing) scheme [10] is used to simulate the free-surface effects and to achieve the sharpness of the interface between water and air while allowing wave breaking.

The body motion is predicted by solving iteratively the equations of its motion while coupling them to Comet via the user-coding interfaces. A fully-implicit predictor–corrector procedure has been employed, taking advantage of the iterative nature of the fluid-flow solver. The adaptation of a body-fitted mesh to body motion is accomplished using a combination of mesh motion, mesh deformation, and sliding mesh interfaces, with the option of extension by using overlapping meshes. The aim is to allow arbitrary motion of multiple bodies.

The simulation of body motion can be conducted in two ways: within a body-fixed coordinate system, or with reference to the inertial coordinate system. In this study, the latter technique is used. The following sections describe the governing equations, boundary conditions, numerical solution method, and the results of test applications.

2. Mathematical model

2.1. Governing equations of fluid motion

The finite volume method for incompressible viscous flows is described in detail in Ferziger and Perić [11]. Here we describe it only briefly. The starting point are the conservation equations for mass, momentum, and scalar quantities (e.g. turbulence quantities like kinetic energy or dissipation rate, or chemical species) in their integral form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho \,\mathrm{d}V + \int_S \rho(\mathbf{v}-\mathbf{v}_\mathrm{b})\cdot\mathbf{n}\,\mathrm{d}S = 0, \tag{1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\mathbf{v}\,\mathrm{d}V + \int_S \rho\mathbf{v}(\mathbf{v}-\mathbf{v}_\mathrm{b})\cdot\mathbf{n}\,\mathrm{d}S = \int_S (\mathbf{T}-p\mathbf{I})\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho\mathbf{b}\,\mathrm{d}V, \tag{2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\phi\,\mathrm{d}V + \int_S \rho\phi(\mathbf{v}-\mathbf{v}_\mathrm{b})\cdot\mathbf{n}\,\mathrm{d}S = \int_S \Gamma\nabla\phi\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho b_\phi\,\mathrm{d}V. \tag{3}$$

In these equations, ρ is the fluid density, V is the control volume (CV) bounded by a closed surface S, $\mathbf{v}$ is the fluid velocity vector, $\mathbf{v}_\mathrm{b}$ is the velocity of the CV surface, t is time, Γ is the diffusion coefficient and b_ϕ the volumetric source of the conserved scalar quantity ϕ, p is the pressure, $\mathbf{b}$ is the body force, $\mathbf{n}$ is the unit vector normal to S and directed outwards, $\mathbf{I}$ is the unit tensor, and $\mathbf{T}$ is the viscous part of the stress tensor defined (for Newtonian fluids considered here) as

$$\mathbf{T} = \mu\Big(\nabla\mathbf{v} + (\nabla\mathbf{v})^\mathrm{T}\Big), \tag{4}$$

with μ being the dynamic viscosity of the fluid. Turbulence effects are modeled using a turbulence model of eddy-viscosity type (k–ϵ), which leads to replacing μ in the above equation by a sum of μ and μ_t. The latter is the so-called turbulent or eddy viscosity, which is computed as a local function of k and ϵ. Details on turbulence models can be found in Refs. [12,13].

When the grid is moving, the so-called space conservation law (SCL) has to be satisfied, which is expressed by the following relation between the rate of change of CV-volume and its surface velocity:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \mathrm{d}V - \int_S \mathbf{v}_\mathrm{b}\cdot\mathbf{n}\,\mathrm{d}S = 0. \tag{5}$$

The interface-capturing method and the high-resolution interface capturing scheme [10] have been used to simulate the free-surface effects. In addition to the conservation equations for mass and momentum, a transport equation for void fraction of the liquid phase c has been introduced:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V c\,\mathrm{d}V + \int_S c(\mathbf{v}-\mathbf{v}_\mathrm{b})\cdot\mathbf{n}\,\mathrm{d}S = 0. \tag{6}$$

The grid extends to both liquid and gas phase; the void fraction c is set equal to 1 for CVs filled by liquid and to 0 for CVs filled by gas. Therefore, both fluids are treated as a single effective fluid whose properties vary in space according to the volume fraction of each phase, i.e.

$$\rho = \rho_\mathrm{l}c + \rho_\mathrm{g}(1-c), \quad \mu = \mu_\mathrm{l}c + \mu_\mathrm{g}(1-c), \tag{7}$$

where subscripts "l" and "g" denote the two fluids (liquid and gas).

The effects of surface tension at the interface between two fluids are treated through a body force as a function of the volume fraction c, which is achieved by introducing the continuum surface force (CSF) model [14]. The CSF model uses the smoothed field of c to define a unit vector normal to the interface. The surface tension force can then be expressed as:

$$F_\mathrm{st} = \int_V \sigma\kappa\nabla c\,\mathrm{d}V, \quad \kappa = -\nabla\cdot\left(\frac{\nabla c}{|\nabla c|}\right), \tag{8}$$

where σ is the surface tension coefficient and κ is the curvature of the interface. The surface tension force is only of importance on the small scale, when the curvature is substantial and well resolved by the numerical grid.

2.2. Governing equations of body motion

The body motion is analyzed in a space-fixed Cartesian coordinate system, called *global coordinate system*. The equations of motion of a rigid body in this coordinate system are given as follows:

$$\frac{\mathrm{d}(m\mathbf{v}_\mathrm{C})}{\mathrm{d}t} = \mathbf{f}, \tag{9}$$

$$\frac{\mathrm{d}(\mathbf{M}_\mathrm{C}\cdot\boldsymbol{\omega}_\mathrm{C})}{\mathrm{d}t} = \mathbf{m}_\mathrm{C}. \tag{10}$$

In the above equations, m represents the mass of the body, $\mathbf{v}_\mathrm{C}$ is the velocity vector of the center of mass of the body, $\mathbf{M}_\mathrm{C}$ is the tensor of the moments of inertia of the body, $\boldsymbol{\omega}_\mathrm{C}$ is the angular velocity vector of the body, $\mathbf{f}$ is the resultant vector of forces acting on the body, and $\mathbf{m}_\mathrm{C}$ represents moments of forces acting on the body with respect to its center of mass.

The resultant force $\mathbf{f}$ is obtained by integrating the pressure and shear stress over the body surface, cf. Eq. (2)

$$\mathbf{f} = \int_S (\mathbf{T}-p\mathbf{I})\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho\mathbf{b}\,\mathrm{d}V. \tag{11}$$

The only body force considered here is the gravity, so the second integral on the right-hand side of the above equation reduces to $m\mathbf{g}$, where $\mathbf{g}$ is the gravity acceleration vector.

Body forces do not contribute to the moment around the center of mass, so only the moments of pressure and shear forces acting on the body surface need to be integrated:

$$\mathbf{m}_C = \int_S (\mathbf{r} - \mathbf{r}_C) \times (\mathbf{T} - p\mathbf{I}) \cdot \mathbf{n}\, dS, \tag{12}$$

where $\mathbf{r}$ represents the position vector of a point on the body surface expressed in global coordinates while $\mathbf{r}_C$ is the position vector of the center of mass of the body.

When the body rotates, its moments of inertia with respect to the global coordinate system are also changing; therefore, they have to be updated each time the body position changes. The following expression for computing components of the moment of inertia tensor is used:

$$\mathbf{M}_C = \int_V \rho[(\mathbf{r} - \mathbf{r}_C) \cdot (\mathbf{r} - \mathbf{r}_C)\mathbf{I} - (\mathbf{r} - \mathbf{r}_C)(\mathbf{r} - \mathbf{r}_C)]\, dV. \tag{13}$$

In order to compute the moments of inertia of an arbitrary body, it has to be subdivided into a finite number of control volumes, akin to the discretization of the flow domain, as will be described below.

An alternative approach is to compute moments of inertia for a body-fixed (local) coordinate system once and use coordinate transformation to obtain moments at times when the local coordinate system does not coincide with the global one. Both approaches have been tested here and they gave the same results within the level of discretization errors.

3. Numerical methods

3.1. Flow calculation

The solution domain is subdivided into a finite number of non-overlapping control volumes (CVs) by an unstructured grid; in the center of each control volume lies the computational point at which the known quantities are specified and the unknown variables are to be computed. Eqs. (1)–(3) are applied to each CV and then discretized, leading to one algebraic equation per CV in which variables from immediate neighbors also feature. All integrals are approximated using the midpoint rule, i.e. the function to be integrated is evaluated at the center of the integration domain and multiplied by the area, volume, or time interval over which the integration takes place. In order to evaluate the function at the center of the integration domain, one needs to introduce further approximations: interpolation and differentiation. In space, linear interpolation is used, while in time either linear or quadratic shape functions are used. The diffusive fluxes require that the derivatives in the direction normal to CV faces be computed at each cell-face center; these are obtained from linear shape functions with the help of least-squares method or Gauss-theorem. The integration in time is fully implicit (either implicit Euler, which is a first-order method, or three-time-level scheme, which is of second order). The spatial integration is of second order. In order to keep the computational molecule limited to cell-center node and centers of nearest neighbor cells, a deferred-correction approach is used: low-order approximations, which use only

nearest neighbors, are used to construct the coefficient matrix, and the difference between the desired approximation and the low-order one is computed explicitly from the values obtained in the previous iteration and added to the source term on the right-hand side of the equation. More details on individual steps in the discretization procedure, in particular with respect to accounting for grid motion and space-conservation law, can be found in Refs. [10,15].

The algebraic equation obtained at the end in each CV has the following form:

$$a_{P_0}\phi_{P_0} + \sum_{j=1}^{n_j} a_{P_j}\phi_{P_j} = b_{P_0}, \tag{14}$$

where n_j is the number of internal cell faces surrounding cell P_0 and the right-hand side b contains source terms and contributions from boundary faces (convective and diffusive fluxes), which are for the sake of computational efficiency treated explicitly using deferred correction approach.

For the solution domain as a whole, a matrix equation results,

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{b}, \tag{15}$$

where $\mathbf{A}$ is a square, $N \times N$ sparse coefficient matrix, N being the number of control volumes and the number of non-zero elements at any row being equal to n_j, the number of neighbor cells.

In order to calculate the pressure field and to couple it to the velocity field, a pressure-correction method of SIMPLE-type for collocated grids is used [9]. Turbulence is taken into account by solving two additional transport equations for turbulent kinetic energy k and its dissipation rate ε and adding an eddy viscosity (computed with help of these two quantities) to the molecular viscosity.

The Eq. (14) is obtained for each variable (velocity components, pressure-correction, turbulent kinetic energy, etc.). Due to the non-linearity of the underlying equations, the solution of the linearized system of algebraic equations (15) is sought by iterative methods from the conjugate-gradient family (*inner iterations*). The segregated algorithm is adopted to achieve the solution of the non-linear and coupled equation system.

The coupled computation of flow and body motion consists of the following steps:

(1) Provide the initial values for the dependent variables (at the time t_0).
(2) Advance the time by Δt and, if the grid moves due to the prescribed motion of the boundary (like at wave-maker), determine the location of CV vertices at time $t_n + \Delta t$.
(3) Assemble and solve by an iterative solver the linearized algebraic equations for the velocity components in turn, employing the currently available other dependent variables.
(4) Assemble and solve the algebraic equations for the pressure correction and correct mass fluxes, velocity components, and pressure.
(5) Assemble and solve the algebraic equations for volume fraction c and use the calculated values to update the properties of the effective fluid, such as density and viscosity etc.
(6) Assemble and solve the algebraic equations for turbulent kinetic energy k and its dissipation rate ε and obtain turbulent diffusion coefficients.
(7) Integrate the pressure and shear forces over the body surface, solve the equations of motion for the floating body (velocity and displacement) using the predictor–corrector method described below, and adapt the grid to the new position of the body.
(8) Calculate the current estimate of volumes δV_j swept by each CV face over the last time step. This completes one *outer iteration*.

(9) Return to Step 3 and repeat until the sum of the absolute residuals for all equations has fallen by a prescribed number of orders of magnitude.
(10) Return to step 2 and repeat until the prescribed number of time steps is completed.

3.2. Computation of body motion

The forces and moments acting on a floating body are obtained from the flow as described before; however, the flow is influenced by body motion and both problems have to be considered simultaneously. For the prediction of body motion, a predictor–corrector method which can be easily coupled with the iterative procedure for flow prediction (SIMPLE-algorithm) has been used here.

For a body of constant mass, Eq. (9) can be written in the following form:

$$\frac{\mathrm{d}\mathbf{v}_C}{\mathrm{d}t} = \frac{\mathbf{f}}{m}. \tag{16}$$

This equation is integrated in time using trapezoid rule so that an estimate of the solution at time t_n is computed as

$$\mathbf{v}_{C,n}^m = \mathbf{v}_{C,n-1} + \frac{\Delta t}{m}\frac{\mathbf{f}_{n-1} + \mathbf{f}_n^m}{2}. \tag{17}$$

The superscript m denotes here the mth outer iteration at the new time level.

Once a new estimate of the velocity at the new time level is obtained, one can compute an estimate of the new body center position:

$$\mathbf{r}_{C,n}^m = \mathbf{r}_{C,n-1} + \frac{\mathbf{v}_{C,n}^m + \mathbf{v}_{C,n-1}}{2}\Delta t. \tag{18}$$

The same procedure is applied to integrate Eq. (10). After each outer iteration within one time step, these equations are solved to produce an estimate of the new body location; the right-hand sides of Eqs. (17) and (18) are computed using values from the previous outer iteration where the new solution is required. If the iterations are converging, the difference between two consecutive estimates becomes smaller and smaller and the process can be stopped when an appropriate limit is reached. The time can then be advanced by Δt and the coupled iteration on fluid flow and body motion started again. Usually, an order of 10 outer iterations per time step is needed to reach a converged solution; this number reduces when the time step is reduced and vice versa.

Under-relaxation has been used for both fluid flow and body motion. In the case of body motion, the under-relaxation corresponds to incorporating an added-mass term in the equation of motion, which is especially important when light bodies are subjected to high acceleration. For example, the under-relaxed form of Eq. (17) reads:

$$\frac{\mathbf{v}_{C,n}^m}{\alpha} = \mathbf{v}_{C,n-1} + \frac{\Delta t}{m}\frac{\mathbf{f}_{n-1} + \mathbf{f}_n^m}{2} + \frac{1-\alpha}{\alpha}\mathbf{v}_{C,n}^{m-1}. \tag{19}$$

When outer iterations converge, the difference between mth and $(m-1)$th iteration becomes negligible.

Other methods for integrating the equations of body motion are possible; the above approach has been found appropriate and easy to implement. It is of the same order as the method used to compute the fluid flow and fits well within the overall iteration scheme.

4. Solution domain, boundary conditions, and grid systems

The solution domain includes both water and air; here only 2D-cases will be studied, but the method is implemented in a 3D code and all six degrees of freedom can be considered. The outer boundary of the solution domain is in all cases a rectangle: bottom boundary is a no-slip wall, top boundary is either a slip wall or the atmospheric pressure is prescribed there, right-hand boundary is either a no-slip wall or the hydrostatic pressure is specified there, and the left-hand side represents either an inlet with prescribed velocities or a moving no-slip wall if the wave-maker is simulated by a moving wall.

The inlet velocities can, for some wave types, be derived from wave theory [16]. Suppose a Cartesian coordinate system (x, y, z) is adapted, with the plane $y = 0$ coinciding with undisturbed free surface and with y-axis positive upwards. The elevation of any point in the free surface can be described by the function

$$y = \eta(x, z, t). \tag{20}$$

For regular plane progressive waves, the two-dimensional description is given by [16]:

$$\eta(x, t) = A\cos(kx - \omega t + \epsilon), \tag{21}$$

where the positive x-axis has been chosen to coincide with the direction of wave propagation, A is the wave amplitude, k is the wave number, t is time, ω is the radian frequency and ϵ is an arbitrary phase angle which can be set equal to zero by a suitable choice of the origin $x = 0$, which means that the inlet boundary does not necessarily coincide with the origin $x = 0$.

The velocity components (u, v) at inlet are in this case:

$$\begin{aligned} u_{\mathrm{i}} &= \omega A \mathrm{e}^{ky}\cos(kx - \omega t), \\ v_{\mathrm{i}} &= \omega A \mathrm{e}^{ky}\sin(kx - \omega t). \end{aligned} \tag{22}$$

For shallow-water waves, the fluid velocity components can be specified as follows [16]:

$$\begin{aligned} u_{\mathrm{i}} &= \frac{gAk}{\omega}\frac{\cosh(k(y + h))}{\cosh(kh)}\cos(kx - \omega t), \\ v_{\mathrm{i}} &= \frac{gAk}{\omega}\frac{\sinh(k(y + h))}{\sinh(kh)}\sin(kx - \omega t). \end{aligned} \tag{23}$$

For non-linear waves with large amplitude, the fluid velocity may be derived from measurements or by adapting other profiles to reproduce the waves of specified crest–trough height.

If a wave-maker is used to generate waves, one can prescribe the motion of the wall according to the experimental set-up; this approach provides the most realistic match between experimental

and numerical boundary conditions and is especially necessary if irregular waves are generated, see below. In the case of a small-amplitude motion of the wave-maker wall, one can keep the solution domain boundary fixed and instead of its motion prescribe an inlet (or outlet) velocity of the fluid. This simplifies the problem as the grid remains fixed, but cannot be used for large-amplitude motions of wave makers.

Body surface is the inner solution domain boundary, where boundary conditions of a moving no-slip wall are applied. The grid has to move with the body in its vicinity in order to remain adapted to the body surface.

There are five possibilities for moving-grid strategies:

(1) Moving the entire grid with the body without deformation. In this case special care is needed to specify the boundary conditions at outer boundaries, since the free-surface location there changes with the grid motion. This strategy is only applicable when a single body in an infinite domain is considered. This approach has been used in Ref. [5].
(2) Moving a part of the grid around the body with it, keeping distant parts fixed, and deforming the grid between the two regions, while keeping its topology the same. In this case the outer boundary of the solution domain is not affected by the body motion, which makes the implementation of boundary conditions easier (e.g. at wave-maker, inlet boundary, outlet boundary etc.). However, the grid in the adapted region may deform too much and attain unacceptable properties (skewness, non-orthogonality etc.), so this approach is only applicable to moderate body motions.
(3) Moving a part of the grid around the body with it, keeping the distant part fixed, and re-gridding the region between the two grid blocks. This approach is applicable to arbitrary body motions, but it requires (i) a reliable, fast and fully automatic grid generation method, since it is not possible to visually monitor the grid quality after each time step and (ii) a special interpolation of old solutions to the new grid locations, which may introduce additional errors in the solution.
(4) Using moving grid blocks and sliding interfaces. In this case the grid is subdivided into several blocks. The block around the body moves with it without deformation; to account for rotation, the block boundary must be either cylindrical (one rotational degree of freedom) or spherical (three rotational degrees of freedom). The surrounding block translates with the body without deformation, but does not rotate. Finally, outer grid blocks expand or contract linearly to adapt to the position of inner block boundaries (which are usually planes aligned with the global coordinate system). This approach preserves the grid quality, but the grid generation requires much effort, therefore application to three-dimensional cases is challenging.
(4) Using overlapping grids. In this case a body-fitted grid is used in the neighborhood of the body and is moving with it without deformation. The rest of the solution domain is covered by a fixed grid (parts of it may move if the outer boundaries move, e.g. at a wave-maker wall) and the two grids overlap in the vicinity of the body. The body can then move without restriction, and one can also consider several bodies moving relative to each other. This approach offers the greatest flexibility, but the coupling of grids in the overlapping region and especially maintaining the conservativeness is a non-trivial issue when arbitrary unstructured grids are used.

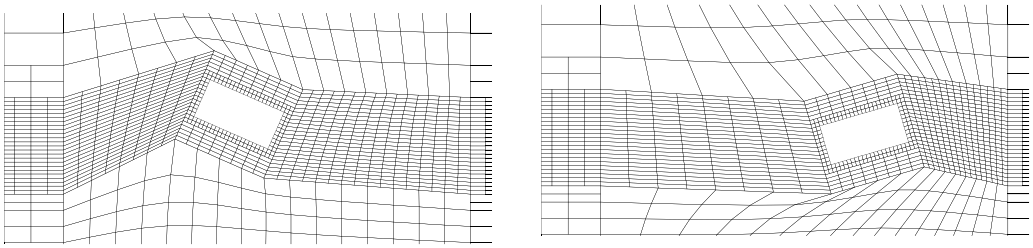

Fig. 1. Coarse grid around a floating body at two time instants, showing grid distortion in the adapted region.

In this paper, the second and the fourth moving-grid strategies have been used. For large angles of rotation, the sliding interface had to be used to avoid excessive grid deformation. For large translation movements, expansion or compression of some grid blocks is possible. For small body movements, the grid is adapted and smoothed around the body. These strategies are applicable to a variety of problems, while special attention has to be paid to grid quality. For some problems, like motion of several bodies relative to each other, overlapping grids would have to be used; the extension of the present approach in this direction is under way and results from it will be reported in the near future.

An example of the second moving-grid strategy is shown in Fig. 1, which shows the grid around a floating body at two time instants. The advantage of this approach is that the grid retains good quality near the body; highly deformed cells are usually at some distance away from it. However, if the body moves over large distance or rotates by more than 30°, some CVs may be so deformed that the numerical method would no longer converge.

The combination of rotating, translating, and expanding/contracting grid blocks as an example of the fourth moving-grid strategy is shown in Fig. 2. The body is embedded in a grid block whose outer surface is cylindrical; when the body rotates, the grid slides along this cylindrical interface while the grid points from the outside grid blocks remain fixed at the cylinder surface. Since the neighborhood relations at this sliding interface change with time, the connectivity has to be updated after each time step. Only cells far away from body change their shape through contraction or expansion; near-body blocks only translate or rotate, so the grid remains the same all the time.

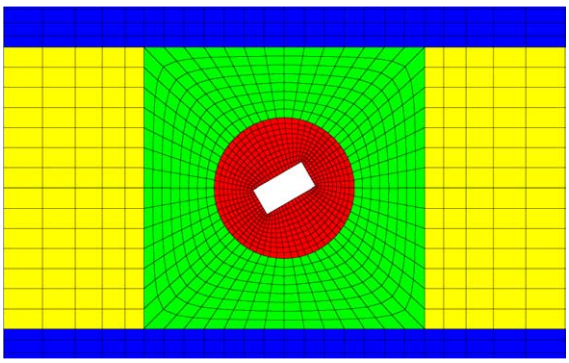

Fig. 2. Coarse grid around a floating body with a sliding cylindrical interface.

5. Examples of application

In order to verify the procedure and estimate the numerical accuracy, a number of simple test cases involving translation or rotation only and small-amplitude waves has been analyzed first, see Ref. [17]. By systematically varying the grid spacing and time-step size, it has been verified that the iterative procedure is converging within each time step and that the results are converging towards grid- and time-step-independent solutions. Here the results of two-dimensional simulations with three degrees of freedom are presented and in one case compared with experimental data.

We studied in detail the motion of a floating body subjected to regular and irregular wave packets. The results were compared with experimental data obtained in a small towing tank of Berlin University of Technology. The body was a rectangular hexahedron 10 cm wide, 5 cm high and 29 cm long, with density relative to water being 0.68. Only the results of one test case are presented here.

The body was located at $x = 2.11$ m away from the wave-maker, whose motion was computer-controlled to produce a wave packet with a high-amplitude concentrated wave at the original location of the body. The free-floating body motion was recorded by a digital camera, saving 60 single frames per second with a resolution of 640×480 pixel. From these images and the assumed time span between two frames of 1/60 s, the motion of body center and its rotation have been reconstructed.

The numerical towing tank was 40 cm deep and with 2 m of air above water. No-slip conditions have been employed at the body surfaces and tank walls. The vertical boundary on the left-hand side simulated a flapping wave-maker by moving the wall according to experiment. The effective numerical towing tank has been set 3 m long with another 5 m of coarse grid acting as a numerical beach to prevent wave-reflection. For the sake of numerical efficiency, local cell-wise grid refinement in the vicinity of the free surface and around the body has been used, while the grid was coarse in distant regions, especially in the air whose motion is of no relevance here. The computational grid contains 5891 CVs and the total computational time is about 3 h on a PC. The time step has been kept small to make temporal discretization errors smaller than spatial discretization errors. Due to grid moving, 10–15 iterations are needed for each time step while normally less than 10 iterations are required if the grid is stationary.

Fig. 3 shows the comparison of computed and measured water elevation approximately half-way between the body and the wave-maker, obtained on the grid with 5891 CVs, which had been found in a grid-refinement study to provide solutions with discretization errors of the order of 1%. The agreement can be considered as satisfactory. The fact that the numerical solution over-predicts the wave amplitude suggests that the amplitudes of the actual motion of the computer-controlled wave-maker in the experiment were slightly smaller than those provided by the control software and used as input in the numerical simulation since the numerical method normally under-predicts amplitudes.

The motion of the floating body and the traveling wave can be observed in Fig. 4, which shows the wave-maker position, the free-surface shape, and the body position at three time instants.

The comparison of measured and computed body motion under the wave-packet load is shown in Fig. 5. The experimental data has been deduced from digital images with the assumed time spacing of 1/60 s; the uncertainty is around 5% in vertical motion and 9% for rotational motion, while time steps between images appear to be larger than assumed. This leads to a phase shift

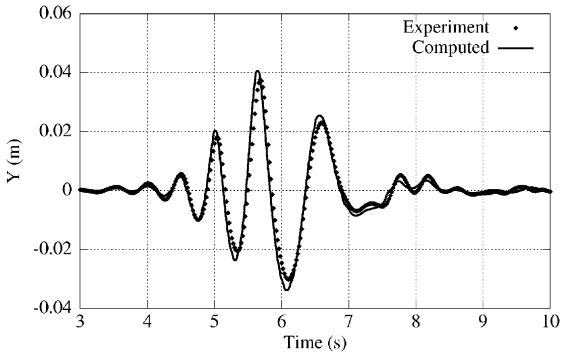

Fig. 3. Comparison of computed and measured water elevation 1.16 m away from wave-maker.

Fig. 4. Free-surface shape and the position of the wave-maker and the floating body at three time instants.

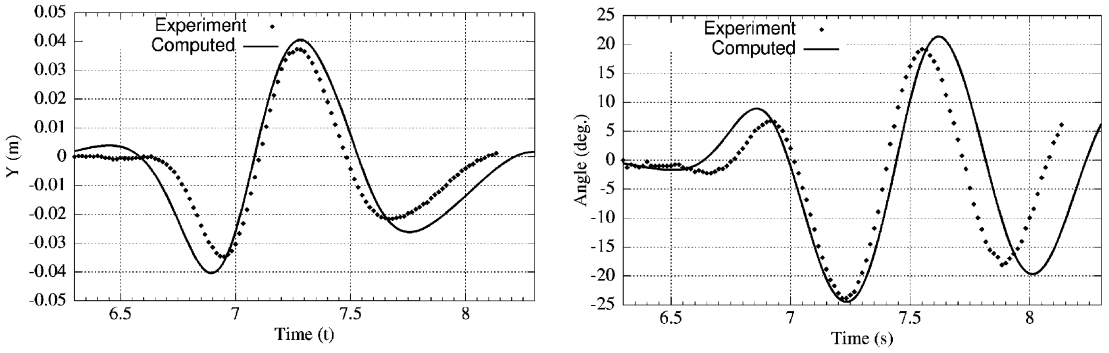

Fig. 5. Comparisons of computed and measured body motion: vertical displacement (left), and rotation (right).

between the predicted body motion by simulation and by experiment, which does not exist in the prediction of the water elevation at $x = 1.16$ m as shown in Fig. 3. The agreement between simulation and experiment is satisfactory for both degrees of freedom, taken into account the

uncertainty in reconstruction of body motion from digital images. Note that the body is 5 cm high and the amplitude of its vertical motion is 4 cm on either side of the neutral position. The angular motion is even more severe: maximum rotations of about $-25°$ and $+20°$ have been observed in both experiment and simulation.

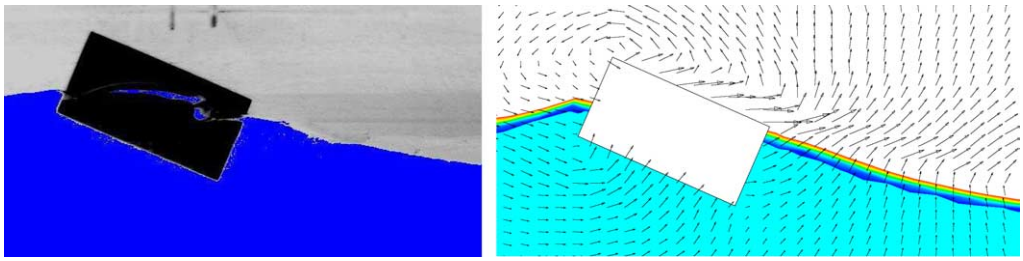

Fig. 6. Free-surface shape and body position as observed in the experiment (left) and the corresponding situation from simulation (right) at $t = 7.2$ s.

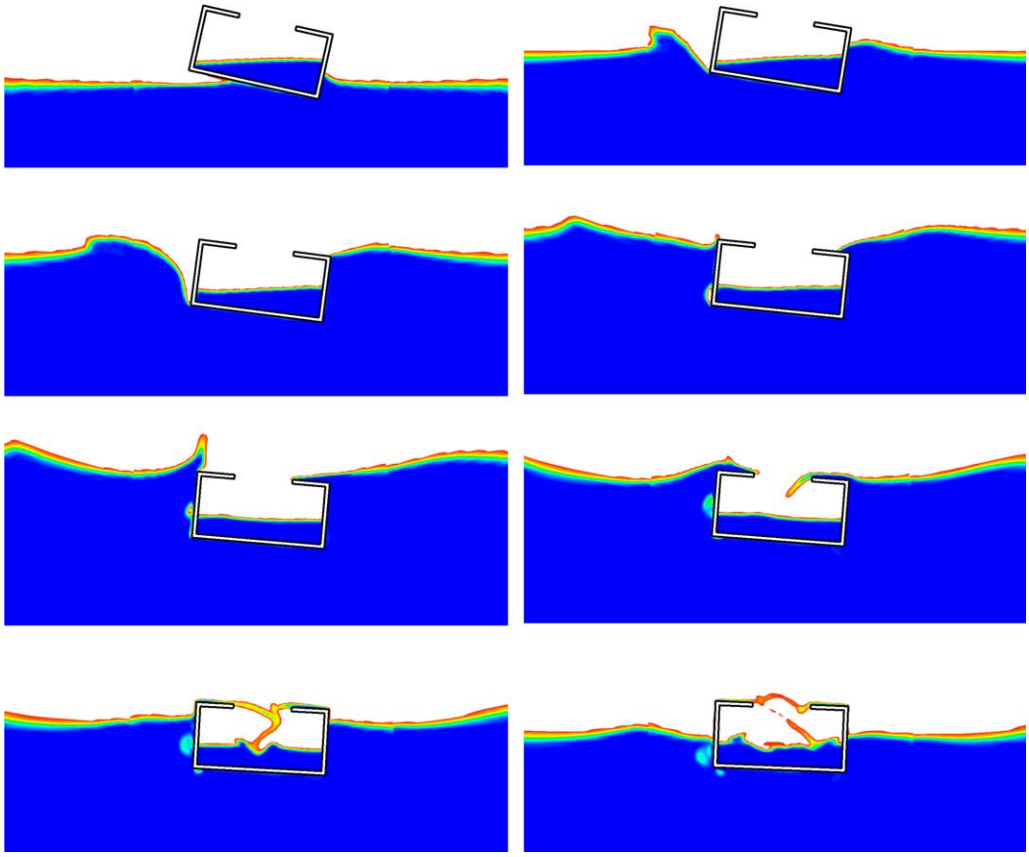

Fig. 7. Free-surface shape and body position in the simulation of a free-fall drop of an open container into water (from left to right, top to bottom).

A close-up of the free-surface shape and body position for one time instant is shown in Fig. 6 for both experiment and simulation. Numerical simulations have also been performed for bodies with forward speed and for other wave configurations in both two and three dimensions and will be reported in the future.

Another type of studies with coupled simulation of flow and body motion involves free fall of bodies through air and/or into water. In Fig. 7 a sequence of plots from one such simulation is presented. They show a partially-filled open container falling in an inclined position onto the free surface, creating a crater in the liquid and causing sloshing of liquid inside the container. Also, some liquid from outside falls into the container. The motion of the container is governed by forces caused by the fluid motion both inside and outside. This is just a demonstration of the possible application of the method to such problems which are relevant e.g. for tanker ships and other floating vessels.

6. Conclusion

Motions of floating bodies have been predicted by a coupled approach solving the Reynolds-averaged Navier–Stokes equations and body dynamics equations simultaneously. The finite volume method used in the flow solver employs moving-grids to account for the instant position of the body motion. The method proved to be capable of reproducing the behaviour of a 2D floating body subjected to wave packages, showing good agreement with experiments. In addition, simulation of motion of a 2D vessel subjected to fluid flow both inside and outside it has been presented, demonstrating the potential for using the method in very complex problems.

Although all test computations presented here dealt with 2D problems, the methodology has been formulated and implemented for 6 DOF and 3D flow problems. The main cost in the simulations comes from fluid flow prediction, with body motion and grid adaptation representing only a minor fraction of computing time. Faster computers and parallel processing will make simulations of 3D flows around floating bodies like ships accessible to industrial users; some results of such simulations are presented in Ref. [18].

Future extensions of the method will include introduction of overlapping grids, in order to avoid the problems associated with grid adaptation to body motion.

Acknowledgement

We gratefully acknowledge financial support received by I. Hadzic and Y. Xing-Kaeding from the Federal Ministry for Education and Research (BMBF) of the FR Germany within the ROLL-S research project.

References

[1] V. Bertram, Practical Ship Hydrodynamics, Butterworth-Heinemann, Oxford, 2000.
[2] T. Kinoshita, H. Kagemoto, M. Fujino, A CFD application to wave-induced floating-bodies, Seventh Int. Conf. on Numerical Ship Hydrodynamics, Nantes/France, 1999, 4.1-(1-20).

[3] Y. Sato, H. Miyata, T. Sato, CFD simulation of 3-dimensional motion of a ship in waves: application to an advancing ship in regular head waves, J. Mar. Sci. Technol. 4 (1999) 108–116.
[4] P. Ferant, L. Gentaz, D.L. Touzé, A new potential/RANSE approach for water wave diffraction, Fifth Numerical Towing Tank Symp., Pornichet/France, 2003.
[5] R. Azcueta, Computation of turbulent, free-surface flows around ships and floating bodies, Ph.D thesis, TU Hamburg-harburg, 2001.
[6] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.
[7] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345–380.
[8] Comet user manual, ICCM (Institute of Computational Continuum Mechanics GmbH), Hamburg, 2000.
[9] C.M. Rhie, W.L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[10] S. Muzaferija, M. Perić, Computation of free surface flows using interface-tracking and interface-capturing methods, chapter 2, in: O. Mahrenholtz, M. Markiewicz (Eds.), Nonlinear Water Wave Interaction, WIT Press, Southampton, 1999, pp. 59–100.
[11] J.H. Ferziger, M. Perić, Computational Method for Fluid Dynamics, third ed., Springer-Verlag, Berlin, 2002.
[12] D.C. Wilcox, Turbulence Modelling for CFD, DCW Industries, Inc., La Cañada, California, 1998.
[13] B.E. Launder, D.B. Spalding, The numerical computation of turbulent flows, Comput. Methods Appl. Mech. Eng. 3 (1974) 269–289.
[14] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modelling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[15] I. Demirdžić, S. Muzaferija, M. Perić, Computation of turbulent flows in complex geometries, in: G. Tzabiras et al. (Eds.), Calculation of Complex Turbulent Flows, WIT Press, Southampton, 2000, pp. 249–299 (Chapter 7).
[16] J.N. Newman, Marine Hydrodynamics, MIT Press, England, 1978.
[17] Y. Xing, I. Hadzić, S. Muzaferija, M. Perić, CFD simulation of flow-induced floating-body motions, in: Proc. 16th Int. Workshop on Water Waves and Floating Bodies, Hiroshima/Japan, 2001, pp. 169–172.
[18] Y. Xing-Kaeding, G. Jensen, I. Hadžić, M. Perić, Simulation of flow-induced ship motions in waves using a RANSE method, J. Ship Techn. Res. 51 (2004) 56–68.