Proceedings of the Twenty-seventh (2017) International Ocean and Polar Engineering Conference **www.isope.org**
San Francisco, CA, USA, June 25-30, 2017

ISBN 978-1-880653-97-5; ISSN 1098-6189

A hybrid solver based on efficient BEM-potential and LBM-NS models: recent BEM developments and applications to naval hydrodynamics

Amin Mivehchi[1]*,Jeffrey C. Harris*[2]*,Stéphan T. Grilli*[1]*,Jason M. Dahl*[1]*,Chris M.O'Reilly*[1,3]*,Konstantin Kuznetsov*[2] *and Christian F. Janssen*[4]
(1) Department of Ocean Engineering, University of Rhode Island, Narragansett, RI, USA
(2) LHSV, Ecole des Ponts, CEREMA, EDF R&D,University of Paris-Est,Chatou, France
(3) Navatek Ltd., South Kingstown, RI, USA
(4) Fluid Dynamics and ship Theory Inst.,Hamburg University of Technology (TUHH), Germany

ABSTRACT

We report on recent developments of a 3D hybrid model for naval hydrodynamics based on a perturbation method, in which velocity and pressure are decomposed as the sum of an inviscid flow and a viscous perturbation. The far- to near-field inviscid flows are solved with a Boundary Element Method (BEM), based on fully nonlinear potential flow theory, accelerated with a fast multipole method (FMM), and the near-field perturbation flow is solved with a Navier-Stokes (NS) model based on a Lattice Boltzmann Method (LBM) with a LES modeling of turbulent properties. The BEM model is efficiently parallelized on CPU clusters and the LBM model on massively parallel GPGPU co-processors.

The hybrid model formulation and its latest developments and implementation, in particular, regarding the improvement and validation of the model for naval hydrodynamics applications, are presented in a companion paper by O'Reilly et. al (2017), in this conference. In this paper, we concentrate on the BEM model aspects and show that the BEM-FMM can accurately solve a variety of problems while providing a nearly linear scaling with the number of unknowns (up to millions of nodes) and a speed-up with the number of processors of 35-50%, for small (e.g., 24 cores) to large (e.g., hundreds of cores) CPU clusters.

INTRODUCTION

The simulation of the dynamic response of maritime structures in waves and wave-induced forces is typically based on linear wave models, such as AEGIR (Kring et al.,1999), or in case of large motions and/or steep waves, on using nonlinear wave models based on potential flow theory (PFT), usually solved with a higher-order Boundary element method (BEM). For structures with a forward speed, semi-empirical corrections are often made to account for viscous/turbulent effects in the total resistance. While standard Computational Fluid Dynamics (CFD) models based on the full Navier-Stokes (NS) equations can also be used to simulate such problems, their computational cost is typically too prohibitive and their accuracy for long-term wave modeling usually less than that of PFT-BEM models. However, in some cases, the viscous/turbulent flow around the structure's hull and possible breaking waves and wakes require to be more accurately modeled to capture the salient physics of the problem.

Here, instead of using a CFD-NS method to solve the complete problem, we present a fully three-dimensional (3D) hybrid method for solving the hydrodynamic problem based on perturbation method, in which the total velocity and pressure fields are decomposed into inviscid and viscous (perturbation) parts (e.g., Alessandrini, 2007; Grilli, 2008; Harris and Grilli, 2012). Further, in the hybrid model, the perturbation flow component is only solved in the near-field, using a NS model based on a Lattice Boltzmann Method (LBM; e.g., d'Humieres et al., 2002; Janssen, 2010; Janssen et al., 2010) with Large Eddy Simulation (LES) of the turbulence (e.g., Krafczyk et al., 2003); and the far- to near-field inviscid flow component is solved with a BEM model, based on Fully Nonlinear PFT (FNPF). The latter model is also referred to as a "Numerical Wave Tank" (NWT; Grilli et al., 2001), since it has the typical functionalities of a physical wave tank (i.e., wave generation, propagation, and absorption). The free surface representation in the LBM is based on a VOF method, with piecewise linear interpolation (PLIC) (e.g., O'Reilly et al., 2015) and in the NWT on an explicit time updating.

More specifically, in the hybrid model, the NWT solution is computed over the entire domain, for the incident wave field, including diffraction around the structure and radiation due to its possible motions (for floating-surface piercing structures). The NWT solution results are then used to force the LBM solution, which as indicated is only computed in the near-field of the marine structures, in a domain overlapping with that of the NWT. Hence, the hybrid approach can be much more computationally efficient than traditional CFD solutions, in which the NS solver must be applied to the entire domain.This concept had already been demonstrated for instance by Reliquet et al.(2014), based on different types of numerical models; see O'Reilly et al. (2016) and O'Reilly et al. (2017) in this conference, for details of the hybrid model characteristics and efficiency.

The LBM has proved to be accurate and efficient for simulating a variety of complex fluid flow and fluid-structure interaction problems and, when implemented on a massively parallel General Purpose Graphical Processor Unit (GPGPU) co-processor, it has also been shown to achieve very high efficiency (over 100 million node updates per second on a single GPGPU; e.g., Janssen, 2010 ; Janssen et al., 2013 ; Banari et al.,2014). In this respect, LBM developments in this work are based on the highly efficient, GPGPU-accelerated, Lattice Boltzmann solver

ELBE (Janssen et al., 2015; www.tuhh.de/elbe), developed at the Hamburg University of Technology (TUHH), which features various LBM models, an on-device grid generator, higher-order boundary conditions, and the possibility of specifying overlapping nested grids. ELBE also includes the earlier LBM perturbation model based on Janssen et al.(2010) approach. Simple validations of the hybrid LBM and hybrid LBM-LES approaches, for viscous and turbulent oscillatory boundary layers, were reported by O'Reilly et al. (2015), Janssen et al. (2016), and in greater details in O'Reilly et al. (2017) at this conference.

For the NWT part of the hybrid model, which is the focus of this paper, we use a 3D-BEM-FNPF model based on the same approach as the wave model of Grilli et al. (2001), which was successful at modeling many wave phenomena, including landslide generated tsunamis, rogue waves, surface effect ships, and the initiation of wave breaking caused by bathymetry (also see Grilli et al., 2010). For the types of applications considered in earlier work, Grilli et al. were able to use structured grids made of quadrilaterals, which enabled simpler approaches for setting up higher-order (cubic) elements. In order to tackle more complex geometries and grids, as well as accommodating surface-piercing fixed or floating bodies, an implementation of Grilli et al.'s model for unstructured triangular grids was developed in recent years e.g., Harris et al., 2014). Additionally, the efficiency of the BEM solution in the model for large grids was improved by using a parallelized Fast Multipole Method (FMM; Greengard and Rokhlin,1987), that was efficiently implement on large computer clusters (Harris et al., 2016) [Note, Grilli et al.'s NWT was accelerated with less efficient scalar FMM; Grilli et al., 2010].

The NWT was initially validated for wave propagation as well as radiation and diffraction from vertical cylinders (Harris et al., 2016). More recent improvements were made to increase the accuracy of such results, for instance, when computing the internal solution within the NWT domain (which is required for coupling to the LBM models). Indeed, the numerical integration of the BEM boundary integrals typically has increasingly large errors for points approaching the domain boundary, such as the free-surface or a maritime structure's hull. Adaptively subdividing the integration over BEM elements in this case (as proposed by Grilli and Subramanya (1994) in 2D and extended by Guyenne and Grilli (2006) in 3D), allows maintaining a constant accuracy of the solution throughout the domain. An adaptive integration method was implemented in the new BEM-FMM-NWT, which also allowed for a more accurate solution near corners and considering elements with larger aspect ratios.

Initially, to represent the solution at corners/edges located at intersections between various part of the BEM boundary (e.g., sidewall-bottom, free surface-structure, sidewall-free surface,...), the NWT used a multiple-node representation in which, following Grilli and Svendsen (GSV; 1990) and Grilli and Subramanya (GSU; 1996), individual nodes were specified on the various intersecting boundaries, having the same coordinates but different outwards normal vectors. Individual BEM equations were expressed at all nodes of a multiple-node, and continuity conditions were specified for the velocity potential in order to have a non-singular algebraic BEM system (see GSV for details of corner continuity conditions at double-nodes, for a variety of 2D Dirichlet-Neumann problems). To more accurately solve for the flow near strongly moving solid structures intersecting the free surface, such as a wavemaker, GSU formulated and implemented extended compatibility at double nodes in their 2D BEM model, where they also specified that, besides a continuous potential, the flow velocity vector should also be unique at double nodes. In this paper, we extend the latter method to the multiple-nodes occurring in our 3D-BEM NWT, in various mixed boundary condition cases and assess its accuracy through systematic numerical benchmarking.

MATHEMATICAL AND NUMERICAL MODEL

Governing Equations and Boundary Conditions

The 3D-BEM-NWT assumes an incompressible, inviscid and irrotational flow represented by a velocity potential $\phi(\mathbf{x},t)$, in Cartesian coordinates, with $\mathbf{x}=(x,y,z)$ and z pointing vertically upward. The governing equation is a Laplace's equation for the velocity potential,

$$\nabla^2\phi = 0 \quad , \quad \text{with} \quad \mathbf{u}=\nabla\phi \quad \text{in} \quad \Omega(t), \tag{1}$$

expressed over the domain $\Omega(t)$ with boundary $\Gamma(t)$, where $\mathbf{u}$ is the velocity vector. Using Green's second identity, this equation is solved as a Boundary Integral Equation (BIE),

$$\alpha(\mathbf{x}_l)\phi(\mathbf{x}_l) = \int_{\Gamma(\mathbf{x}(t))} \left\{ \frac{\partial\phi}{\partial n}(\mathbf{x})G(\mathbf{x},\mathbf{x}_l) - \phi(\mathbf{x})\frac{\partial G}{\partial n}(\mathbf{x},\mathbf{x}_l) \right\} \mathrm{d}\Gamma(\mathbf{x}(t)) \tag{2}$$

in which $\mathbf{x}=(x,y,z)$ and $\mathbf{x}_l=(x_l,y_l,z_l)$ are points on the boundary, $\mathbf{n}=(n_x,n_y,n_z)$ is the unit outward normal vector on the boundary, and $\alpha(\mathbf{x}_l)$ is a geometric coefficient function of the interior angle of the boundary at $\mathbf{x}_l$. In this BIE, the 3D free space Green's function and its normal derivative are defined as,

$$G(\mathbf{x},\mathbf{x}_l) = \frac{1}{4\pi r_l} \quad \text{and} \quad \frac{\partial G}{\partial n} = -\frac{\mathbf{r}_l\cdot\mathbf{n}}{4\pi r^3} \tag{3}$$

in which $r_l=|\mathbf{r}_l|=|\mathbf{x}-\mathbf{x}_l|$ is the distance between any point $\mathbf{x}$ from collocation point $\mathbf{x}_l$, both on the boundary $\Gamma(t)$.

On the free surface $\Gamma_f(t)$,ϕ satisfies the nonlinear kinematic and dynamic boundary conditions,

$$\frac{D\mathbf{R}}{Dt} = \mathbf{u} = \nabla\phi \qquad \text{on} \quad \Gamma_f \tag{4}$$

$$\frac{D\phi}{Dt} = -g\zeta + \frac{1}{2}\nabla\phi\cdot\nabla\phi - \frac{p_a}{\rho} \qquad \text{on} \quad \Gamma_f \tag{5}$$

respectively, with $\mathbf{R}$ the position vector of nodes on the free surface, g the acceleration due to gravity, ζ the vertical elevation of the free surface (referred to $z=0$, denoting the still water level), p_a the atmospheric pressure, ρ the fluid density, and the material derivative being defined as,

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u}.\nabla \tag{6}$$

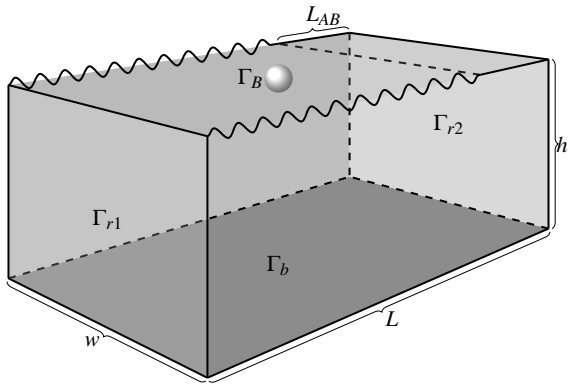

Fig. 1: Definition sketch of NWT computational domain for wave interaction with a rigid body (in this paper a Karman-Trefftz foil) (length L by width w by depth h). The no flow condition has been defined for the bottom (Γ_b) and lateral (Γ_{r2}) boundaries; waves are generated on the leftward boundary (Neumann boundary condition for known velocity and acceleration) and are damped on the far end of the NWT over an absorbing beach (AB) of length L_{AB}.

In the NWT, waves can be generated by simulating a piston wavemaker motion on the "open sea" boundary of the computational domain, $\Gamma_{r1}(t)$. In this case, wavemaker motion and velocity are specified over the wavemaker paddle as,

$$\overline{x} = x_p; \qquad u_p = -\nabla\phi\cdot\mathbf{n} = -\overline{\frac{\partial\phi}{\partial n}} \qquad \text{on} \quad \Gamma_{r1}(t) \tag{7}$$

respectively, where the overline denotes a specified value (see Grilli et al., 2001 for detail). Along the stationary bottom Γ_b and on other fixed boundaries Γ_{r2}, a no-flow condition is prescribed as,

$$\overline{\frac{\partial\phi}{\partial n}} = 0 \qquad \text{on} \quad \Gamma_b,\ \Gamma_{r2} \tag{8}$$

The boundary condition along a rigid surface piercing maritime structure, which moves with velocity $\mathbf{V}_B$ would be define as,

$$\overline{\frac{\partial\phi}{\partial n}} = \mathbf{V}_B\cdot\mathbf{n} \qquad \text{on} \quad \Gamma_B \tag{9}$$

To prevent wave reflection at open boundaries of the NWT domain an *absorbing beach* (AB) is specified by adding terms: $-\nu(x)\zeta$ and $-\nu(x)\phi$ to the right side of the kinematic and dynamic boundary conditions, respectively, where $\nu = 0$ for all of the domain except for points with abscissa $x \geq x_{AB}$, where $\nu(x) = \nu_0((x-x_{AB})/L_{AB})^2$ (Grilli and Horrillo, 1997).

BIE for internal velocities

Based on the BIE (2), another BIE can be derived that directly computes the internal velocity as a function of boundary values of the velocity potential and its normal derivative,

$$\mathbf{u} = \nabla\phi(\mathbf{x}_i) = \int\left[\frac{\partial\phi}{\partial n}(\mathbf{x})Q(\mathbf{x},\mathbf{x}_i) - \phi(\mathbf{x})\frac{\partial Q}{\partial n}(\mathbf{x},\mathbf{x}_i)\right]d\Gamma \tag{10}$$

where $\mathbf{x}_i$ is a point inside domain Ω and (Guyenne and Grilli, 2006),

$$\begin{aligned} Q(\mathbf{x},\mathbf{x}_i) &= \frac{\mathbf{r}_i}{4\pi r_i^3} \\ \frac{\partial Q}{\partial n}(\mathbf{x},\mathbf{x}_i) &= \frac{1}{4\pi r_i^3}\left\{\mathbf{n} - 3(\mathbf{r}_i\cdot\mathbf{n})\frac{\mathbf{r}_i}{r_i^2}\right\} \end{aligned} \tag{11}$$

Eqs. (10) and (11) are explicit (i.e., they do not include any new unknown), and can be evaluated with the same discretization as that used to compute the the boundary solution with BIE (2) (see next Section). However, as pointed out by Guyenne and Grilli (2006), these are potentially hypersingular equations for $\mathbf{r}_i \to 0$, as they have highly varying kernels for very small $\mathbf{r}_i$ values as compared to the equivalent boundary element length. Hence, as indicated in introduction, an adaptive integration method was implemented; details are given later.

Boundary discretization and standard algebraic system

The BIE (2) is discretized and solved by a BEM, using N_Γ collocation nodes (equal to the number of unknowns) and M_Γ boundary elements, defined over boundary $\Gamma(t)$. In this NWT, in past work, linear isoparametric triangular and quadrangular elements were mostly used, with spline elements being used in a more limited way; future applications, however, will make increasing use of the latter more accurate elements. Given the BEM discretization, BIE (2) is transformed into a sum of integrals over each element, which are computed by numerical integration after transforming each element k, of boundary Γ_e^k, from the physical space to a standard 2D reference element of domain $\Gamma_{\xi,\eta}$ defined with curvilinear coordinates (ξ,η), by way of a Jacobian matrix $\mathbf{J}^k$. Polynomial shape functions $N_j(\xi,\eta)$, which interpolate both the geometry and field variables over each isoparametric elements, are defined over the reference element. Therefore the integrals in Eq. (2) read,

$$\begin{aligned} &\int_{\Gamma(x)}\frac{\partial\phi}{\partial n}(\mathbf{x})G(\mathbf{x},\mathbf{x}_l)d\Gamma = \sum_{k=1}^{M_\Gamma}\int_{\Gamma_e^k}\frac{\partial\phi}{\partial n}(\mathbf{x})G(\mathbf{x},\mathbf{x}_l)d\Gamma \\ &= \sum_{j=1}^{N_\Gamma}\left\{\sum_{k=1}^{M_\Gamma}\int_{\Gamma_{\xi,\eta}} N_j(\xi,\eta)G(\mathbf{x}(\xi,\eta),\mathbf{x}_l)|\mathbf{J}^k(\xi,\eta)|\,d\xi d\eta\right\}\frac{\partial\phi}{\partial n}(\mathbf{x}_j) \\ &= \sum_{j=1}^{N_\Gamma}\left\{\sum_{k=1}^{M_\Gamma} D_{lj}^k\right\}\frac{\partial\phi_j}{\partial n} \\ &= \sum_{j=1}^{N_\Gamma} K_{lj}^d\frac{\partial\phi_j}{\partial n} \end{aligned} \tag{12}$$

and,

$$\begin{aligned} &\int_{\Gamma(x)}\phi(\mathbf{x})\frac{\partial G}{\partial n}(\mathbf{x},\mathbf{x}_l)d\Gamma = \sum_{k=1}^{M_\Gamma}\int_{\Gamma_e^k}\phi(\mathbf{x})\frac{\partial G}{\partial n}(\mathbf{x},\mathbf{x}_l)d\Gamma \\ &= \sum_{j=1}^{N_\Gamma}\left\{\sum_{k=1}^{M_\Gamma}\int_{\Gamma_{\xi,\eta}} N_j(\xi,\eta)\frac{\partial G}{\partial n}(\mathbf{x}(\xi,\eta),\mathbf{x}_l)|\mathbf{J}^k(\xi,\eta)|\,d\xi d\eta\right\}\phi(\mathbf{x}_j) \\ &= \sum_{j=1}^{N_\Gamma}\left\{\sum_{k=1}^{M_\Gamma} E_{lj}^k\right\}\phi_j \\ &= \sum_{j=1}^{N_\Gamma} K_{lj}^n\phi_j \end{aligned} \tag{13}$$

in which $l = 1,...,N_\Gamma$, D_{lj}^k and E_{lj}^k denote the *local* Dirichlet and Neumann element matrices, and K_{lj}^d, K_{lj}^n the corresponding *global* (assembled) matrices, respectively; note that j refers to local nodal values of element k, but is expressed in the global node numbering, by way of assembling.

Eqs. (10) and (11) yield the algebraic form of Eq. (2) as,

$$\alpha_l\phi_l = \sum_{j=1}^{N_\Gamma}\left\{K_{lj}^d\frac{\partial\phi_j}{\partial n} - K_{lj}^n\phi_j\right\} \tag{14}$$

Boundary conditions are expressed with Eqs. (7) to (9) and the final algebraic system is assembled by moving unknowns to the left hand side and knowns to the right hand side of the equation (see Grilli et al., 2001 for a detailed description of the method.)

Evaluating Eqs. (10) and (11) matrix terms D_{lj}^k and E_{lj}^k requires integrating complex kernels over each boundary element k, which become singular when $r_l \to 0$ in the Green's functions. For triangular elements, the weakly singular integrals are desingularized using Dunavant's (1984) rule, and for quadrangular elements by way of a tensor product of Gauss integration. For linear triangular elements, singular integrals can then be analytically integrated. Although analytical solutions of non-singular integrals exist for linear triangular elements, to allow using the same formulation for higher-order elements, we compute these integrals numerically; and likewise for quadrangular elements (e.g., Grilli et al. 2001, 2010). Coefficients α in the BIE are found by applying the rigid mode method (e.g., Grilli et al. 1989), which expresses that for a Dirichlet problem with a homogeneous $\phi = 1$ value specified over the entire boundary, the discretized BIE solution must yield $\partial\phi/\partial n = 0$; the α coefficients are then found as the residuals of this Dirichlet problem. The discretized algebraic BEM system is solved with BiCGSTAB, a Krylov iterative solver.

In the non-singular integrals, as the free space Green's function Eq. (3) varies rapidly when collocation point l is specified close to the considered element (i.e., $r_l \to 0$), an adaptive integration technique is used, both

for collocation points belonging to the boundary discretization or for internal points where the internal velocity is computed with Eqs. (10) and (11). The method used is similar to that described by Grilli et al.(2001) for the same purpose, but here we consider a simpler distance criterion: when the point under consideration is closer to the center of the element than twice the maximum element edge length, the element is recursively divided into four smaller elements, and this recursive process is done up to 16 times.

Fast Multipole Method

In the FMM, the free space Green's function is approximated for "distant" points by a truncated (order P) multipole expansion,

$$G(\mathbf{x},\mathbf{x}_l) \approx \sum_{m_x=m_y=m_z=0}^{m_x+m_y+m_z=P} \frac{(x-x_l)^{m_x}(y-y_l)^{m_y}(z-z_l)^{m_z}}{m_x!\,m_y!\,m_z!} \left\{ \left(\frac{\partial}{\partial x}\right)^{m_x} \left(\frac{\partial}{\partial y}\right)^{m_y} \left(\frac{\partial}{\partial z}\right)^{m_z} G(\mathbf{x},\mathbf{x}_l) \right\} \quad (15)$$

In this approach, both interactions that are "distant enough" are neglected (yielding a sparse algebraic system matrix) and the full system matrix of the BIE does not have to be assembled and solved, which is typically one of the most time consuming part of the NWT solution, as it has a $O(N_\Gamma^2)$ numerical complexity. More specifically, to decide how to approximate (or even neglect) interactions, the FMM uses a divide-and-conquer strategy based on the distance between two points. Importantly, by assigning intermediate points (e.g., at the centers of groups of nodes or elements of the boundary mesh) and applying the binomial theorem, one is able to manipulate multipole coefficients that only need to be computed once, instead of directly evaluating the BIE between each element and node (see Harris et al., 2016 for details).

Theoretically, if efficiently implemented and assuming N_Γ is more than a few thousand, the computational time of the FMM should scale with $O(N_\Gamma)$ or so. This is much faster than the solution of the complete BEM system with the best iterative solvers (which is $O(N_\Gamma^2)$). Several variations have been proposed for the implementation of the FMM on parallel clusters (Yokota, 2013), which generally rely on domain decomposition, whereby the FMM is first applied on each processor over some region of space and then results are combined. Harris et al. (2016) implemented this domain decomposition parallel FMM approach and studied the scaling of the two main phases of the FMM-BEM, using an iterative solver : (i) the assembling of the global system (sparse) matrix, and (ii) the matrix-vector products involved in the FMM. They showed good scaling of the FMM-BEM assembling and matrix-vector products for a mesh with quadrangular elements and $N_\Gamma = 79,202$ nodes, with varying numbers of CPUs, up to a few hundreds, over a simple parallelipipedical domain, typical of modeling of nonlinear waves (e.g., Fig. 1). The grid was partitioned into 1,024 sub-domains and a 15th-order FMM expansion was used, with 100 integration points were used on each element.

While this approach permits good scaling, up to hundreds of processors and a billion unknowns (Yokota et al. 2011), for BEM problems solved on small desktop computer clusters, that may only have $O(10^5)$ unknowns and less than 100 processors, a simpler approach referred to as single-level FMM was found to scale much more efficiently. This was pointed out by Waltz et al. (2007), who compared this approach to many other parallelization attempts of the FMM-BEM and showed that this is due to the fact that while the number of unknowns is large enough for the FMM to be efficient, the number of unknowns per processor is low. The FMM scaling of the present NWT will be studied later in applications.

Curvilinear coordinate transformation

A local non-orthogonal curvilinear coordinate system is used, following Fochesato et al. (2005), to represent the geometry of higher-order BEMs, and field variables and their derivatives at each collocation node on the boundary, which extends the orthogonal coordinate assumption made by Grilli et al. (2001). Thus, at any point $\mathbf{x}$ within a n-node isoparametric BEM element k, the geometry and local non-orthogonal unit tangential vectors are defined as,

$$\mathbf{x} = \sum_{j=1}^{n} N_j(\xi,\eta)\mathbf{x}_j^k \quad \text{and} \quad \mathbf{s} = \frac{\partial \mathbf{x}}{\partial \xi} \Big/ \left|\frac{\partial \mathbf{x}}{\partial \xi}\right|, \quad \mathbf{m} = \frac{\partial \mathbf{x}}{\partial \eta} \Big/ \left|\frac{\partial \mathbf{x}}{\partial \eta}\right| \quad (16)$$

where $\mathbf{x}_j^k$ are the element k nodal coordinates. The unit vectors of a corresponding local orthogonal coordinate system $(\mathbf{s},\mathbf{m}',\mathbf{n})$ are then defined as,

$$\mathbf{m}' = \frac{1}{\sqrt{1-\kappa^2}}\mathbf{m} - \frac{\kappa}{\sqrt{1-\kappa^2}}\mathbf{s} \qquad \text{with} \qquad \kappa = \mathbf{s}\cdot\mathbf{m} \quad (17)$$

the cosine of the angle between the unit tangential vectors. The normal vector to the $(\mathbf{s},\mathbf{m})$ plane (pointing outwards depending on proper number of element nodes) completes this orthogonal coordinate system and can be calculated as,

$$\mathbf{n} = \mathbf{s} \times \mathbf{m}' \quad (18)$$

The Jacobian of the transformation between element k, in the global coordinate system, to the reference element is defined as,

$$|\mathbf{J}^k(\xi,\eta)| = \left\{ \left|\frac{\partial \mathbf{x}}{\partial \xi}\right|, \left|\frac{\partial \mathbf{x}}{\partial \eta}\right|, \mathbf{n} \right\}, \quad (19)$$

which can be computed at any point $\mathbf{x}$ of element k with Eqs. (14-16).

Similarly, the gradient of the velocity potential, i.e., flow velocity, is expressed as (Fochesato et al., 2005),

$$\nabla\phi = \frac{1}{1-\kappa^2}\left(\frac{\partial\phi}{\partial s} - \kappa\frac{\partial\phi}{\partial m}\right)\mathbf{s} + \frac{1}{1-\kappa^2}\left(\frac{\partial\phi}{\partial m} - \kappa\frac{\partial\phi}{\partial s}\right)\mathbf{m} + \frac{\partial\phi}{\partial n}\mathbf{n} \quad (20)$$

which can also be computed based on Eqs. (14-16), assuming the tangential and normal derivatives of the potential are known. The normal derivatives of the potential are obtained from BIE (2)'s solution and the computation of tangential derivatives is detailed below.

As for the geometry, the potential ϕ over an isoparametric element k is defined as the sum of nodal values multiplied by shape functions defined over the reference element,

$$\phi = \sum_{j=1}^{n} N_j(\xi,\eta)\phi_j^k \quad (21)$$

The tangential derivatives of the potential over each element (i.e., tangential velocities) can thus be calculated by direct differentiation of Eq. (21) with the operators,

$$\frac{\partial}{\partial s} = \frac{\partial}{\partial \xi} \Big/ \left|\frac{\partial \mathbf{x}}{\partial \xi}\right| \qquad \text{and} \qquad \frac{\partial}{\partial m} = \frac{\partial}{\partial \eta} \Big/ \left|\frac{\partial \mathbf{x}}{\partial \eta}\right| \quad (22)$$

Hence, the tangential derivatives at point $\mathbf{x}_l = \mathbf{x}(\xi_l,\eta_l)$ of element k read,

$$\frac{\partial\phi}{\partial s} = \sum_{j=1}^{n} \frac{\partial N_j(\xi,\eta)}{\partial \xi} \frac{1}{\left|\frac{\partial \mathbf{x}^k}{\partial \xi}\right|} \phi_j^k = \sum_{j=1}^{n} \mathscr{S}_j^k \phi_j^k \quad (23)$$

$$\frac{\partial\phi}{\partial m} = \sum_{j=1}^{n} \frac{\partial N_j(\xi,\eta)}{\partial \eta} \frac{1}{\left|\frac{\partial \mathbf{x}^k}{\partial \eta}\right|} \phi_j^k = \sum_{j=1}^{n} \mathscr{M}_j^k \phi_j^k \quad (24)$$

Higher-order derivatives of the geometry and the potential on the boundary can be defined in the same manner, in the local orthogonal coordinate system (see Fochesato et al., 2005).

Treatment of corners in global system matrix

As mentioned in the introduction, following Grilli et al. (1990, 1996, 2001), corners and edges of the BEM discretization, which mark intersections of different parts of the boundary of the computational domain, in general have different normal directions and boundary conditions. These are represented by multiple nodes, for which multiple BIEs are expressed based on different values of the normal derivative of the potential, but using a single value of the potential, as the latter must be unique (i.e., continuous) at the same location. This condition is enforced in the BEM solution by modifying all but one of the assembled algebraic equations for each multiple node, to satisfy a *potential continuity* condition. This assumes that initially (for $t = 0$) the boundary conditions at all corners are prescribed in a way that does not cause a mathematical singularity in the problem.

In a 3D space there are both *double* (e.g., between the wavemaker and the free surface) and *triple* (e.g., where the wavemaker and free surface boundaries meet with a sidewall boundary) nodes. Thus, for double/triple nodes, there are 2/3 discretized BIEs expressed at each multiple node. To ensure uniqueness and well-posedness of the solution, however, and in particular a single potential at any given location, 1 (in case of a double node) or 2 (in case of a triple node) of these BIEs must be modified in the final algebraic system to ensure that both the global matrix is not singular and the solution yields a single (continuous) potential. For the simple domain shown in Fig. 1, the multiple nodes can be categorize based on their boundary conditions as : (i) Dirichlet-Neumann (DN) double nodes (e.g., wakemaker and free surface boundary); (ii) Neumann-Neumann (NN) double nodes (e.g., bottom and sidewall boundaries); (iii) Neumann-Neumann-Neumann (NNN) triple nodes (e.g., where the wavemaker, bottom and sidewall boundaries meet); and (iv) Drichlet-Neumann-Neumann (DNN) (e.g., where the wavemaker, free surface and sidewall boundaries meet).

In a NWT with a moving free surface, possibly moving maritime structure(s), and a moving wavemaker, Grilli and Subramaniya (1996) showed in their 2D work that, to ensure a stable and accurate solution near multiple-nodes, the velocity vector should also be unique at such nodes, particularly on the free surface. Besides achieving an accurate solution, this will also ensure that individual nodes on the free surface, that are part of multiple nodes, move to an identical location through time updating. These authors indeed showed that if velocity is not explicitly *enforced* to be unique at multiple nodes, by modifying the algebraic BIE system in a proper way, large numerical errors will occur at and near such nodes in the BIE solution, which will grow even larger through time updating and, eventually, lead to instability of the solution, particularly close to strongly moving rigid boundaries.

Hence, following Grilli and Subramanya (1996), in the 3D-NWT, we extended the simple potential continuity condition at multiple nodes used in earlier work to also enforce uniqueness of the velocity (i.e., gradient of the potential) at individual nodes of multiple nodes. This was done for all cases of mixed boundary conditions (e.g., DN, NN, NNN, DNN) by replacing all but one of the assembled equations of a multiple node in the algebraic system, by a so-called velocity *compatibility* condition, which also includes the potential continuity condition; in the following, we only present one example for a DN double-node case. These extended multiple-node conditions, in fact, make the representation of the solution compatible (i.e., consistent) on both sides of a corner and effectively eliminate the occurrence of (numerical) singularities in the discretized solution. When using compatibility conditions at corners when solving mixed boundary value problems in simple rectangular domains, Grilli and Subramaniya (1996) showed that numerical errors at corners could be reduced to almost arbitrarily small values in their 2D-NWT.

For instance, at a DN double node located at the intersection between a piston wavemaker and the free surface, the compatibility condition forced the BIE solution to compute a (corrected) value of the tangential velocity on the free surface node $l = f$ (of the double node), $\partial\phi_f/\partial s$, as a function of both the normal velocity $\partial\phi_f/\partial n$ obtained from the solution of the BIE (13) at the current time step, and the (specified) wavemaker velocity on the wavemaker node $l = p$ (of the double node), $\partial\phi_p/\partial n = -u_p$ (see Eq. (7)). Here, expressing this compatibility condition at a similar DN double nodes, defined at the intersection between the free surface and a wavemaker/maritime structure boundary in the 3D-NWT, we specify that $\mathbf{u}_p = \mathbf{u}_f$, i.e.,

$$\overline{\frac{\partial\phi_f}{\partial s}}\mathbf{s}_f + \overline{\frac{\partial\phi_f}{\partial m'}}\mathbf{m'}_f + \frac{\partial\phi_f}{\partial n}\mathbf{n}_f = \frac{\partial\phi_p}{\partial s}\mathbf{s}_p + \frac{\partial\phi_p}{\partial m'}\mathbf{m'}_p + \overline{\frac{\partial\phi_p}{\partial n}}\mathbf{n}_p \tag{25}$$

in which the overlines indicate known/specified values. Note that the tangential derivatives of the specified potential on the free surface (Dirichlet boundary) are computed by way of the tangential derivative operators defined in Eqs. (23) and (24), assuming $l = f$ or $l = p$.

Moving the unknowns to the left-hand-side and projecting the equation in the direction of unit vector $\mathbf{i}$, we find,

$$\begin{aligned} - & \frac{\partial\phi_p}{\partial s}(\mathbf{s}_p\cdot\mathbf{i}) - \frac{\partial\phi_p}{\partial m'}(\mathbf{m'}_p\cdot\mathbf{i}) + \frac{\partial\phi_f}{\partial n}(\mathbf{n}_f\cdot\mathbf{i}) = \\ - & \overline{\frac{\partial\phi_f}{\partial s}}(\mathbf{s}_f\cdot\mathbf{i}) - \overline{\frac{\partial\phi_f}{\partial m'}}(\mathbf{m'}_f\cdot\mathbf{i}) + \overline{\frac{\partial\phi_p}{\partial n}}(\mathbf{n}_p\cdot\mathbf{i}) = \overline{u^i} \end{aligned} \tag{26}$$

Assuming that element k is on the free surface boundary and element m is on the wavemaker/maritime structure boundary, and replacing the tangential derivatives into Eq. (26), while specifying the potential continuity condition $\phi_1^m = \overline{\phi_1^k}$, we find

$$\begin{aligned} & -\sum_{j=2}^{n}\frac{\{\mathscr{S}_j^m - \kappa_m\mathscr{M}_j^m\}C_{s_pi} + \{\mathscr{M}_j^m - \kappa_m\mathscr{S}_j^m\}C_{m_pi}}{1-\kappa_m^2}\phi_j^m + \frac{\partial\phi_f}{\partial n}C_{n_fi} = \\ & -\overline{\sum_{j=1}^{n}\frac{\{\mathscr{S}_j^k - \kappa_k\mathscr{M}_j^k\}C_{s_fi} + \{\mathscr{M}_j^k - \kappa_k\mathscr{S}_j^k\}C_{m_fi}}{1-\kappa_k^2}\phi_j^k + \overline{\frac{\partial\phi_p}{\partial n}}C_{n_pi}} \\ & +\overline{\frac{\{\mathscr{S}_1^m - \kappa_m\mathscr{M}_1^m\}C_{s_pi} + \{\mathscr{M}_1^m - \kappa_m\mathscr{S}_1^m\}C_{m_pi}}{1-\kappa_m^2}\phi_1^k} = \overline{u'^i} \end{aligned} \tag{27}$$

in which the C coefficients are the cosines of the angles between the unit vectors indicated as lower indices. For a DN wavemaker boundary, such as considered in this example, direction $\mathbf{i}$ can be individually selected at each double node, as the horizontal projection of the local normal vector to the wavemaker boundary $\mathbf{n}_p$. Note that in Eq. (27) there are n unknowns in the left-hand side, $n-1$ potential values at the non-multiple nodes of element m on the wavemaker boundary and 1 normal derivative of the potential on the free surface node of the multiple node in element k; the right-hand side of the equation $\overline{u'^i}$ is built with specified or known values at the current time. Eq. (27) is substituted for 1 of the two BIE equations assembled in the BEM algebraic system for this DN double node.

In case of orthogonal elements (i.e., when $\kappa = 0$) and with $\mathbf{m}\cdot\mathbf{i} = 0$ for 2D problems, Eq. (27) simplifies to the equation developed by Grilli and Subramanya (1996) and used as extended compatibility condition in their 2D-NWT.

For triple nodes, such as DNN, for instance at the intersection between a wavemaker $l = p$, free surface $l = f$, and sidewall $l = q$ boundary, 2 equations such as Eq. (25) would be derived, expressing $\mathbf{u}_p = \mathbf{u}_f = \mathbf{u}_q$, leading to 2 equations similar to Eq. (27), based on projections of each of these in directions $\mathbf{i}$ and $\mathbf{j}$, for instance pointing in the direction of the horizontal projections of the local normal vectors to the wavemaker and sidewall boundaries, respectively ($\mathbf{n}_p$ and $\mathbf{n}_q$). These 2 equations would then replace 2 of the 3 BIE equations assembled in the BEM algebraic

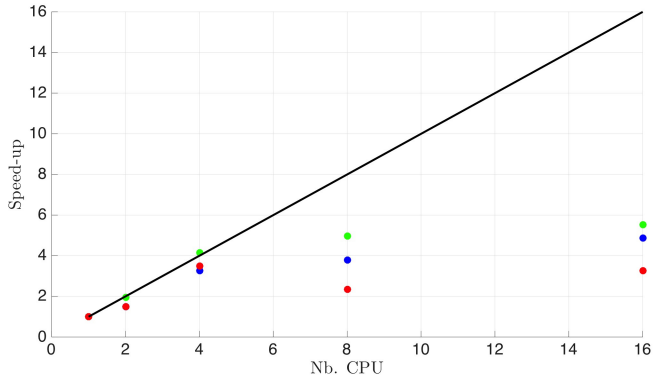

Fig. 2: Speed-up of the 3D-FMM-BEM solution of a mixed-boundary condition Laplace problem over the domain of Fig. 1, as a function of the number of CPUs on a small desktop cluster, for: (i) matrix assembling (●); (ii) internal velocity calculation (as a vector product sample) (●); and (iii) complete solution (●). A grid of quadrangular linear elements is used with $N_\Gamma = 54,000$ collocation points.

N_Γ	MAX with CC	RMS($\frac{\partial\phi}{\partial n}$) with CC	RMS(ϕ) with CC	MAX without CC	RMS($\frac{\partial\phi}{\partial n}$) without CC	RMS(ϕ) without CC
54	5.2984e-07	5.2938e-07	5.7738e-07	1.8131e-05	8.3901e-06	5.2273e-07
150	8.2993e-06	5.4940e-06	4.4905e-07	4.8857e-05	2.0582e-05	4.8828e-07
726	6.8341e-06	1.51491r-06	2.7939e-07	1.0558e-05	1.0558e-05	3.5162e-07
2,646	4.3483e-06	1.1496e-06	2.6854e-07	6.9098e-05	1.9098e-05	3.4782e-07
10,086	1.7760e-06	1.1326e-06	2.3885e-07	3.1012e-05	1.3773e-05	3.2742e-07

Table 1: Numerical errors (maximum (MAX); and root-mean-square (RMS) of the 3D-NWT solution over a unit size cube domain, with and without multiple-node compatibility conditions (CC), as a function of the number of nodes N_Γ.

system for this DNN triple node. Doing this, in the BEM solution, the value of the normal velocity at the free surface node of the triple node would depend upon both distributions of potential along the 2 intersecting Neumann boundaries.

APPLICATIONS

Scaling of the FMM-BEM solution on parallel CPU clusters

We performed the same scaling study as Harris et al. (2016), but on a small desktop CPU cluster with shared memory, of the solution of a mixed boundary condition Laplace problem over the domain of Fig. 5; the domain was discretized with $N_\Gamma = 54,000$ nodes, and the problem solved with 1,2,4,8 or 16 CPUs. In the FMM, the grid was partitioned into 32 sub-domains, and 10 integration points were used on each element. Fig. 2 shows the speed-up of the system matrix assembly, total 3D-FMM-BEM solution, and internal velocity, computational time as a function of the CPU time on a single core. We see that on this small system, while a significant speed up, almost optimal, of the complete solution is achieved up to 4 CPUs, the marginal gain in speed-up is much smaller when further increasing the number of CPUs. This is related to the internal architecture and CPU to CPU communications within the small desktop cluster.

Next, on the same system and for the same Laplace problem, we studied the scalability of the 3D-FMM-BEM complete solution for 1 and 8 CPUs, as a function of the number of nodes $N_\Gamma = 5,000$ to 100,500. Results in Fig. 3 show an $O(N^{1.05})$ scaling for 1 and $O(N_\Gamma^{1.09})$ scaling for 8 CPUs, which both are quite close to the optimal theoretical scaling $O(N_\Gamma)$.

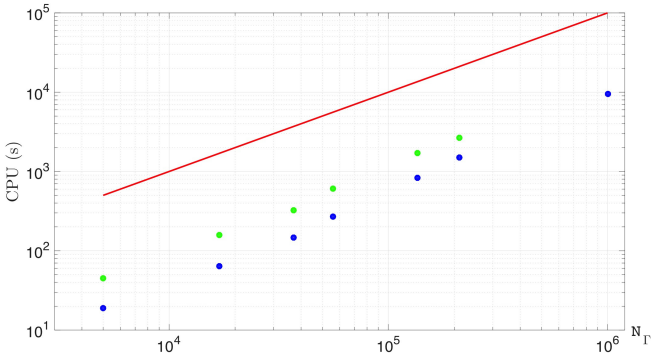

Fig. 3: Same case as 2. CPU time of the 3D-FMM-BEM solution for $N_\Gamma = 5,000$ to 100,500 nodes, using 1 (●) and 8 (●) CPUs on a small desktop cluster. The red line shows an $O(N_\Gamma)$ scalability, whereas it is $O(N_\Gamma^{1.05})$ for 1 CPU.

Compatibility conditions at multiple nodes

To assess the accuracy of the new velocity compatibility conditions (CC) implemented at multiple nodes in the 3D-NWT, we performed a convergence test of the solution of a mixed boundary value problem over a unit size cube, which has a simple analytical solution, as a function of the number nodes, $N_\Gamma = 54$ to 10,086 (note, 54 nodes is the minimum number for a cube to contain both double and triple nodes).

Both maximum (MAX) and root-mean-square (RMS) errors (over the entire grid) of the solution were computed, with and without compatibility conditions, based on values of the normal velocity $\partial\phi/\partial n$ or potential ϕ. These are listed in Table 1. In all cases, the potential continuity conditions were specified at multiple nodes. Although RMS errors are only slight smaller with compatibility conditions than without, the maximum errors on normal velocity, which occur at multiple nodes, are much reduced when specifying compatibility conditions, particularly for the larger discretization. We verified that differences are much larger between the two methods for a domain with complex geometry and, based on earlier work (Grilli and Subramanya, 1996), we expect these to be even larger once we will be updating the free surface geometry and or the position of wavemaker/maritime structures as a function of time, since this will cause cumulative error effects.

Computation of internal velocities

As discussed in introduction, this work is part of broader project to develop a hybrid solver for naval hydrodynamics problem, based on an coupling an efficient BEM-FNPF model, i.e., the 3D-BEM-FMM NWT discussed here, and a LBM solution of Navier-Stokes (NS) equations, through a perturbation approach. In the hybrid solver, the potential flow solution is used to force the LBM-NS solution for the viscous perturbation flow (see, Harris and Grilli (2012) and O'Reilly et al. (2017), in this conference). This requires computing the internal velocity field at many points within the NWT domain, at each time step.

Here, we validate the computation of the internal velocity field in the 3D-NWT by computing the flow around a symmetric Karman-Trefftz foil (see Abbot and Von Doenhoff, 1959), for a free flow velocity U in the x-direction. Specifying the 3D-NWT boundary conditions to solve a 2D problem (using no-flow conditions on the sidewalls in the y-direction and upper and bottom boundaries in the z-direction), an analytical solution of the flow around the foil can be expressed based on a conformal mapping

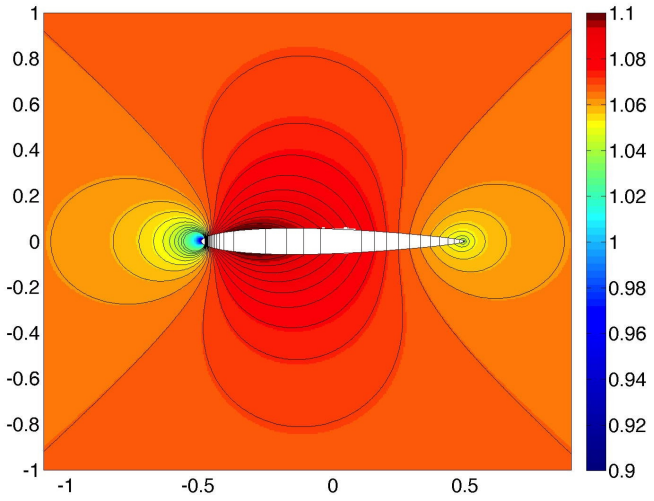

Fig. 4: Analytical solution for the scaled module of velocity $\mathbf{u}/U$ of the (uniform) flow around a symmetric Karman-Trefftz foil.

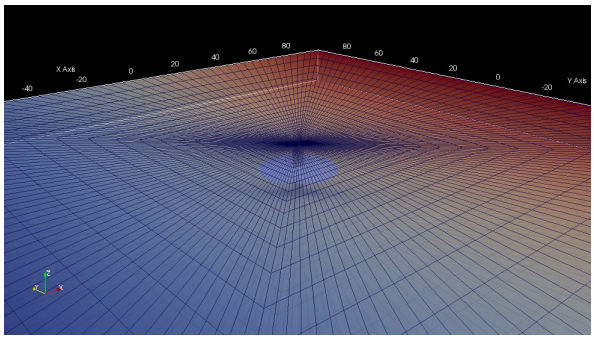

Fig. 5: Computational domain for the computation of the (uniform) flow around a symmetric Karman-Trefftz foil, using $M_\Gamma = 15,488$ linear quadrangular elements.

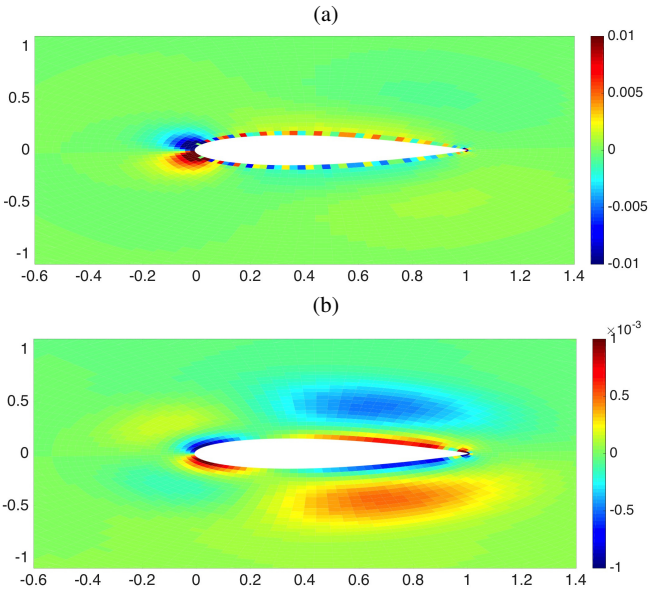

Fig. 6: Case of Fig. 5. Zoom-in on relative errors of velocity module computed using: (a) linear, or (b) B-spline quadrilateral elements.

in the complex plane $\zeta = x + iy$,

$$Z(\zeta) = n\lambda \frac{(1+\frac{\lambda}{\zeta})^n + (1-\frac{\lambda}{\zeta})^n}{(1+\frac{\lambda}{\zeta})^n - (1-\frac{\lambda}{\zeta})^n} \tag{28}$$

which maps a cylinder of radius R centered at $\zeta = x = s$, for which the complex potential is the trivial superposition of a uniform flow of velocity U and a dipole: $W(\zeta) = U(\zeta + R^2/\zeta)$, to the symmetric foil, where $n = 2 - \beta/\pi$, with β the angle at the foil trailing edge, and $\lambda = R + s$. Selecting $R = 1$ and the foil center at $s = -0.045$ we find $\beta = 8^o$. The 2D analytical solution for the scaled module of velocity $\mathbf{u}/U$ around this foil is shown in Fig. 4.

We consider a rectangular domain, with extension $-100 < x < 100$, $-100 < z < 100$, and solve this 2D flow in 3D using a transverse direction y extending from, $-15 < z < 0$. The computational domain boundary is discretized with $M_\Gamma = 15,488$ quadrangular elements with increasing resolution towards the foil (Fig. 5), which are either: (i) linear isoparametric; or (ii) cubic B-spline elements; in the FMM, 15th-order expansions are used in the BEM solution of Eq. (2). Compatibility conditions are specified here at all multiple nodes. We then compute the velocity field with Eq. (10) at 10,000 internal points $\mathbf{x}_i$, most of these being located near the foil surface, and compare it with the analytical solution. In Fig. 6a, we find that for case (i), maximum errors in velocity reach 0.45% near the foil leading and trailing edges, while the L^2-error is 10^{-6} over the entire set of internal points. For case (ii), however, in Fig. 6b, these errors are about 10 times smaller confirming the well known property and importance of using higher-order elements in the BEM.

CONCLUSION

In this paper, we reported on recent improvements in the implementation of a 3D-NWT solving FNPF with a free surface. The NWT is based on a BEM, using linear or cubic B-spline elements, and the solution is accelerated with an efficient FMM, showing nearly linear scaling in both scalar and parallel computations. The NWT is a component of a hybrid model, in combination with a Navier-Stokes LBM model with Large eddy simulation, aimed at solving naval hydrodynamic problems (e.g., ship seakeeping).

We showed, in particular, a good scaling of the FMM-BEM numerical solution with problem size N_Γ near the theoretical optimal ($O(N_\Gamma)$) and reasonable additional speed-up with the number of processors in a parallel implementation. Well-posed velocity compatibility conditions were developed and implemented for multiple nodes at the corners and edges of the 3D domain, which extend earlier 2D formulations by Grilli and Svendsen (1990), Grilli and Subramanya (1996), and Grilli et al. (2001). These were shown to reduce errors in the numerical solution for various discretizations and cases.

We presented results for a uniform flow past a submerged symmetric foil and showed that internal velocities could be both efficiently and accurately computed by applying the same FMM-BEM approach. Such internal velocity fields are used in the hybrid model to force the viscous perturbation LBM solution based on the inviscid flow results in the NWT (e.g., O'Reilly et al., 2016, 2017, the latter paper at this conference).

By extending the NWT formulation to a moving coordinate systems and (later) to arbitrary geometries, we will be able to handle a broader range of more complex applications of particular interest to Naval Hydrodynamics and ocean engineering. The extension of the BEM to higher-order B-spline elements (e.g., Maestre et al. 2016; Harris et al. paper at this conference), which was achieved without fundamental changes in the FMM-NWT formulation, already demonstrated for the foil that numerical errors can be significantly reduced. This will be very important in naval hydrodynamics applications, when modeling submerged or floating bodies of complex geometry.

ACKNOWLEDGEMENTS

A. Mivehchi, S.T. Grilli, J.M. Dahl and C. O'Reilly gratefully acknowledge support for this work from grants N000141310687 and N000141612970 of the Office of Naval Research (PM Kelly Cooper).

REFERENCES

Abbot, I.H., Von Doenhoff A.E. (1959) *Theory of Wing Sections*, Dover Pub., New York.

Alessandrini, B. (2007). *Thèse d'Habilitation en Vue de Diriger les Recherches*. Ecole Centrale de Nantes, Nantes.

Banari, A., Janssen, C.F., and Grilli, S.T. (2014). An efficient lattice Boltzmann multiphase model for 3D flows with large density ratios at high Reynolds numbers. *Comp. and Math. with Applications*, **68**(12), 1819-1843.

Dunavant, D.A. (1984). High degree efficient symmetrical Gaussian quadrature rules for the triangle. *Intl. J. Numerical Meth. Engng.*, **21**, 1129-1148.

Fochesato, C., Grilli, S.T. and Guyenne, P. (2005). Note on non-orthogonality of local curvilinear co-ordinates in a three-dimensional boundary element method.*Intl. J. Num. Meth. Fluids*, **48**, 305-324.

Grilli, S.T. (2008). On the Development and Application of Hybrid Numerical Models in Nonlinear Free Surface Hydrodynamics. Keynote lecture in *Proc. 8th Intl. Conf. on Hydrodynamics* (Nantes, France, 9/08) (P. Ferrant and X.B. Chen, eds.), pps. 21-50.

Grilli, S.T., Dias, F., Guyenne, P., Fochesato, C. and F. Enet (2010). Progress In Fully Nonlinear Potential Flow Modeling Of 3D Extreme Ocean Waves. Chapter 3 in *Advances in Numerical Simulation of Nonlinear Water Waves* (ISBN: 978-981-283-649-6, edited by Q.W. Ma) (Vol. 11 in Series in Advances in Coastal and Ocean Engineering). World Scientific Publishing Co. Pte. Ltd., pps. 75- 128.

Grilli, S.T., Guyenne, P. and Dias, F. (2001). A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom.*ntl. J. Numerical Meth. Fluids*, **35**(7), 829-867.

Grilli, S.T. and Horrillo, J. (1997). Numerical Generation and Absorption of Fully Nonlinear Periodic Waves. *J. Engng. Mech.*, **123**(10), 1060-1069.

Grilli, S.T., Skourup, J. and Svendsen, I.A. (1989). An Efficient Boundary Element Method for Nonlinear Water Waves.*Engng. Anal. Boundary Elements*, **6**(2), 97-107.

Grilli, S.T. and Svendsen, I.A. (1990) .Corner Problems and Global Accuracy in the Boundary Element Solution of Nonlinear Wave Flows. *Engng. Anal. Boundary Elements*, **7**(4), 178-195.

Grilli, S.T. and Subramanya, R. (1994). Quasi-singular Integrations in the Modeling of Nonlinear Water Waves. *Engng. Anal. Boundary Elements*,**13**(2), 181-191.

Grilli, S.T. and Subramanya, R. (1996). Numerical Modeling of Wave Breaking Induced by Fixed or Moving Boundaries. *Comput. Mech.*, **17**(6), 374-391.

Guyenne, P. and Grilli, S.T. (2006). Numerical study of three-dimensional overturning waves in shallow water. *J. Fluid Mech.*, 547, 361-388.

Harris, J.C. and S.T. Grilli (2012). A perturbation approach to large-eddy simulation of wave-induced bottom boundary layer flows. *Intl. J. Numer. Meth. Fluids*, **68**, 1,574-1,604.

Harris, J.C., Dombre, E., Benoit, M. and Grilli , S.T. (2014). A comparison of methods in fully nonlinear boundary element numerical wave tank development. In *Proc. 13th Journées de l'Hydrodynamique* (JH2014) (Val de Reuil, France, Nov. 18-20, 2014), 13 pps.

Harris J.C., Dombre E., Mivehchi A., Benoit M., Grilli S.T. and C. Peyrard (2016). Progress in fully nonlinear wave modeling for wave-structure interaction. In *Proc. 15th Journée de l'hydrodynamique* (JH2016) (November 22-24, Brest, France), 12pps.

d'Humieres D., Ginzburg I., Krafczyk M., Lallemand P., and Luo L.-S. (2002). Multiple Relaxation-Time Lattice Boltzmann models in three-dimensions. *Phil. Trans. Royal Soc. London*, **A360**, 437-451.

Janssen C.F., S.T. Grilli and M. Krafczyk (2010). Modeling of Wave Breaking and Wave-Structure Interactions by Coupling of Fully Nonlinear Potential Flow and Lattice-Boltzmann Models. In *Proc. 20th Offshore and Polar Engng. Conf.* (ISOPE10, Beijing, China, June 20-25, 2010), pps. 686-693.

Janssen, C.F. (2010). *Kinetic approaches for the simulation of non-linear free surface flow problems in civil and environmental engng.*. PhD thesis, Technical Univ. Braunschweig.

Janssen, C.F., S.T. Grilli and M. Krafczyk (2013) On enhanced non-linear free surface flow simulations with a hybrid LBM-VOF approach. *Comp. and Math. with Applications*, **65**(2), 211-229.

Janssen, C.F., D. Mierke, M Überrück, S. Gralher, and T. Rung (2015). Validation of the GPU-accelerated CFD solver ELBE for free surface flow problems in civil and environmental engineering. *Computation*, **3**(3),354-385.

Krafczyk M., Tölke J., and Luo L.-S. (2003). Large eddy simulations with a multiple-relaxation-time LBE model. *Intl. J. Modern Phys.*, **17**, 33-39.

Kring, D.C., Korsmeyer, F.T., Singer, J., Danmeier, D., White, J. (1999). Accelerated nonlinear wave simulations for large structures, in *7th Proc. Intl. Conf. Num. Ship Hydrodyn.*, Nantes, France.

Maestre, J. Cuesta, I. and Pallares, J. (2016). An unsteady 3D Isogeometrical Boundary Element Analysis applied to nonlinear gravity waves. *Compute. Meth. in Appl. Mechanics and Engng.*, **310**, 112-133.

O'Reilly, C.M., Grilli, S.T., Dahl, J.M., Banari, A., Janssen, C.F., Shock, J.J. and M. Uberrueck (2015). Solution of viscous flows in a hybrid naval hydrodynamic scheme based on an efficient Lattice Boltzmann Method. In *Proc. 13th Intl. Conf. on Fast Sea Transportation* (FAST 2015; Washington D.C., September 1-4, 2015).

O'Reilly, C.M., Grilli, S.T., Harris, J.C., Mivehchi, A., Janssen, C.F. and Dahl, J.M. (2016). Development of a hybrid LBM-potential flow model for Naval Hydrodynamics. In *Proc. 15th Journée de l'hydrodynamique* (JH2016) (Nov. 22-24, Brest, France), 15pps.

O'Reilly C.M., S.T. Grilli, J.C. Harris, A. Mivehchi, C.F. Janssen and J.M. Dahl (2017). A Hybrid Solver Based on Efficient BEM-potential and LBM-NS Models: Recent LBM Developments and Applications to Naval Hydrodynamics. In *Proc. 27th Offshore and Polar Engng. Conf.* (ISOPE17, San Francsico, USA. June 2017), 8pps.

Reliquet, G. Drouet, A. Guillerm,P.-E. Gentaz,L. and Ferrant,P. (2014). Simulation of wave-ship interaction in regular and irregular seas under viscous flow theory using the SWENSE method. In *Proc. 30th Symp. Naval Hydrodyn.* (Tasmania, 11/2014), 11 pps.

Yokota, R., Bardhan, J.P., Knepley, M.G., Barba, L.A. and Hamada, T. (2011). Biomolecular electrostatics using a fast multipole BEM on up to 512 GPUs and a billion unknowns.*Comput. Phys. Communications*, **182**, 1272-1283.

Yokota, R. (2013). An FMM based on dual tree traversal for many-core architectures.*J. Algorithms and Comput. Tech.*, **7**, 301-324.

Waltz, C.,Sertel, K., Carr, M.A., Usner, B.C. and Volakis,J.L.(2007). Massively parallel fast multipole method solutions of large electromagnetic scattering problems.*IEEE Trans. Antennas and Propagation*, **55**(6), 1810-1816.