A fully nonlinear potential model for ship hydrodynamics directly interfaced with CAD data structures

Andrea Mola, Luca Heltai, Antonio DeSimone
SISSA - International School for Advanced Studies,
Trieste, TS, Italy

ABSTRACT

We present a model for ship hydrodynamics simulations currently under development at SISSA. The model employs potential flow theory and fully nonlinear free surface boundary conditions. The spatial discretization of the equations is performed by means of a collocation BEM. This gives rise to a Differential Algbraic Equations (DAE) system, solved using an implicit BDF scheme to time advance the solution. The model has been implemented into a C++ software able to automatically generate the computational grids from the CAD geometry of the hull. Numerical results on Kriso KCS and KVLCC2 hulls are presented and discussed.

KEY WORDS: ship hydrodynamics; free surface; fully nonlinear potential; Boundary Element Method; semi-Lagrangian; CAD surfaces; automatic mesh generation.

INTRODUCTION

In the ship design process, the integration of computational tools for hydrodynamic performance prediction has steadily progressed in the last decades. Given the constant increase in computing power and the improvement in models and numerical algorithms, ship performance prediction through computer simulation has nowadays become a reliable complement –and in some case a cost effective alternative– to the experimental approach.

Yet, even in the best practice, one of the main factors limiting such integration process is represented by the complex problem of obtaining the computational grid needed for the simulations from the hull geometrical description. At the industrial level, the geometrical model of the hull is in fact typically generated by a Computer-Aided Design (CAD) tool, and exported in electronic files for print, machining, or other manufacturing operations. In most cases, such geometrical model is not immediately suitable for mesh generation purposes. To allow for more freedom in the shape design, the parametric surface patches composing the hull shape are in fact typically disconnected one from each other. The resulting small gaps and overlaps between each patch pose serious problems in delimiting the domain for volume mesh generation. Moreover, exploiting the patches parametric coordinates to generate new nodes is not possible if neighboring patches do not share an edge having a common parametrization. These problems are of course even more relevant for hydrodynamic models to be used in the early design stages, when a faster performance prediction is required.

In this work we present a model based on flow potential theory, which allows a Boundary Integral Formulation: only the boundary of the domain needs to be discretized and partitioned into a surface grid in the three dimensional space. In this approach, it is not necessary to reconstruct a volumetric domain from the surface geometry of the hull.

In potential flow theory, the governing equation for the velocity potential is the Laplace equation written in the flow domain represented by the portion of water surrounding the hull. In this work, the Laplace equation is complemented by fully nonlinear unsteady boundary conditions on the water surface. In order to avoid re-meshing whenever the computational grid changes, both the free surface markers and the collocation points are constructed in order coincide with the computational grid nodes. For the same reason, the flow equations are solved in a frame of reference attached to the ship hull, requiring the presence of a water current in the simulations. The boundary conditions are then written in semi-Lagrangian form (Beck, 1994) to avoid the streamwise transport of the grid nodes, which would again force periodic remeshing during the simulations.

Historically, the numerical model described has always posed stability problems for the numerical treatment of the free surface boundary conditions time evolution. This is due to inaccurate reconstructions of the position vector and of potential field gradients, and –more importantly– to the presence of a dominant transport instability in the evolution equations for free surface position and potential. In this work, an L^2 projection approach is used for accurate gradient reconstructions on the free surface nodes, and a Streamwise Upwind Petrov-Galerkin method is used to remove instabilities (Mola, Heltai and DeSimone, 2013).

With this numerical machinery in place, it was possible to build an algorithm which could import the hull CAD model, and use it both to generate the computational grid and manage its evolution during the simulations. To allow for arbitrary hull surfaces composed of disconnected patches, the new nodes are created in the physical, three dimensional space of the simulations, and successively projected onto the CAD surfaces. Given the limited number of nodes of a surface grid, the computational cost added by the projections is not significant. The open source software library OpenCASCADE Community Edition (OCE) has been employed to implement the modules to read, process and interrogate the CAD models.

Results of the proposed algorithm on geometries of industrial interest are finally presented and commented. The test cases considered are benchmarks proposed by the Korea Research Institute of Ships and Ocean Engineering (KRISO) in (Kim, Van and Kim, 2001). The first hull considered is the KRISO Container Ship (KCS), which presents both a bow and stern bulb. The second is the KRISO Very Large

Crude-Oil Carrier 2 (KVLCC2), which also has bulbous bow and stern. Since the CAD model of both hulls is composed of several unconnected parametric surfaces, also the grid generation algorithm is put to test.

FULLY NONLINEAR POTENTIAL MODEL FOR FREE SURFACE FLOWS

We study the hydrodynamics of a ship with fixed sink and trim in a coordinate system attached to the ship hull. In such reference frame, the □ axis is directed as the undisturbed stream velocity V_∞, the z axis is directed vertically, pointing upwards, and the y axis is perpendicular to the symmetry plane of the hull, pointing towards its port side. Thus, in the reference system chosen, the asymptotic flow velocity vector is $\boldsymbol{V}_\infty = (V_\infty, 0, 0)$. The flow domain considered, is the simply connected and time dependent region $\Omega(t)$ representing the portion of water surrounding the hull. In such volume, for an irrotational and inviscid flow, the velocity field $\boldsymbol{v}(\boldsymbol{x}.t)$ admits a representation through a scalar potential function $\Phi(\boldsymbol{x}, t)$, namely

$$\boldsymbol{v} = \boldsymbol{\nabla}\Phi = \boldsymbol{V}_\infty \cdot \boldsymbol{x} + \phi \qquad \forall\, \boldsymbol{x} = (x, y, z) \in \Omega(t), \tag{1}$$

in which $\phi(\boldsymbol{x}, t)$ is the so called *perturbation potential*. In this case, the equations of motion simplify to the unsteady Bernoulli equation and to the Laplace equation for the flow potential

$$\begin{aligned} &\frac{\partial \phi}{\partial t} + \frac{1}{2}\left|\boldsymbol{\nabla}\phi + \boldsymbol{V}_\infty\right|^2 + \frac{p - p_a}{\rho} - \boldsymbol{g}\cdot\boldsymbol{x} = C(t) && \text{in } \Omega(t) \\ &-\Delta\phi = 0 && \text{in } \Omega(t) \end{aligned} \tag{2}$$

where $C(t)$ is an arbitrary function of time and $\boldsymbol{g} = (0, 0, -g)$ is the gravity acceleration vector. The unknowns ϕ and p of Syst. 2 are uncoupled. Hence, it is possible to recover the pressure field making use of Bernoulli's equation to postprocess the solution of the Laplace equation for the perturbation potential. On the water free surface $\Gamma^w(t)$ –the shape of which is unknown *a priori*– the Laplace equation is complemented by the kinematic and dynamic fully nonlinear boundary conditions expressed in semi-Lagrangian form, which respectively read

$$\begin{aligned} \frac{\delta\eta}{\delta t} &= \frac{\partial\phi}{\partial z} + \boldsymbol{\nabla}\eta\cdot(\boldsymbol{w} - \boldsymbol{\nabla}\phi - \boldsymbol{V}_\infty) && \text{in } \Gamma^w(t) \\ \frac{\delta\phi}{\delta t} &= -g\eta + \frac{1}{2}|\boldsymbol{\nabla}\phi|^2 + \boldsymbol{\nabla}\varphi\cdot(\boldsymbol{w} - \boldsymbol{\nabla}\phi - \boldsymbol{V}_\infty) && \text{in } \Gamma^w(t). \end{aligned} \tag{3}$$

The former equation expresses the fact that a point on the free surface will move remaining on the free surface, which is here assumed to be the graph of a single valued function $\eta(x, y, t)$ of the horizontal components x and y of the position vector $\boldsymbol{x}$. The latter condition represents a manipulation of Bernoulli's equation, under the assumption of constant atmospheric pressure on the water surface. These fully nonlinear boundary conditions, proposed by Beck (1994), allow for the computation of the vertical velocity of markers which move on the water free surface with a prescribed horizontal speed (w_x, w_y) in the boat reference frame, along with the corresponding velocity potential values. The resulting vector $\boldsymbol{w} = (\mathrm{w}_x, w_y, \frac{\delta\eta}{\delta t}) = \dot{\boldsymbol{x}}$ is the time derivative of the position of the free surface markers.
In this work, such free surface markers are chosen as the free surface nodes of the computational grid. Their arbitrary horizontal velocity is null along the x direction, while the y component chosen in order to preserve mesh quality.

NUMERICAL DISCRETIZATION BASED ON BEM AND DAE

The boundary value problem described is governed by the linear Laplace operator. Yet, it is nonlinear due to the presence of the boundary conditions in Eq. 3. Further sources of nonlinearity are given by continuous change of the domain shape over time and by the arbitrary shape of the ship hull. Thus, for each time instant, we will look for the correct values of the unknown potential and node displacement fields by solving a specific nonlinear problem resulting from the spatial and time discretization of the original boundary value problem.
The spatial discretization of the Laplace equation, is based upon a boundary integral formulation. This results in a system of Differential Algebraic Equations (DAE) which is solved to advance the solution over time. This section will briefly describe such solution strategy. The present section will present the main features of such methodology.

Spatial Discretization Method Based on Boundary Integral Formulation

While Eq. 2 is defined in the entire domain $\Omega(t)$, we are really only interested in its solution on the boundary $\Gamma(t)$, in particular on the unknown free surface part of the boundary, and on the ship hull $\Gamma^b(t)$, where we would like to recover the pressure distribution through Bernoulli's equation.
At any given time instant $\bar{t}$ we want to compute ϕ satisfying

$$\begin{aligned} -\Delta\phi &= 0 && \text{in } \Omega(\bar{t}) \\ \phi &= \overline{\phi} && \text{on } \Gamma^w(\bar{t}) \\ \phi_n &= -\boldsymbol{V}_\infty\cdot\boldsymbol{n} && \text{on } \Gamma^b(\bar{t}) \end{aligned} \tag{4}$$

where $\overline{\phi}$ is the potential on the free surface at time $\bar{t}$.
This is a purely spatial boundary value problem, in which time appears only through boundary conditions and through the shape of the time dependent domain. The discretization of Prob. 4 moves from the definition of the *free space Green's function* G, namely

$$G(\boldsymbol{r}) = \frac{1}{4\pi|\boldsymbol{r}|}, \tag{5}$$

which is the distributional solution of

$$\begin{aligned} &-\Delta G(\boldsymbol{x} - \boldsymbol{x}_0) = \delta(\boldsymbol{x}_0) && \text{in } \mathbb{R}^3 \\ &\lim_{|\boldsymbol{x}|\to\infty} G(\boldsymbol{x} - \boldsymbol{x}_0) = 0, \end{aligned} \tag{6}$$

where $\delta(\boldsymbol{x}_0)$ is the Dirac distribution centered in $\boldsymbol{x}_0$.
If we select $\boldsymbol{x}_0$ to be in $\Omega(t)$, multiply the Laplace equation by G, and use the defining property of the Dirac delta and the second Green identity, we obtain

$$\begin{aligned} \phi(\boldsymbol{x}_0, t) \quad = \quad \int_{\Gamma(t)} \big[&(\boldsymbol{\nabla}\phi(\boldsymbol{x}, t)\cdot\boldsymbol{n})G(\boldsymbol{x} - \boldsymbol{x}_0) - \\ &(\boldsymbol{\nabla}G(\boldsymbol{x} - \boldsymbol{x}_0)\cdot\boldsymbol{n})\phi(\boldsymbol{x}, t)\big]\,\mathrm{d}\Gamma. \end{aligned} \tag{7}$$

In the limit for $\boldsymbol{x}_0$ touching the boundary $\Gamma(t)$, the integral on the right hand side will have a singular argument, and should be evaluated according to the Cauchy principal value. This process yields the so called *Boundary Integral Equation* (BIE)

$$\alpha\phi = \int_{\Gamma(t)} \left[\phi_n G - \frac{\partial G}{\partial n}\phi\right] \mathrm{d}\Gamma \qquad \text{on } \Gamma(t), \tag{8}$$

where $\alpha(\boldsymbol{x},t)$ is the fraction of solid angle 4π with which the domain $\Omega(t)$ is seen from $\boldsymbol{x}$.
With Eq. 8, the continuity equation has been reformulated as a boundary integral equation of mixed type defined on the moving boundary $\Gamma(t)$, where the main ingredients are the perturbation potential $\phi(\boldsymbol{x},t)$ and its normal derivative $\phi_n(\boldsymbol{x},t)$.

Iso-parametric spatial discretization. We approximate the geometry of the domain boundary by means of arbitrary order quadrilateral panels, that is, panels for which the sides and the interior are composed of polynomials and polynomial tensor products respectively.
In particular we define the Lagrangian shape functions $\mathrm{N}_l(u,v) \quad l = 1,\ldots,N_L$ on the reference panel, which allow us to introduce a local parametrization of the k-th panel as

$$\begin{aligned} \tilde{\boldsymbol{x}}_k(u,v) &:= \sum_{l=1}^{N_L} \tilde{\boldsymbol{x}}^{k_l} N_l(u,v) \qquad\qquad u,v \in [0,1]^2 \\ \boldsymbol{x}_k(u,v,t) &:= \sum_{l=1}^{N_L} \boldsymbol{x}^{k_l}(t) N_l(u,v) \qquad\qquad u,v \in [0,1]^2, \end{aligned} \tag{9}$$

where the weights are the positions of the nodes in the reference domain $\tilde{\Gamma}_h$, or the nodes in the current domain $\Gamma_h(t)$, and k_l is the *local to global* numbering index which identifies the N_L basis functions φ^{k_l} which are different from zero on the k-th panel. The current geometry $\Gamma_h(t)$ is constructed with $\boldsymbol{x}^{k_l}(t)$ instead of $\tilde{\boldsymbol{x}}^{k_l}$, but with the same local parametrization.
The global basis functions $\varphi^i(\tilde{\boldsymbol{x}})$ can be identified and evaluated on each panel k via their local parametrization as

$$\begin{aligned} \varphi^i_k(u,v) := \varphi^i(\tilde{\boldsymbol{x}}_k(u,v)) &= \sum_{l=1}^{N_L} \delta_{i\,k_l} N_l(u,v), \\ \delta_{ij} &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{10}$$

The local representation of $\phi(\boldsymbol{x}^k(u,v,t),t)$ and of its normal derivative on the k-th panel are

$$\begin{aligned} \phi_k(u,v,t) &= \sum_{l=1}^{N_L} \phi^{k_l}(t) N_l(u,v) \\ \phi_{nk}(u,v,t) &= \sum_{l=1}^{N} {\phi_n}^{k_l}(t) N_l(u,v), \end{aligned} \tag{11}$$

where $\phi^{k_l}, {\phi_n}^{k_l} \; l = 1,\ldots,N_L$ are the nodal values of the potential and of its normal derivative in panel k.

Collocation Boundary Element Method. With the iso-parametric representation, we write the boundary integral equation for each support point $\boldsymbol{x}^i$, $i = 1,\ldots,N_V$, and finally recast the discrete version of the boundary integral equation as

$$[\alpha]\{\phi\} + [N]\{\phi\} = [D]\{\phi_n\} \tag{12}$$

where we have made use of the following notation:

- $\{\phi\}$ and $\{\phi_n\}$ are the vectors containing the potential and its normal derivative node values, respectively;
- $[\alpha]$ is a diagonal matrix composed by the $\alpha(\boldsymbol{x}_i(t))$ coefficients;
- $[D]$ and $[N]$ are the Dirichlet and Neumann matrices respectively whose elements are

$$\begin{aligned} D_{ij} &= \sum_{k=1}^{M} \int_{\hat{K}} G(\boldsymbol{x}_i(t) - \boldsymbol{x}^k(u,v,t))\varphi^j_k(u,v) J^k(u,v,t)\,\mathrm{d}u\,\mathrm{d}v \\ N_{ij} &= \sum_{k=1}^{M} \int_{\hat{K}} \frac{\partial G}{\partial n}(\boldsymbol{x}_i(t) - \boldsymbol{x}(u,v,t))\varphi^j_k(u,v) J^k(u,v,t)\,\mathrm{d}u\,\mathrm{d}v, \end{aligned} \tag{13}$$

  $\mathrm{J}^k(u,v,t)$ being the determinant of the first fundamental form.

The numerical evaluation of the panel integrals appearing in Eq. 13 needs some special treatment, due to the presence of the singular kernels $G(\boldsymbol{y}-\boldsymbol{x})$ and $\frac{\partial G}{\partial n}(\boldsymbol{y}-\boldsymbol{x})$. Whenever $\boldsymbol{y}$ is not a node of the integration panel, the integral argument is not singular, and standard Gauss quadrature formulas can be used. If $\boldsymbol{y}$ is a node of the integration panel, the integral kernel is singular and special quadrature rules are used, which remove the singularity by performing an additional change of variables (Lachat and Watson, 1976).
This collocation BEM solver has been implemented using the Open Source C++ library deal.II (Bangerth, Hartmann, and Kanschat, 2007, DeSimone, Heltai and Manigrasso, 2009, Bangerth, Heister, Heltai, Kanschat, Kronbichler, Maier and Young, 2013), and it is an evolution of the software presented in (Mola, Heltai and DeSimone, 2013).

Time Discretization Based on Differential Algebraic System Formulation

Once the spatial discretization described has taken place, the time dependent boundary value problem composed by Laplace equation and by the kinematic and dynamic boundary condition s can be recast in the following form

$$F(t,y,y') = 0, \tag{14}$$

where we grouped the variables of the system in the vector y:

$$y = \left\{ \begin{array}{c} \{\boldsymbol{x}\} \\ \{\phi\} \\ \{\phi_n\} \end{array} \right\}. \tag{15}$$

Eq. 14 represents a system of nonlinear Differential Algebraic Equations (DAE), which we solve using the IDA package of the SUNDIALS OpenSource library (Hindmarsh, Brown, Grant, Lee, Serban, Shumaker et al., 2005). The integration of such DAE system is performed through a variable-order, variable-coefficient BDF (Backward Difference Formula), which reads

$$\sum_{i=0}^{q} \alpha_{n,i} y_{n\;\;i} = h_n \dot{y}_n, \tag{16}$$

where y_n and $\dot{y}_n$ are the computed approximations to $y(t_n)$ and $y'(t_n)$, respectively, and the step size is $h_n = t_n - t_{n-1}$. The coefficients $\alpha_{n,i}$ are uniquely determined by the order q, and the history of the step sizes. The application of the BDF scheme to the DAE system results, at each time step, in a nonlinear algebraic system, solved by means of Newton iteration.

SUPG Stabilization

The gradient of the perturbation potential is not, in general, continuous across the edges of the panels that compose $\Gamma_h(t)$, and therefore it is

not possible to write directly an evolution equation for the vertices of the triangulation and for the potential nodal values, since on the location of the nodes the forcing term is not single valued.
It is certainly possible to solve this issue by using smooth finite dimensional spaces, as in~\cite{grilli2001}, but this is not strictly necessary. A possible alternative is to impose the evolutionary boundary conditions via an L^2 projection in the V_h space, *i.e.*, we substitute Eqs. 3 with

$$\begin{aligned}\left(\frac{\delta\phi}{\delta t},\varphi\right)_w &= \left(V_\phi,\varphi\right)_w \\ &= \left(\frac{\partial\phi}{\partial z}+\boldsymbol{\nabla}\eta\cdot\left(\boldsymbol{w}-\boldsymbol{\nabla}\phi-\boldsymbol{V}_\infty\right),\varphi\right)_w \qquad \forall\varphi\in V_h \\ \left(\frac{\delta\phi}{\delta t},\varphi\right)_w &= \left(V_\phi,\varphi\right)_w \\ &= \left(-g\eta+\frac{1}{2}|\boldsymbol{\nabla}\phi|^2+\boldsymbol{\nabla}\varphi\cdot\left(\boldsymbol{w}-\boldsymbol{\nabla}\phi-\boldsymbol{V}_\infty\right),\varphi\right)_w \quad \forall\varphi\in V_h\end{aligned} \tag{17}$$

where

$$\begin{aligned}\left(a,b\right)_w &= \int_{\Gamma^w(t)} ab \ \mathrm{d}\Gamma \\ \left(a,b\right) &= \int_{\Gamma(t)} ab \ \mathrm{d}\Gamma.\end{aligned} \tag{18}$$

However, the forcing terms of the evolution Eqs. (3) contain transport terms, respectively $\boldsymbol{\nabla}\eta\cdot\left(\boldsymbol{w}-\boldsymbol{\nabla}\phi-\boldsymbol{V}_\infty\right)$ and $\boldsymbol{\nabla}\phi\cdot\left(\boldsymbol{w}-\boldsymbol{\nabla}\phi-\boldsymbol{V}_\infty\right)$ which become dominant whenever $\left(\boldsymbol{\nabla}\phi+\boldsymbol{V}_\infty\right)$ is very different from $\boldsymbol{w}$, causing a sawtooth numerical instability which in most cases develops in proximity of the hull stern, with consequent blow up of the simulations.
Once the L^2 projection machinery is in place, as in Eqs. 17, a natural and consistent stabilization mechanism that reduces the observed instabilities is the Streamwise Upwind Petrov–Galerkin (SUPG) scheme (see for example, Hughes and Brooks, 1979, Tezduyar, 2003). In the SUPG framework, the plain L^2 projection in Eqs. 17 is replaced by the weighted projection

$$\begin{aligned}\left(\frac{\delta\phi}{\delta t},\varphi+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi\right)_w &= \left(V_\phi,\varphi+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi\right)_w \qquad \forall\varphi\in V_h \\ \left(\frac{\delta\eta}{\delta t},\varphi+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi\right)_w &= \left(V_\eta,\varphi+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi\right)_w \qquad \forall\varphi\in V_h,\end{aligned} \tag{20}$$

where

$$\boldsymbol{d}:=\tau\left(\frac{\boldsymbol{v}-\boldsymbol{w}}{|\boldsymbol{v}-\boldsymbol{w}|}\right), \tag{21}$$

and τ is a positive stabilization parameter which involves a measure of the local length scale (*i.e.* the ``element length'') and the local Reynolds and Courant numbers. Element lengths and stabilization parameters were proposed for the SUPG formulation of incompressible and compressible flows in (Hughes and Brooks, 1979), and an in depth study of the stabilization properties for free boundary problems was presented in (Tezduyar, 2003). The use of this type of stabilization for free surface problems based on boundary element approximations was first introduced in (Mola, Heltai and DeSimone, 2013).
Expressing everything in terms of the basis functions for finite dimensional space V_h, we get

$$\begin{aligned}[M]\{\phi\}' &= \{V_\phi\} \\ [M]\{\eta\}' &= \{V_\eta\},\end{aligned} \tag{22}$$

where

$$\begin{aligned}M^{ij} &:= \left(\varphi^j,\varphi^i+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i\right) \\ &= \sum_{k=1}^{M}\int_{\hat{K}}\varphi_k^j(u,v)\left(\varphi^i(u,v)+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i(u,v)\right)J^k(u,v,t)\,\mathrm{d}u\,\mathrm{d}v \\ V_\phi^i &:= \left(V_\phi,\varphi^i+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i\right) \\ &= \sum_{k=1}^{M}\int_{\hat{K}}V_\phi(u,v)\left(\varphi^i(u,v)+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i(u,v)\right)J^k(u,v,t)\,\mathrm{d}u\,\mathrm{d}v \\ V_\eta^i &:= \left(V_\eta,\varphi^i+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i\right) \\ &= \sum_{k=1}^{M}\int_{\hat{K}}V_\eta(u,v)\left(\varphi^i(u,v)+\boldsymbol{d}\cdot\boldsymbol{\nabla}_s\varphi^i(u,v)\right)J^k(u,v,t)\,\mathrm{d}u\,\mathrm{d}v.\end{aligned} \tag{23}$$

Horizontal nodes motion and mesh smoothing

The procedure described is applied to write the discrete form of the differential equations appearing in the DAE System 14. Taking a look at Eq. 15, only the vertical position of the free surface nodes and the corresponding potential values are governed by differential equations. As for the rest of the components of vector y, their evolution is ruled by algebraic equations. For the potential ϕ and for its normal gradient $\frac{\partial\phi}{\partial n}$, all the algebraic equations are obtained from the BEM algebraic system presented in Eq. 12. The equations for the coordinates of nodes not belonging to the free surface are simply used to keep such nodes idle during the simulations. The algebraic equations associated to the horizontal coordinates of the free surface nodes require instead a specific treatment.
The equations for the horizontal coordinates of the nodes belonging to both water and hull are written to impose a null horizontal distance between such nodes and the boat surface. As for the horizontal coordinates of the remaining free surface nodes, their position is determined by means of the following equations

$$\begin{aligned}\left(\nabla(x-L),\nabla\varphi\right)_w &= \left(-\Delta L,\varphi\right)_w \qquad \forall\varphi\in V_h \\ \left(\nabla(y-L),\nabla\varphi\right)_w &= \left(-\Delta L,\varphi\right)_w \qquad \forall\varphi\in V_h.\end{aligned} \tag{24}$$

Here, $L(\boldsymbol{x})$ is a lift operator used to impose the proper boundary values to the coordinates. Eq. 24 represents the semi-discrete form of a Laplacian smoothing for both x and y coordinates of the free surface nodes.
We remark here that, even if the equations for the horizontal coordinates of free surface nodes are algebraic, the time derivative of such coordinates $\boldsymbol{w}$ is readily available through the evaluation of the BDF in Eq. 16. Such velocities are used in the discretized Eqs. 17 which appear in the DAE system. In particular, at each time step, the convergence of Newton corrections ensures that the vertical velocity of the nodes is the one corresponding to the $\boldsymbol{w}$ velocity originated by the horizontal nodes displacement computed by the smoothing operator.

TREATMENT OF CAD SURFACES FOR FULLY AUTOMATED MESH GENERATION

One of the most important advantages of the Boundary Integral Formulation adopted, is that only the boundary of the three dimensional domain needs to be discretized. Thus, the reduced complexity of the superficial computational grids needed, has been exploited in this work to achieve full automation of the mesh generation process.

Nowadays, the design of a ship hull is typically carried out using a Computer-Aided Design (CAD) tool. Such tools allow for the generation of a virtual model of the hull surface. Their output is often in the form of electronic files for print, machining, or other manufacturing operations. Fig. 1 shows a 3D rendering of the CAD model of a Kriso Container Ship (KCS) hull. As can be observed in the picture, several parametric surfaces are patched together to compose the whole ship surface. At the industrial level, there is no requirement that such patches are logically connected one to each other, nor that the full hull surface is C^0 or even C^1. Most of CAD shapes, in fact, present several small gaps or overlaps between each surface composing them. A small tolerance can in fact be set during the CAD model generation, to control the dimension of such imperfections in a way that makes them negligible once the actual hull is crafted.

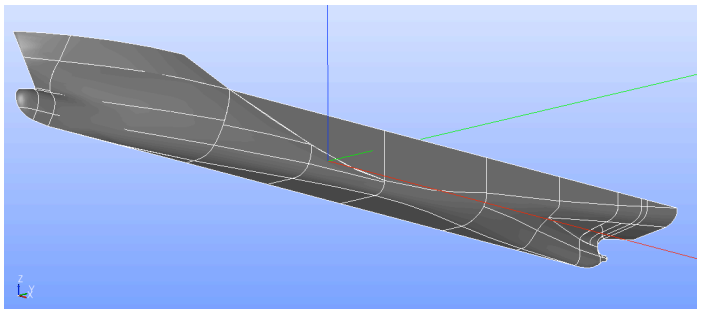


Fig. 1. The CAD model of the Kriso Container Ship (KCS) hull, composed of 57 unconnected surfaces.

Yet, the presence of disconnected and overlapping patches impairs in most cases the generation of computational grids for fluid mechanics simulations purposes. A first reason for this, is that most solvers for fluid dynamics equations require the generation of a three dimensional grid in the volume of fluid surrounding the hull surface. Such domain is in most cases obtained via boolean subtraction of the hull volume from a large box of fluid surrounding it. Clearly, in presence of gaps and overlaps between the hull surface patches, both delimiting the hull volume and using it to perform boolean operations is extremely difficult. But there is a second and even more important reason. Most mesh generators are based on the choice of generating the surface mesh nodes in the parametric space of each patch composing the CAD model. This gives the obvious advantage that any point created will be located exactly on the desired surface. Yet, this forces the CAD model to have logically connected surfaces, in which neighboring surfaces will share the same parametrization at the common edges. If this *water tightness* requirement is not met by the CAD model, it is impossible to obtain meshes that are connected at the patches junctions.

The automatic mesh generation algorithm presented in this work, has been designed to deal with arbitrary, non water tight hull geometries. This was obtained exploiting the peculiar characteristics of the BEM solver developed, which only requires a surface mesh on the flow domain boundaries. Thus, it is not necessary here, to delimit the volume of the hull, and use it to perform any boolean operations. In addition, the algorithm does not operate in the parametric space of each patch. New nodes in the mesh are in fact generated in the physical, three dimensional space in which the simulations will be run. Clearly, this results in the additional computational cost represented by their projection on the CAD surface. But again, since the BEM solver only requires a surface mesh the number of nodes to be projected is usually not large enough to significantly slow the mesh generation process.

The present section will present details of the mesh generation algorithm developed. First, it will describe the basic geometric operations performed on the CAD surfaces. Then, it will illustrate the procedure for the initial mesh generation and hierarchic refinement.

Basic Geometric Operations

In this work, we made use of the Open CASCADE Community Edition (OCE) open source library to import and interrogate CAD models in the mesh generation module of the BEM solver developed. While building the grid, a certain number of fundamental geometric functions are in fact needed to properly place the newly created nodes on the hull surface and on its edges.

To this end, we implemented a series of wrappers that use the OCE functions to provide the mesh handler with the geometrical tools needed. In this framework, as soon as the CAD model is imported, it is processed to generate a set of useful set of curves. In particular, the curve representing the intersection between the hull surface and its symmetry plane is created by identifying and joining all the patches edges lying on such plane. The same procedure is applied to create the hull intersection with undisturbed free surface, and the transom edge.

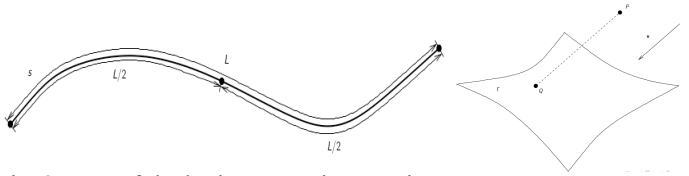


Fig. 2. Two of the basic geometric operations performed interrogating the CAD model. On the left, the generation of a new point which splits the arc length of a curve at a prescribed fraction. On the right, the projection of a point on a surface along a specified direction.

Such curves and the hull surface are then used by a series of projectors to place the new nodes on the boat surface. If a mesh edge sitting on one of the curves is to be refined, the algorithm resorts to the *arc length projector* to place the new node so as to split the original arc length of the edge at the specified fraction. In addition, a *surface projector* is used to project a new internal node on the hull surface along a prescribed direction. The arc length and surface projectors are depicted in Fig. 2, on the left and right side respectively.

Hierarchic mesh refinement

A grid composed of a very small number of quadrilaterals is initially generated. The points are placed at strategic locations on the hull intersections with its symmetry plane and with the undisturbed free surface. At this stage, the hull is discretized only in two cells per each side. Fig. 3 shows a top view of the initial mesh (left) and a close-up view of the hull cells (right).

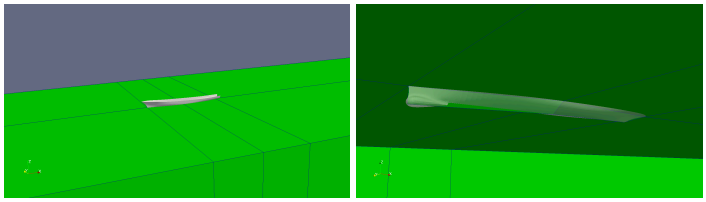


Fig. 3. The initial grid. The left picture shows a top view, and the right picture portraits a detail of the boat cells.

Starting from this initial mesh, the cells are hierarchically refined to improve the level of approximation of the original CAD geometry. For each quadrilateral cell on the hull, five new nodes are generated (see Fig. 4) on the cell surface. These nodes are then projected on the hull surface making use of the projectors previously described. In particular, all the internal nodes are projected on the hull along the mesh normal direction employing the surface projector. The new nodes located on

the hull symmetry plane curve and transom stern edge, will be projected on the corresponding curve using the arc length projector.

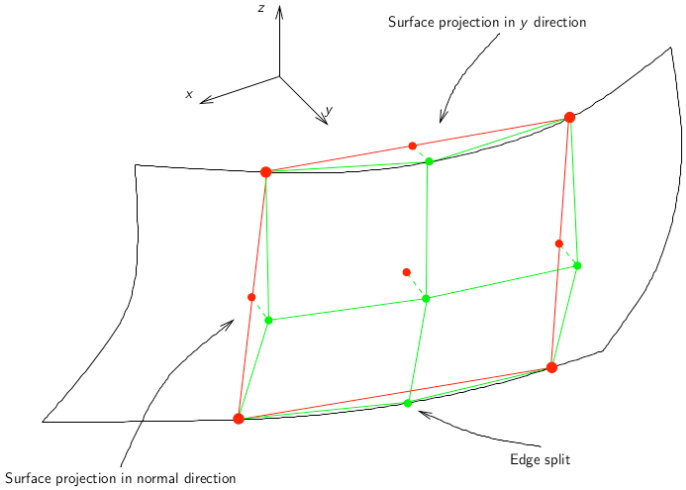


Fig. 4. The refinement procedure for a hull cell at the first refinement cycle. Internal nodes are projected on the surface in the mesh normal direction, while the boat nodes located on the free surface are projected in the y direction. Finally, the cell edge placed on the intersection with the symmetry plane, is refined via the arc length projection previously described.

All the cells on the ship hull are refined according this procedure. It is worth pointing out, that every new node is first generated on the old mesh, and then projected onto the CAD surface. Being independent on the surface parametrization, such procedure is not affected by the presence of gaps and overlaps between the surface patches. Even if a node projection falls in a small gap, the projector finds the closest point to the node on the gap border.
The uniform refinement of the hull mesh described is iterated several times, until sufficient quality is achieved. Yet, to enhance the grid quality with a limited amount of cells, it is desirable to concentrate the nodes in the regions where the highest curvature is located. For this reason, a final number of local refinement cycles is finally carried out. At each refinement step, the cells to be refined are identified based on the on an error estimator represented by the distance between the center cell and its projection on the boat surface. The specified fraction of cells with highest error values are then refined. A sketch of this procedure is shown in Fig. 5.

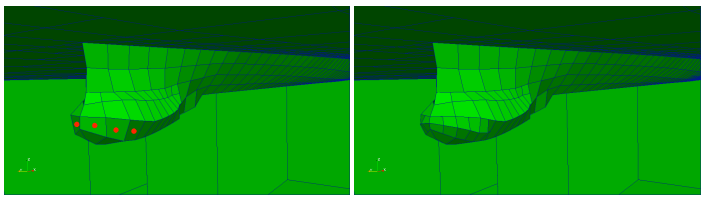


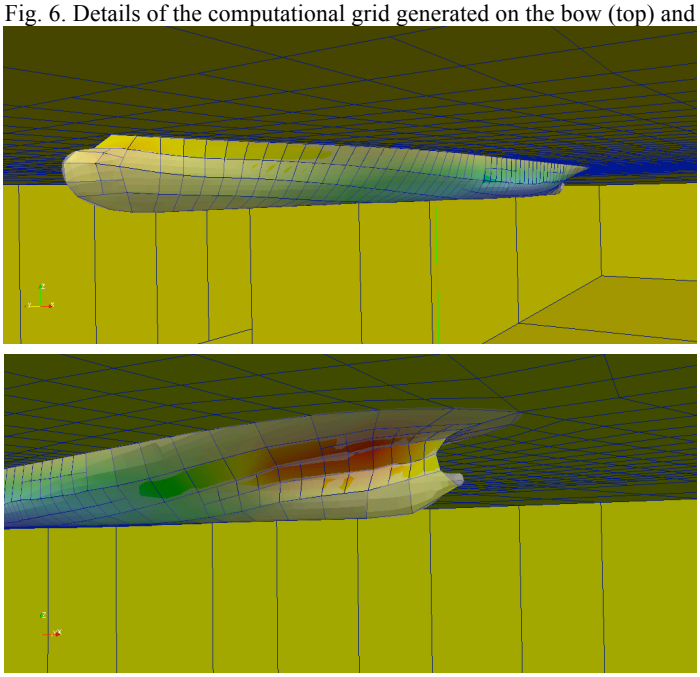
Fig. 5. A local refinement cycle based on center cell distance from CAD surface. The cells denoted by red dots are initially identified for refinement (left), and successively refined (right).

NUMERICAL SIMULATIONS AND RESULTS

The methodology described has been applied to two different modern commercial hull types designed by the Korea Research Institute of Ships and Ocean Engineering (KRISO). The first hull considered is the KRISO Container Ship (KCS), which presents both a bow and stern bulb. The second is the KRISO Very Large Crude-Oil Carrier 2 (KVLCC2), which also has bulbous bow and stern. Several tests have been carried out by KRISO researchers to characterize the behavior of these hulls, and the experimental data have been made available in (Kim, Van and Kim, 2001). For this reason, the KCS and KVLCC2 hulls have become well established benchmarks for ship hydrodynamics simulation tools. Thus, these hulls have been chosen as they represent an ideal test case both for the automatic mesh generation algorithm and for the fluid dynamic solver.

KCS and KVLCC2 grids

The CAD model of the KCS hull is available (in the form of an Iges file) on the website of SIMMAN 2008, the Workshop on Verification and Validation of Ship Maneuvering Simulation Methods. The CAD surfaces are referred to the 1/31.6 scale model tested in the KRISO towing tank, which has length $L = 7.2786$ m and displacement $\Delta = 1.6490$ m^3. As shown in Fig. 1, the model is composed of 57 different unconnected parametric surfaces. This feature, along with the presence of bulbous bow and stern profiles, makes this industrial hull very challenging by the grid generation point of view.
Yet, the automatic mesh generation algorithm, is able to produce a computational grid that is fully satisfactory. Fig. 6 presents details of the mesh generated on the hull surface. All the cells present a quality compatible with the Boundary Element Method used for the simulations. Since the parallel version of the solver has not been implemented, the mesh dimension had to be limited to 3200 nodes. In particular, the hull surface was discretized with 40 nodes in the longitudinal direction, and 4 to 8 nodes in the vertical direction. Despite these limitations, it is possible to notice how the shape of the hull has been well recovered. In particular, the local refinement strategy adopted allowed for a concentration of the nodes on the bow and stern region, which present the highest curvatures.
Fig. 6. Details of the computational grid generated on the bow (top) and



stern (bottom) of the Kriso KCS Hull.

Also the CAD model of the KVLCC2 hull is available as an Iges file on the website of SIMMAN 2008. The CAD surfaces are here referred to the 1/58 scale model tested in the KRISO towing tank, which has length $L = 5.5172$ m and displacement $\Delta = 1.6029$ m^3. The KVLCC2 model is composed of 14 different unconnected parametric surfaces. The mesh has again been limited to 3200 nodes. The hull surface was discretized with 40 nodes in the longitudinal direction, and

4 to 8 nodes in the vertical direction. Also in this case, the automatically generated grid is perfectly suitable for the fully nonlinear potential flow solver implemented. A detail of the KVLCC2 hull surface mesh is presented in Fig. 7.

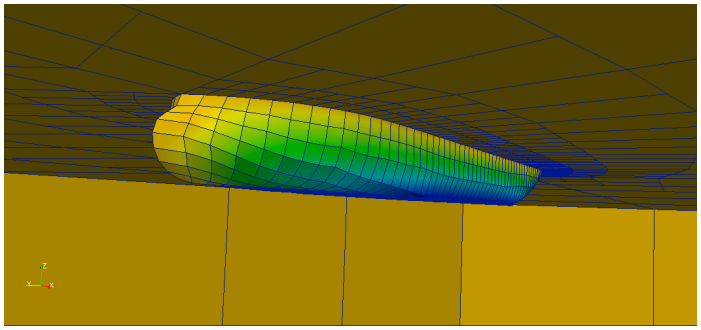


Fig. 7. A detail of the computational grid generated on the bow of the Kriso KVLCC hull.

Numerical simulations and Results

The KCS model speed of the towing tank test was set to $V_\infty = 2.1964$ m/s, which corresponds to $\mathrm{Fr} = 0.26$. As for the KVLCC2 test, the velocity was $V_\infty = 1.047$ m/s, corresponding to $\mathrm{Fr} = 0.142$. As already mentioned, the parallel version of the solver has not been implemented yet, so each simulation ran for about 30 hour on a single Intel Xeon E5530, 2.40GHz processor.

The comparison between the wave profile along the hull surface of the KCS is presented in Fig. 8. By a qualitative standpoint, the solver appears able to reproduce in a correct way the wave elevation field on the hull. As is the case for the experimental values, roughly two principal wavelengths are observed along the hull surface. The first crest is located at $x/L = 0.474$, closely matching the experimental value $x_{\mathrm{exp}}/L = 0.475$. The second crest is located at $x/L = 0.034$, while the experimental value is $x_{\mathrm{exp}}/L = 0.05$. The horizontal positions of the first and second troughs are respectively $x/L = -0.207$ and $x/L = 0.238$, against corresponding experimental values of $x_{\mathrm{exp}}/L = -0.150$ and $x_{\mathrm{exp}}/L = 0.250$. Thus, the numerical simulation troughs positions are shifted slightly upstream, with respect to the experimental locations.

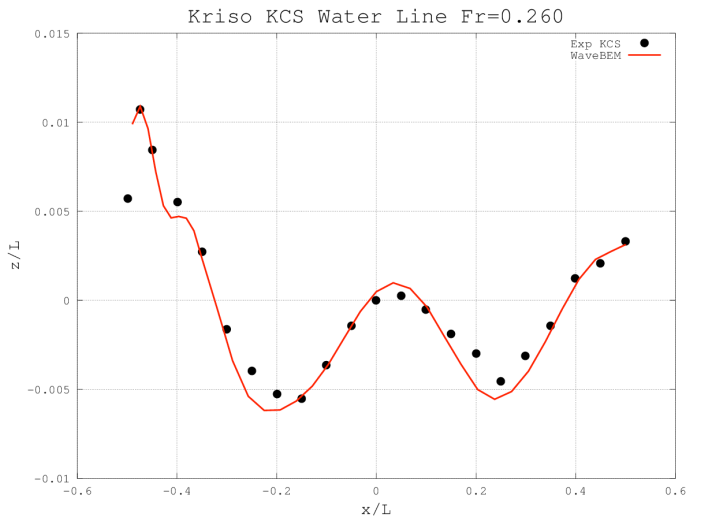


Fig. 8. Comparison between predicted (red line) and experimental (black dots) nondimensionalized wave profile on the hull of the KCS.

Both the height of the wave crests and wave troughs is recovered with accuracy in the simulations. In particular, the wave elevation of the first wave crest appears extremely close to the experimental values. The same can be said for the water elevation in correspondence with the transom stern. The magnitude of the troughs and of the second crest is instead slightly overestimated by the numerical simulation. This might be related to the lack of a dissipation term in the potential model employed. Finally, at about $x/L = -0.41$, the numerical elevation curve presents an inflection point which is not found in the correspondent experimental curve. This could be due to the fact that at this Froude number, the flow is close to wave breaking conditions on the first crest. This could make the semi-Lagrangian boundary conditions employed inaccurate in proximity of the crest. Thus, a possible improvement tot he model cold be represented by the implementation of a breaking wave suppression model, as that illustrated by Muscari and Di Mascio (2003).

The comparison between the wave profile along the hull surface of the KVLCC2 is instead presented in Fig. 9. Even for this hull model, the solver appears able to reproduce correctly way the wave elevation field. Given the lower Frounde number, the wave pattern generated by KVLCC2 hull can be characterized by a big bow wave followed by much smaller waves. The bow wave is reproduced with good accuracy. Both in the experimental and numerical curve, the peak position corresponds with the bow location, at approximatively $x/L = x_{\mathrm{exp}}/L = -0.5$. Also the numerical wave height is closely matching the experimental value. The flat trough following the bow wave is again reproduced with accuracy in the simulation. In the experiment, the through is shaped as a small plateau between $x_{\mathrm{exp}}/L = -0.425$ and $x_{\mathrm{exp}}/L = -0.351$, whereas the corresponding numerical values are respectively $x/L = -0.436$ and $x/L = -0.372$. As for the second trough the predicted position is $x/L = 0.285$, while the experimental one is $x_{\mathrm{exp}}/L = 0.249$.

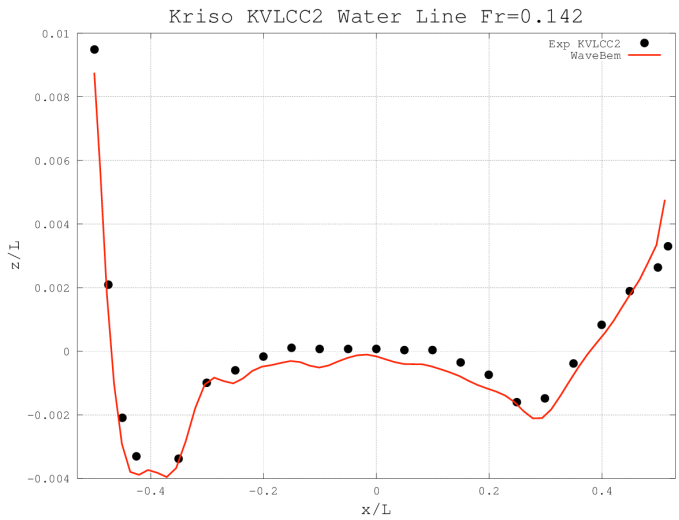


Fig. 9. Comparison between predicted (red line) and experimental (black dots) nondimensionalized wave profile on the hull of KCVLCC2.

Small amplitude and high frequency waves are present in the numerical curve past the first through. Yet, their amplitude seems higher than that of the experimental ones. This could be again related to the non dissipative model adopted. The whole wave elevation curve of the KVLCC2 hull appears shifted slightly downwards with respect to the experimental one. This could be due to a minimal discrepancy between the experimental and the numerical sink. A possible cause for such discrepancy, could be a non completely accurate evaluation of the sink

given the cad surfaces and the assigned displacement. The problem is currently under investigation. A further inaccuracy of the model, seems to appear in proximity of the transom stern, where the water elevation is clearly over predicted. This error might instead be related to the presence of a possible vortical flow region around the transom stern, which could be inaccurately reproduced by the viscous model developed.

CONCLUSIONS AND FURTHER DEVELOPMENTS

This work presented a model for hydrodynamics simulations currently under development at SISSA. The model is based on flow potential theory. On the free surface, the governing Laplace equation is complemented by unsteady fully nonlinear boundary conditions, written in semi-Lagrangian form. The spatial discretization is carried out by a collocation BEM. The resulting DAE system is discretized over time by means of an arbitrary order implicit BDF scheme with adaptive time step.

The fluid dynamic model is directly interfaced with CAD data structures, in order to provide automatic mesh generation and handling functionalities on hulls of arbitrary shape.

Industrially relevant test cases were considered, in order to test both the fluid dynamic model and the mesh generation algorithm. The results confirm the good accuracy of the solver, and the robustness of the CAD interface module.

Future work will consist in several developments to improve the model in many of its aspects.

On one hand, it will be necessary to improve the numerical formulation so as to increase the computational efficiency. A parallel version of the model will be implemented in order to increase the maximum number of nodes used for the simulations. In addition, a Fast Multipole Algorithm under implementation to reduce both computational cost and memory consumption. Significant improvements in the accuracy of the results can be also obtained through the implementation of a iso-geometric BEM based upon the CAD handling module implemented. A part of the current work is headed in this direction.

On the other hand, also the physical model will have to be improved and extended. A first objective is to write stationary version of the DAE system, which will massively reduce the computational cost of each steady state simulation. At the same time, it will be important to complete the unsteady model. To this end a fluid structure interaction, 6 degrees of freedom model for boat motions prediction will be implemented in the next future.

ACKNOWLEDGMENTS

This work was performed in the context of the project OpenSHIP, “Simulazioni di fluidodinamica computazionale (CFD) di alta qualita per le previsioni di prestazioni idrodinamiche del sistema carena-elica in ambiente OpenSOURCE”, supported by Regione FVG - POR FESR 2007-2013 Obiettivo competitivita regionale e occupazione.

BIBLIOGRAPHY

Bangerth W, Hartmann R and Kanschat G (2007) “deal.II—a general purpose object oriented finite element library”, *ACM Trans. Math. Software*, 33(4): 24/1–24/27.

Bangerth W, Heister T, Heltai L, Kanschat K, Kronbichler M, Maier M, and Young TD (2013) "The deal.II Library, Version 8.1" arxiv:1312.2266v4.

Beck, RF (1994) “Time-domain computations for floating bodies”, *Appl. Ocean Res.*, 16:267–82.

DeSimone A, Heltai L and, Manigrasso C (2009) “Tools for the Solution of PDEs Defined on Curved Manifolds with the deal.II Library”, *Technical Report SISSA,* 42/2009/M.

Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, et al. (2005) “Sundials: suite of nonlinear and differential/algebraic equation solvers”, *ACM Trans. Math. Software*, 31(3):363–96.

Hughes TJR and Brooks A (1979) “A multidimensional upwind scheme with no crosswind diffusion”, *Finite Elem. Methods Convection Dominated Flows*, 34:19–35.

Kim, WJ, Van, SH and Kim, DH (2001) “Measurement of flows around modern commercial ship models”, *Experiments in Fluids,* 31 567–578

Lachat JC, Watson JO (1976) “Effective numerical treatment of boundary integral equations: a formulation for three-dimensional elastostatics”, *Int. J. Numer. Methods Eng.*, 10(5):991–1005.

Mola, A, Heltai, L and DeSimone, A (2013) “A stable and adaptive semi-Lagrangian potential model for unsteady and nonlinear ship-wave interactions”. *Engineering Analysis with Boundary Elements*, 37(1):128–143.

Muscari, R and Di Mascio, A (2003) “A local model for the simulation of two–dimensional spilling breaking waves”, *J. Marine Science and Technology*, Vol.8, n.2, pp.61-67.

Tezduyar TE (2003) “Computation of moving boundaries and interfaces and stabilization parameters”, *Int. J. Numer. Methods Fluids*, 43(5):555–75.