



HAL
open science

PROGRESSIVE WAVE SIMULATION USING STABILIZED EDGE-BASED FINITE ELEMENT METHODS

Renato N Elias, Milton A Gonçalves, Alvaro L G A Coutinho, Paulo T T
Esperança, Marcos a D Martins, Marcos D a S Ferreira

► **To cite this version:**

Renato N Elias, Milton A Gonçalves, Alvaro L G A Coutinho, Paulo T T Esperança, Marcos a D Martins, et al.. PROGRESSIVE WAVE SIMULATION USING STABILIZED EDGE-BASED FINITE ELEMENT METHODS. 28th International Conference on Ocean, Offshore and Arctic Engineering, May 2009, Honolulu, United States. 10.1115/OMAE2009-79641 . hal-01433790

HAL Id: hal-01433790

<https://hal.science/hal-01433790>

Submitted on 13 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

PROGRESSIVE WAVE SIMULATION USING STABILIZED EDGE-BASED FINITE ELEMENT METHODS

**Renato N. Elias, Milton A. Gonçalves Jr.,
Alvaro L. G. A. Coutinho**
Center for Parallel Computations
Federal University of Rio de Janeiro,
P. O. Box 68506,
RJ 21945-970 – Rio de Janeiro, Brazil
{renato, milton, alvaro}@nacad.ufrj.br

Paulo T. T. Esperança
Department of Naval and Ocean Engineering,
Federal University of Rio de Janeiro,
P. O. Box 68506,
RJ 21945-970 – Rio de Janeiro, Brazil
ptarso@peno.coppe.ufrj.br

Marcos A. D. Martins, Marcos D. A. S. Ferreira
PETROBRAS Research Center
Av. Horacio Macedo 850
Rio de Janeiro, Brazil
{marcos.martins, marcos.donato}@petrobras.com.br

ABSTRACT

Flows involving waves and free-surfaces occur in several problems in hydrodynamics, such as sloshing in tanks, waves breaking in ships and motions of offshore platforms. The computation of such wave problems is challenging since the water/air interface (or free-surface) commonly present merging, fragmentation and cusps, leading to the use of interface capturing Arbitrary Lagrangian-Eulerian (ALE) approaches. In such methods the interface between the two fluids is captured by the use of a marking function which is transported in a flow field. In this work we simulate these problems with a 3D incompressible SUPG/PSPG parallel edge-based finite element flow solver associated to the Volume-of-Fluid (VOF) method [1]. The hyperbolic equation for the transport of the marking function is also solved by a fully implicit parallel edge-based SUPG finite element formulation. Global mass conservation is enforced adding or removing mass proportionally to the absolute value of the normal velocity at the interface. The performance and accuracy of the proposed solution method is tested in the simulation of progressive waves and the interaction of a fixed cylinder with a progressive wave.

INTRODUCTION

Flows involving waves occur in several problems in hydrodynamics. Sloshing of liquids in tanks, wave breaking in ships, offshore platforms motions and green water on decks are important examples of this class of problems. The main computational challenge when solving such highly nonlinear problem is determining the evolution of the interface location. There are a large number of numerical methods devoted to the computation of free-surface problems. These methods are frequently classified as interface tracking and interface capturing methods.

Interface tracking methods are based on a Lagrangian framework where the moving interface or boundary is explicitly tracked by the computational grid or by the particles of meshless methods which must be deformed or moved in order to follow the fluid flow. The deforming-spatial-domain/stabilized space time finite element formulation (DSD/SST) proposed by Tezduyar [1] and Tezduyar et al [2, 3] is a mesh based example of interface tracking method. Particle methods, such as those of Koshizuka et al [4] and Violeau and Issa [5] are examples of smoothed particle hydrodynamics (SPH) methods to the simulation of free-surface problems. However, these methods still present a high computational cost since they need to compute the interaction between the particles using search algorithms. As a compromise between the

advantages offered by mesh based and meshless methods, Del Pin et al present in [6] the particle finite element method (PFEM) applied to free-surface flows. In this method the critical parts of the continuum are discretized with particles while the remaining parts are treated by a Lagrangian finite element formulation. Another technique mixing Lagrangian and Eulerian flavors was proposed in [7] by Takizawa et al. In this work the authors enhanced the Constrained Interpolation Profile method (CIP) for solving hyperbolic equations with a meshless Soroban grid. The resulting formulation was used to treat fluid-object and fluid-structure interaction in the presence of free-surfaces.

As a cost effective alternative to interface tracking methods, interface capturing methods have emerged. Interface capturing methods are Eulerian in their concept, thus they rely on a unique and fixed computational grid to capture the interface evolution. In this class of methods the interface is represented by a scalar function which marks the regions filled with the fluids involved. In other words, the interface position is implicitly captured in a scalar marking function value and the interface evolution is determined by the additional cost of solving an advection equation for the marker. As opposed to interface tracking methods, interface capturing methods require little effort to represent all complicated features of moving interfaces. Additionally, the parallel implementation and post-processing of interface capturing methods are straightforward. The main drawback of interface capturing methods is the need to average the fluid properties at the interface cells (elements) due to the discontinuity of the Eulerian representation of the interface. Moreover, the accuracy and computational cost of interface capturing methods are typically associated to grid resolution, properties of the marking function chosen to represent the interface and numerical methods for solving the fluid flow and marking function advection. The well known volume-of-fluid scheme (VOF), firstly proposed by Hirt and Nichols [8] for Cartesian grids, is an interface capturing technique which employs a step function ranging from 0 to 1 to represent the fraction of fluid within the grid cells. In this sense, the interface is implicitly represented by the partially filled cells. The main issues associated to VOF methods include the difficulty in advecting a discontinuous step function and the accurate modeling of surface tension effects. Level set methods [9] implement free-surface flows in a different manner than VOF by changing the marking function employed to represent the interface. Therefore, the fluids are associated to the range of the distance function signs while the interface is implicitly represented by the zero level set. However, the level set method suffers when the distance function loses its properties and must be rebuilt. In fact, the success of level set method lies in its ability of building and keeping a signed distance function without losing its properties. The enhanced-discretization interface-capturing method, firstly proposed by Tezduyar in [10] and the work of Lohner et al presented in [11], are both examples of unstructured grid formulations based on the finite

element method to solve free-surface flows using a VOF marking function.

In this work we use our VOF edge-based stabilized finite element solver [12] and [13], to deal with complex wave phenomena. The main characteristics of our solver are: Streamline-Upwind/Petrov-Galerkin (SUPG), Pressure-Stabilizing/Petrov-Galerkin (PSPG) and Least Squares Incompressibility Constraint (LSIC) stabilized finite element formulation; implicit time marching scheme with adaptive time stepping control; advanced Inexact Newton solvers; edge-based data structures to save memory and improve performance; support to message passing and shared memory parallel programming models; and large eddy simulation extensions using a classical Smagorinsky model.

GOVERNING EQUATIONS

Incompressible Fluid Flow

Let $\Omega \subset \mathbb{R}^{n_{sd}}$ be the spatial domain, where n_{sd} is the number of space dimensions. Let Γ denote the boundary of Ω . We consider the following velocity-pressure formulation of the Navier-Stokes equations governing the incompressible flow of two immiscible fluids within an Arbitrary Lagrangian-Eulerian frame [14]:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{u}_{mesh}) \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{on } \Omega \times [0, t_f] \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega \times [0, t_f] \quad (2)$$

where ρ and \mathbf{u} are the density and velocity, \mathbf{u}_{mesh} is the mesh velocity, \mathbf{f} is the body force vector carrying the gravity acceleration effect and $\boldsymbol{\sigma}$ is the stress tensor given as

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + \mathbf{T} \quad (3)$$

where p is the pressure, \mathbf{I} is the identity tensor, \mathbf{T} is the deviatoric stress tensor

$$\mathbf{T} = 2\mu\boldsymbol{\epsilon}(\mathbf{u}) \quad (4)$$

and $\boldsymbol{\epsilon}(\mathbf{u})$ is the strain rate tensor defined as

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (5)$$

In the present work a large eddy simulation (LES) approach to turbulence is considered by the use of a classic Smagorinsky turbulence model [15]. In this model, the viscosity μ is

augmented by a subgrid-scale viscosity μ_{SGS} proportional to the norm of the local strain rate tensor and to a filter width h defined here as the cubic root of the element volume,

$$\mu_{SGS} = \rho (C_s h)^2 |\mathbf{2}\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})| \quad (6)$$

where C_s is the Smagorinsky constant, taken as 0.1.

The essential and natural boundary conditions associated with equations (1) and (2) can be imposed at different portions of the boundary Γ and represented by,

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_g \quad (7)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h} \quad \text{on } \Gamma_h \quad (8)$$

where Γ_g and Γ_h are complementary subsets of Γ .

Let us assume that we have some suitably defined finite-dimensional trial solution and test function spaces for velocity and pressure, S_u^h , V_u^h , S_p^h and $V_p^h = S_p^h$. The finite element formulation of equations

(1) and (2) using SUPG and PSPG stabilizations for incompressible fluid flows can be written (see Tezduyar and Osawa [1]) as follows: Find $\mathbf{u}^h \in S_u^h$ and $p^h \in S_p^h$ such that

$\forall \mathbf{w}^h \in V_u^h$ and $\forall q^h \in V_p^h$:

$$\begin{aligned} & \int_{\Omega} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + (\mathbf{u}^h - \mathbf{u}_{mesh}^h) \cdot \nabla \mathbf{u}^h - \mathbf{f} \right) d\Omega \\ & + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) d\Omega - \int_{\Gamma} \mathbf{w}^h \cdot \mathbf{h} d\Gamma + \int_{\Omega} q^h \nabla \cdot \mathbf{u}^h d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \frac{1}{\rho} \left[\tau_{SUPG} \rho (\mathbf{u}^h - \mathbf{u}_{mesh}^h) \cdot \nabla \mathbf{w}^h + \tau_{PSPG} \nabla q^h \right] \\ & \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + (\mathbf{u}^h - \mathbf{u}_{mesh}^h) \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) - \rho \mathbf{f} \right] d\Omega^e \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{LSIC} \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h d\Omega^e = 0 \end{aligned} \quad (9)$$

In the above equation the first four integrals on the left hand side represent terms that appear in the Galerkin formulation of the problem (1)-(8), while the remaining integral expressions represent the additional terms which arise in the stabilized finite element formulation. Note that the stabilization terms are evaluated as the sum of element-wise integral expressions,

where n_{el} is the number of elements in the mesh. The first summation corresponds to the SUPG term and the second to the PSPG term. We have evaluated the SUPG and PSPG stabilization parameters according to Tezduyar *et al.* [2], as follows:

$$\tau_{SUPG} = \tau_{PSPG} = \left[\left(\frac{2 \|\mathbf{u}^h - \mathbf{u}_{mesh}^h\|}{h} \right)^2 + 9 \left(\frac{4\nu}{h^2} \right)^2 \right]^{-1/2} \quad (10)$$

here \mathbf{u}^h is the local velocity vector, and ν is the kinematic viscosity.

In Equation (9), the last summation is the Least Squares Incompressibility Constraint (LSIC) term [3], added to prevent spurious oscillations at high Reynolds number flows. The LSIC stabilization parameter is,

$$\tau_{LSIC} = \frac{\|\mathbf{u}^h - \mathbf{u}_{mesh}^h\| h}{2} \quad (11)$$

The discretization of Equation (9) leads us to a nonlinear system of equations to be solved at each time step.

Interface Capturing

In volume-of-fluid (VOF) method [8], a scalar marking function can be employed to capture the position of the interface between the fluids by simply using the fluids fraction relationship.

The volume-of-fluid can be stated as: assuming the value 1 in regions filled with fluid A, e.g., water, and the value 0 in regions filled with fluid B, e.g., air, the position of the fluid interface will be defined by the isovalue contour $\phi(\mathbf{x}, t) = \phi_c$, where $\phi_c \in [0, 1]$. The value $\phi_c = 0.5$ is usually assumed.

Finally, the function $\phi(\mathbf{x})$ is driven by a velocity field \mathbf{u} satisfying the following transport equation, given in conservative form as,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot ((\mathbf{u} - \mathbf{u}_{mesh}) \phi) = 0 \quad (12)$$

In the volume-of-fluid (VOF) formulation the fluid density and viscosity, employed in the fluid flow solution, are interpolated across the interface as follows:

$$\rho = \phi(\mathbf{x}, t) \rho_B + [1 - \phi(\mathbf{x}, t)] \rho_A \quad (13)$$

$$\mu = \phi(\mathbf{x}, t) \mu_B + [1 - \phi(\mathbf{x}, t)] \mu_A \quad (14)$$

where subscripts A and B denote the values corresponding to each fluid.

The finite element formulation of Equation (12) can be written as follows: Find $\phi^h \in S_\phi^h$, such that, $\forall w^h \in V_\phi^h$,

$$\begin{aligned} & \int_{\Omega} w^h \left(\frac{\partial \phi^h}{\partial t} + (\mathbf{u}^h - \mathbf{u}_{mesh}^h) \cdot \nabla \phi^h \right) d\Omega \\ & \quad + \int_{\Omega} w^h (\nabla \cdot \mathbf{u}^h) \phi^h d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{SUPG} \mathbf{u}^h \cdot \nabla w^h \left(\frac{\partial \phi^h}{\partial t} + (\mathbf{u}^h - \mathbf{u}_{mesh}^h) \cdot \nabla \phi^h + (\nabla \cdot \mathbf{u}^h) \phi^h \right) d\Omega^e \\ & \quad + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta \nabla w^h \cdot \nabla \phi^h d\Omega = 0 \end{aligned} \quad (15)$$

where S_ϕ^h and V_ϕ^h are standard test and weight finite element spaces. The first two integrals represent the Galerkin formulation of Equation (12), while the first element-wise summation represents the SUPG and the second summation term is the nonlinear discontinuity-capturing term, useful when sharp gradients and/or boundary layers are present in directions other than the streamlines [16]. The evaluation of τ_{SUPG} and δ stabilization terms follow the definitions described in [17] and [16] respectively. The discretization of Equation (15) leads us to a non-linear ordinary differential equation system, due to the discontinuity-capturing term. In this work, we have adopted the $YZ\beta$ discontinuity capturing term as proposed in [18] where δ is computed as:

$$\delta(\phi^h) = \left(\frac{h_e}{2} \right)^\beta \left| \bar{\phi}^{-1} R^e(\phi) \right| \left(\sum_{j=1}^3 \left| \bar{\phi}^{-1} \frac{\partial \phi^h}{\partial x_j} \right|^2 \right)^{\frac{\beta-1}{2}} \quad (16)$$

where $R^e(\phi)$ is the element residual of Equation (15)

$$R^e(\phi^h) = \frac{\partial \phi^h}{\partial t} + \nabla \cdot ((\mathbf{u}^h - \mathbf{u}_{mesh}^h) \phi^h) \quad (17)$$

Note that if $\beta = 1$ and the reference value $\bar{\phi} = 1$, the $YZ\beta$ discontinuity capturing term renders to the Consistent Approximated Upwind (CAU) method [16]. In order to avoid non physical results, values lying outside the range [0,1] were truncated with the following function:

$$\phi^h = \min[\max[\phi^h, 0], 1] \quad (18)$$

SOLUTION PROCEDURE

The computational solution kernels consist of predictor multiscorrector time integration schemes as described in [17], [19] for both incompressible fluid flow and interface transport equations. The generalized trapezoidal rule is employed in the time discretization. The nonlinearities due to the convective term on the Navier-Stokes equation are treated by an Inexact Newton-GMRES algorithm as described in Elias et al [20]. In this solution algorithm, at the beginning of the nonlinear iterations in each time step, the algorithm computes large linear tolerances, producing fast nonlinear steps. As the iterations progress towards the solution, the inexact nonlinear method adapts the GMRES tolerances to reach the desired accuracy. A nodal-block diagonal preconditioner is employed for the flow and a simple diagonal preconditioner for the marker. Moreover, for both the fluid flow and the marking function advection we use an adaptive time stepping procedure based on a Proportional-Integral-Derivative (PID) controller (see [21]) for further details). Most of the computational effort spent in this solution procedure is due to the matrix-vector products within the GMRES driver for both flow and marker. To improve the computational efficiency with respect to standard element-by-element and sparse matrix-vector storage schemes, we adopt an edge-based data structure in order to minimize indirect memory addressing, diminish floating point operation counts (flops) and memory requirements, as described in Elias et al [22] and Coutinho et al [23] for both the Navier-Stokes equations and the marking function advection. Further computational gains are obtained from data preprocessing performed by the EdgePack library – a package to improve cache reutilization based on reordering and grouping techniques [24]. All computational kernels are parallelized covering most of distributed and shared memory systems as well as vectorized and pipelined processors [25].

Enforcing Mass Conservation

The most challenging feature for a good interface capturing method resides in its ability to preserve the mass of the species involved. According to Lohner [11], in VOF methods the mass loss can be associated to reasons such as: interface smearing due to numerical diffusion of the step function, inexact divergence free velocity field and boundary conditions, undulations in the solution of the marking function advection. Level set methods suffer when the marking function loose its signed distance property. These problems have been reported by many authors and are still the subject of several researches. In this work we have followed the procedure proposed in [11] and [12] to overcome mass losses. In this procedure the mass lost/gained are found by comparing the expected value, composed by the initial mass plus the inlet and outlet fluxes, at the end of each time step. Therefore, the values to be added or removed are made proportional to the absolute value of the normal velocity of the interface given by

$$u_n = \left| \left(\mathbf{u}^h - \mathbf{u}_{mesh}^h \right) \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right| \quad (19)$$

the amount computed from Equation (19) guarantees that the mass correction will act mainly in regions where the interface is moving faster, while keeping the stationary regions untouched. Therefore, the portion of mass correction corresponding to each element is computed and projected onto the global nodes by L_2 projection. In [12], Elias and Coutinho evaluate different test problems with the help of the mass preservation algorithm showing its efficiency.

In order to guarantee the effectiveness of the mass correction method we must have an accurate procedure for computing the volume of the fluid phases. In this work we have adopted a method, described in [12], which makes use of the interface explicitly determined for those elements lying on the interface. To introduce the method, firstly we need to identify the elements as filled, partially filled or empty. This can be done by computing an element key according to the nodal value of the marking function ϕ . Thus, if $\phi \geq 0.5$ we assign a value 1, otherwise, it is marked as 0. Finally, the element key is found by summing its nodal values. Note that the empty elements can be skipped from the computation since the void region will be the difference between the domain and the filled region. Moreover, for elements partially filled, the filling volume can be found from the sub-tetrahedra and wedges formed by the interface plane cutting the original element.

TEST PROBLEMS

Progressive Wave

This case represents the simulation of generation and propagation of a progressive regular wave, from a piston-type wave maker. The computational domain is presented in Figure

1. It has 120 meters length, 20 meters height and 0.5 meters width. Figure 1 also presents the initial condition and initial mesh configuration.

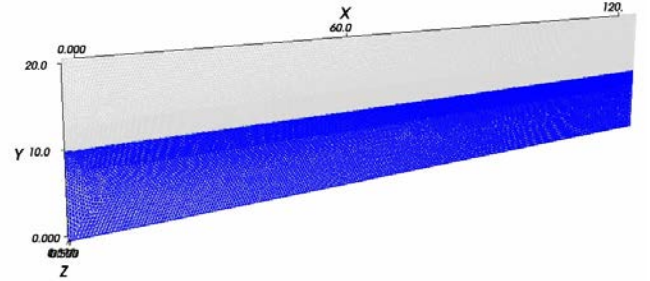


Figure 1. Initial condition and mesh – 67,320 nodes and 254,052 tetrahedral elements.

The movement is prescribed by a periodic function at the left face, simulating a piston-type wave maker. To avoid mesh entangling we introduce a buffer region where the periodic function describing the nodes displacement suffers a damping effect, given by,

$$x = A e^{-\alpha_0} \cos(\omega t + \varphi) \quad (20)$$

where A is the piston displacement amplitude, x and x_0 are position coordinates, ω is the piston angular frequency, t refers to the time and φ is the phase angle. In this problem, the piston moves periodically, with amplitude 0.40 m from its original position and the oscillation period is 4 s. All nodes, from the piston up to 20 m, also oscillate gradually according to Equation (20).

The time step was fixed in 0.005 seconds and the simulation time is 40 seconds. Linear GMRES tolerance was set to 10^{-3} and 10^{-5} for the Navier-Stokes equations and scalar marking function equation, respectively. The Navier-Stokes Inexact-Newton flow solver converges in 3 iterations while the scalar marking function equation converges in 6 iterations.

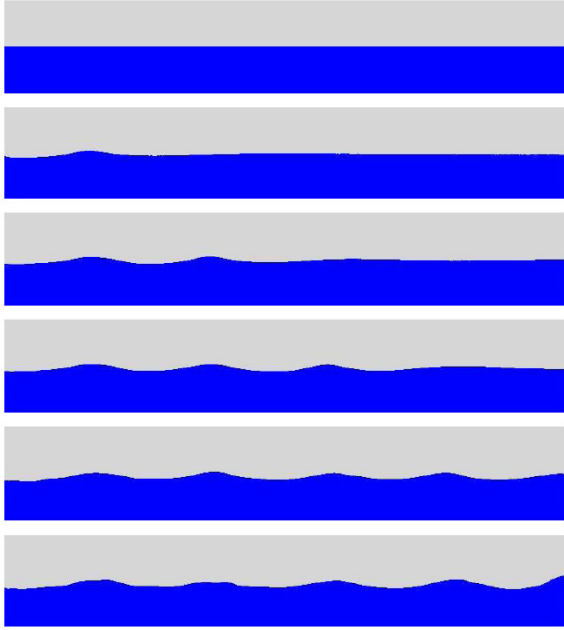


Figure 2. Generation and propagation of a progressive wave. Free surface elevation at $t = 0s$, $t = 8s$, $t = 16s$, $t = 24s$, $t = 32s$ and $t = 40s$.

According to linear wave theory, the wave celerity can be computed as the ratio between its length and period [26]. The analytical and numerical results for the wave celerity are 6.17 m/s and 6.42 m/s, which are in good agreement. Beyond the wave celerity, other comparison parameters were verified. The wave length and height provided by the linear wave generator theory ([27] and [28]) are 24.67 m and 1.49 m, respectively. They compare well with the values obtained from the simulation, 24.70 m and 1.58 m.

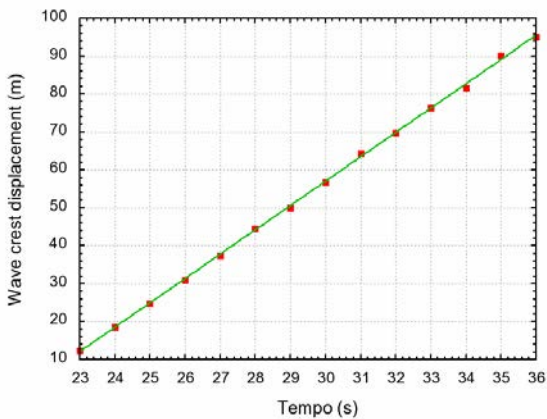


Figure 3. Progressive wave displacement.

Three positions were chosen to track pressure and velocity fields. Figure 4 shows the monitored points.

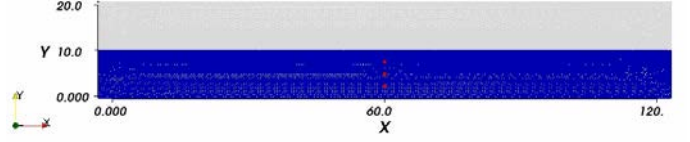


Figure 4. Monitored points. $x = 60m$ and $y = 2.5m, 5.0m, 7.5m$.

The piston is in the origin of coordinate system. The points are at x coordinate 60.0 meters and y coordinates 2.5, 5.0 e 7.5 meters.

The analytical solution was calculated from equations given by the potential formulation for regular progressive wave propagation, known as linear theory [26]. The velocity components and pressure equations are shown below:

$$u = \zeta_a \omega \frac{\cosh[k(h+y)]}{\sinh(ky)} \cos(kx - \omega t) \quad (21)$$

$$v = \zeta_a \omega \frac{\sinh[k(h+y)]}{\sinh(ky)} \sin(kx - \omega t) \quad (22)$$

$$p = -\rho g y + \rho g \zeta_a \frac{\cosh[k(h+y)]}{\cosh(ky)} \cos(kx - \omega t) \quad (23)$$

In equations 21, 22 e 23, u and v are the velocity components, p is pressure and x and y are position coordinates. The wave amplitude and frequency are ζ_a e ω , respectively. Water depth is h and k is the wave number. The wave number is calculated from dispersion relationship, given by equation 24 .

$$k \tanh(kh) = \frac{\omega^2}{g} \quad (24)$$

where g denotes gravity.

Figure 5 shows the pressure field for all monitored points. The numerical simulation is compared to the linear theory. The transient phase of the pressure time history is due to the initial condition.

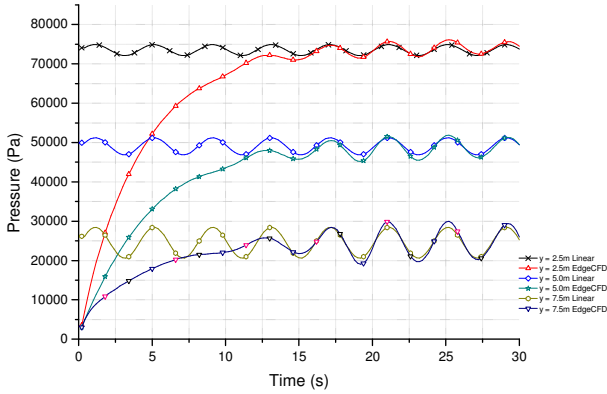


Figure 5. Pressure on monitored points.

In all three points, after the initial transient, a good agreement between the numerical and analytical pressure solutions is observed.

Mean pressure is presented in table 1 and the dynamic pressure amplitudes are presented in table 2. The values were taken in the last 5 minutes of simulation.

	y = 2.5 m	y = 5.0 m	y = 7.5 m
Linear mean pressure (Pa)	73634,47	49174,80	24776,55
EdgeCFD mean pressure (Pa)	73269,62	48896,79	24510,88

Table 1. Mean pressure.

	y = 2.5 m	y = 5.0 m	y = 7.5 m
Linear amplitude of dynamic pressure (Pa)	1371,88	2185,01	3914,47
EdgeCFD amplitude of dynamic pressure (Pa)	1953,80	2941,15	4831,95

Table 2. Amplitude of dynamic pressure.

Velocity field for each one of three monitored points is shown in Figures 6 – 11. As it happens in pressure solution, the simulation needs an adjustment time to converge to steady-state solution.

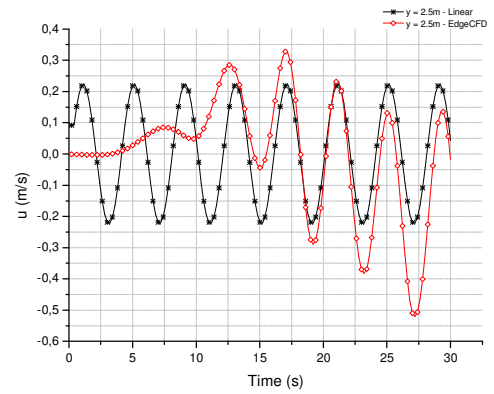


Figure 6. Velocity. Component u . $x = 60.0$ and $y = 2.5$ m.

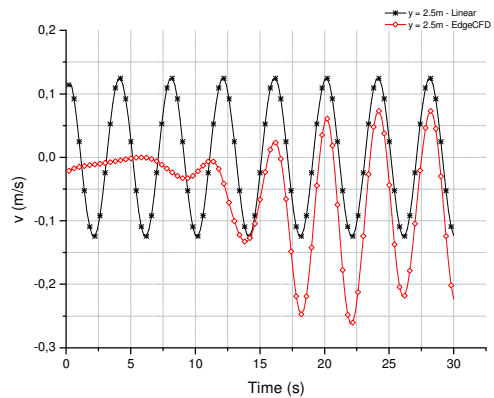


Figure 7. Velocity. Component v . $x = 60.0$ and $y = 2.5$ m.

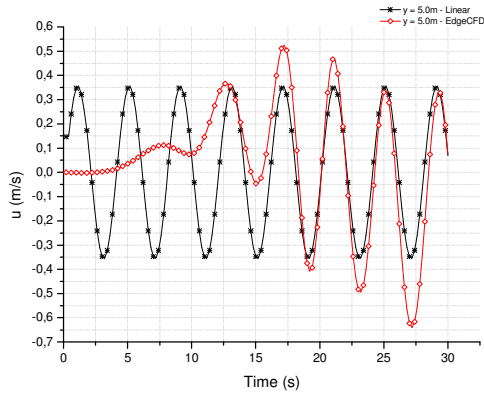


Figure 8. Velocity. Component u . $x = 60.0$ and $y = 5.0$ m.

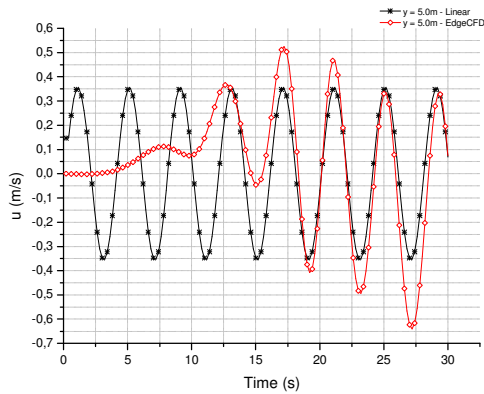


Figure 9. Velocity. Component v . $x = 60.0$ and $y = 5.0$ m.

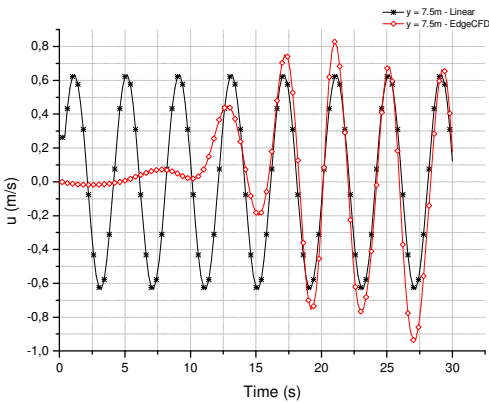


Figure 10. Velocity. Component u . $x = 60.0$ and $y = 7.5$ m.

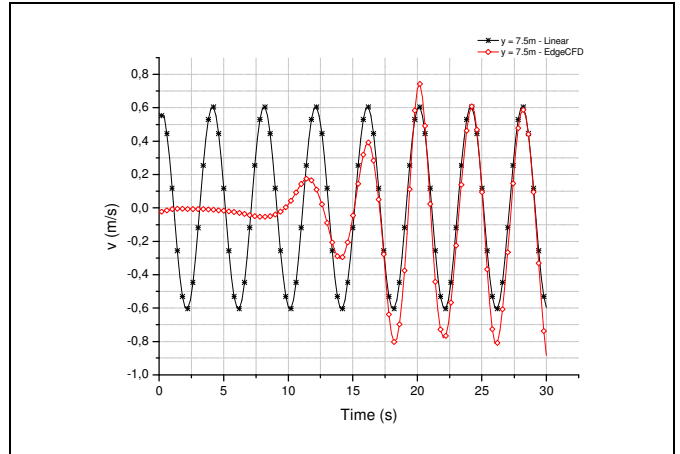


Figure 11. Velocity. Component v . $x = 60.0$ and $y = 7.5$ m.

Note that, there is a good agreement of the frequency, but the amplitudes are not so close. On the other hand, it is important to observe that linear theory can not cope with viscosity effects and non-linear free surface effects.

Progressive wave interaction with a fixed cylinder

This case refers to the interaction of a progressive wave with a fixed circular cylinder placed on the center of the domain, as shown in Figure 12. The ground based circular section structure has a depth of 15 m (5m in the air and 10 m under the water) and its radius is 5m. All dimensions of progressive wave original case domain were kept, except the width. This dimension is taken as 50 m, being a good choice to diminish the effects of lateral walls on the wave-cylinder interaction. The piston, represented by the face at $y = 0$ m, moves cyclically in x - direction with a 0.40 m amplitude and period of 4 seconds. The fluid properties are: $\rho_{water} = 1000.0 \text{ kg/m}^3$, $\mu_{water} = 0.001 \text{ kg/(m s)}$, $\rho_{air} = 1.0 \text{ kg/m}^3$, $\mu_{air} = 0.001 \text{ kg/(m s)}$. The air viscosity is considered equal to the water in order to avoid unstable regions in the air phase.

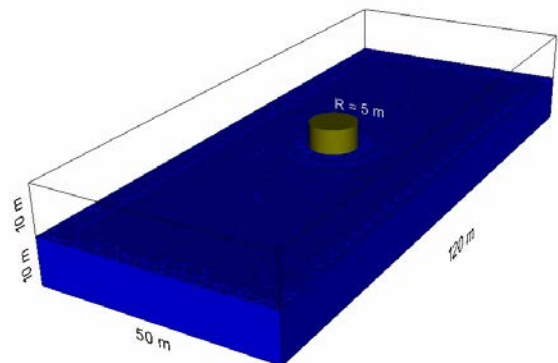


Figure 12. Progressive wave over a cylinder. Free surface initial condition and domain.

Figure 13 shows the computational domain and the partitions for parallel processing. The computation was performed in parallel (MPI) in a SGI Altix XE 1200, employing 8 of its 88 Xeon cores (2.66 GHz). The maximum inexact-Newton linear tolerance was set to 10^{-3} while the linear tolerance for the marking function equation was 10^{-5} . The nonlinear loops were stopped after the relative residual as well as the relative step increment decreased 3 orders of magnitude. In addition, a fixed time step of 0.005 was employed. For 50 seconds of simulation it was spent about 70 hours of CPU time. Figure 14 presents five snapshots representing the time solutions at 10, 20, 30, 40 and 50 seconds.

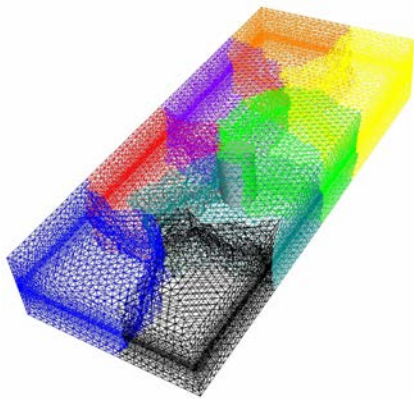


Figure 13. Progressive wave over a cylinder. Computational domain: Tetrahedral finite elements unstructured mesh with 195,426 nodes and 1,140,159 elements and partitioning for 8 processors.

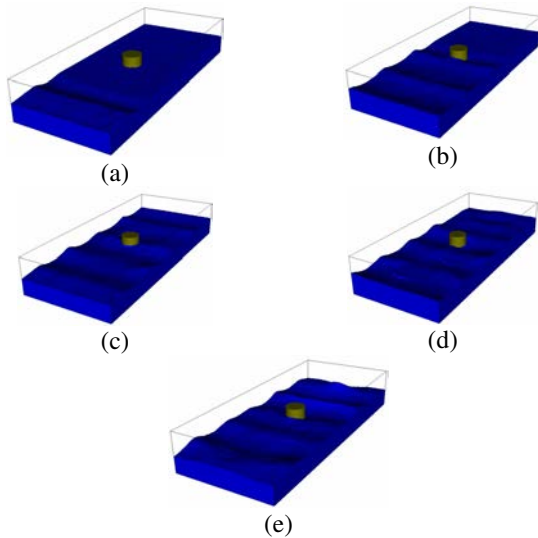


Figure 14. Progressive wave over a cylinder. Free surface elevation at (a) $t = 10$ s, (b) $t = 20$ s, (c) $t = 30$ s, (d) $t = 40$ s, (e) $t = 50$ s.

Figure 15 presents the kinetic energy of the fluid. Since the fluid is initially at rest, the kinetic energy is introduced by the piston. The oscillatory movement of the piston, the wave-cylinder and wave-opposite face interactions are captured in the kinetic energy oscillatory behavior. Figure 16 shows the water mass preservation with respect to the initial water mass, and reveals how efficiently the mass correction algorithm works. Note that most of the disturbances occur at the initial stage, up to 1 second, which coincides with the moment that the piston pushes the fluid, which is initially at rest. It is important to remind that the mass correction algorithm is based on the velocity field normal to the free surface, making the correction effects more pronounced when the interface is moving faster.

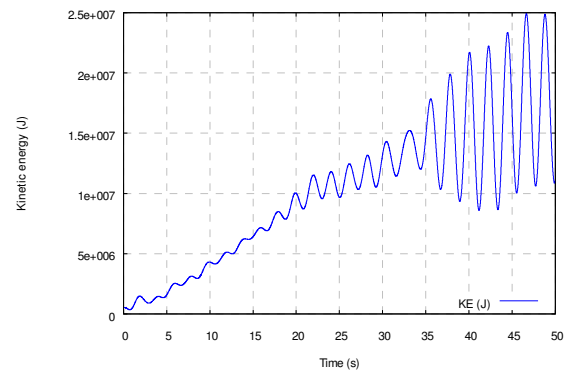


Figure 15. Kinetic energy of progressive wave.

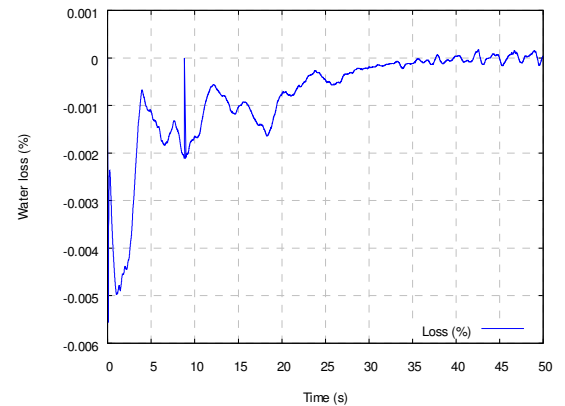


Figure 16. Water phase preservation

The maximum computed water loss was around 0.005% which is one order of magnitude smaller than the adopted nonlinear stopping criteria. It means that material losses are negligible. The history of the horizontal hydrodynamic force exerted by the fluid over the cylinder is illustrated by Figure 17. The amplitude of the linear force evaluated using the MacCamy and Fuchs analytical formulation is equal to 612,299 N [29] and by the WAMIT® code is 591,257 N, [30]. These results, based on linear diffraction theory, are close to the numerical results

obtained by the program. The causes of the changes in the time history of the horizontal force (between 37s and 45s for example) could at first be imputed to the wall reflections at the end of the channel, but deserves more investigation

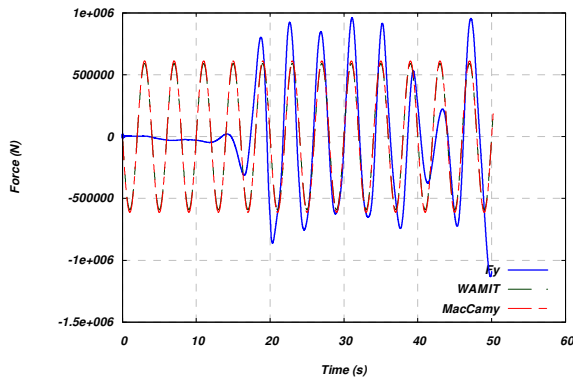


Figure 17. Force exerted by the progressive wave over the cylinder.

CONCLUSIONS

In this work we have successfully extended our parallel incompressible flow solver to problems involving wave interactions with moving physical boundaries. This goal was reached by introducing Arbitrary Lagrangian-Eulerian formulation into the stabilized finite element formulation. We have validated the present formulation simulating a progressive wave generated from a piston-like wave maker and its interaction with a vertical circular cylinder. The main conclusions are that some important parameters in progressive wave behavior, as celerity, wavelength and height, are in good agreement with existing linear theoretical values. In the problem of the progressive wave interaction with a fixed circular cylinder, we have computed the horizontal force over the cylinder and the kinetic energy of the fluid, which is introduced by the piston movement. We noticed the linear force given by the MacCamy and Fuchs analytical formulation and the WAMIT® code are close to the numerical results obtained in the simulation. Furthermore, we verified in this case that the maximum loss computed is around 0.005%, which evidences the efficiency of global mass conservation procedure.

ACKNOWLEDGMENTS

The authors would like to thank the financial support of the PETROBRAS Research Center; the MCT/CNPq Brazilian Council for Scientific Research and MCT/FINEP CT-AQUAVIÁRIO-2006. The Center for Parallel Computations (NACAD) and the Ocean Basin (LabOceano) at the Federal University of Rio de Janeiro provided the computational resources for this research.

REFERENCES

- [1] Tezduyar, T. E. and Osawa Y., 2000, "Finite Element Stabilization Parameters Computed from Element Matrices and Vectors. *Computer Methods in Applied Mechanics and Engineering*; **190**, pp. 411-430.
- [2] Tezduyar, T. E., Mittal S., Ray S. E. and Shih R., 1992, "Incompressible Flow Computations with Stabilized Bilinear and Linear Equal-Order-Interpolation Velocity-Pressure Elements", *Comput. Methods Appl. Mech. Engrg.* **95**, pp. 221-242.
- [3] Tezduyar, T. E., 2001, "Finite Element Methods for Flow Problems with Moving Boundaries and Interfaces". *Archives of Computational Methods in Engineering*; **8**, pp. 83-130.
- [4] Koshizuka S, Tamako H and Oka Y., 1995, "A particle method for incompressible viscous flow with fluid fragmentation", *Computational Fluid Mechanics Journal*, **113**, pp. 134-147.
- [5] Violeau D and Issa R., 2007, "Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview", *International Journal for Numerical Methods in Fluids*, **53**, pp.277-304.
- [6] Del Pin F, Idelsohn S, Oñate E, Aubry R., 2007, "The ALE/Lagrangian particle finite element method: a new approach to computation of free-surface flows and fluid-object interactions", *Computers and Fluids*, **36**, pp. 27-38.
- [7] Takizawa K, Yabe T, Tsugawa Y, Tezduyar TE and Mizoe H., 2006, "Computation of free-surface flows and fluid-object interactions with the CIP method based on adaptive meshless Soroban grids", *Computational Mechanics*, **40**, pp. 167-183.
- [8] Hirt, C. W. and Nichols, B. D. 1981, "Volume of Fluid (VOF) Methods for the Dynamics of Free Boundaries", *Journal of Computational Physics*; **39**, pp. 201-225.
- [9] Sethian, J. A., 1999, "Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, *Cambridge University Press*, Cambridge, UK.
- [10] Tezduyar, T. E., Aliabadi, S. and Behr, M., 1997, "Enhanced-Discretization Interface-Capturing Technique", in: Matsumoto Y, Prosperetti A (eds.). *Proceedings of the ISAC'97 High Performance Computing on Multiphase Flows*; vol. **1-6**.
- [11] Lohner, R., Yang, C., Onate, E., 2006, "On the Simulation of Flows with Violent Free-Surface Motion", *Computer Methods in Applied Mechanics and Engineering*, **195**, pp. 5597-5620.
- [12] Elias, R. N. and Coutinho, A. L. G. A., 2007, "Stabilized Edge-Based Finite Element Simulation of Free-Surface Flows", *Int. J. Numer. Meth. Fluids*, **54**, pp. 965-993.

- [13] Elias, R. N., Coutinho, A. L. G. A., Esperança, P. T. T., Gonçalves Jr., M. A., Ferreira, M. D. A. S. and Martins, M.A.D., 2008, “Computational Techniques for Stabilized Edge-Based Finite Element Simulation of Free-Surface Flows”, *Int. Conf. on Offshore and Arctic Engineering, OMAE, Estoril, Portugal*.
- [14] Donea, J. and Huerta, A., 2003. Finite Element Method for Flow Problems. *Wiley & Sons*.
- [15] Smagorinsky, J., 1963, “General Circulation Experiments with Primitive Equations, part i: the Basic Experiment”, *Monthly Wea. Rev.*, **91**, pp. 99-152.
- [16] Galeão, A. C, do Carmo, E. G. D. 1988, “A Consistent Approximate Upwind Petrov-Galerkin Method for Convection-Dominated Problems” *Computer Methods in Applied Mechanics and Engineering*; **68**, pp. 83-95.
- [17] Brooks, A. N. and Hughes, T. J. R., 1982, “Streamline-upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equation”, *Comput. Methods Appl. Mech. Engrg*; **32**, pp. 199-259.
- [18] Bazilevs, Y., Calo, V. M., Tezduyar, T. E., Hughes, T. J. R., 2007, “ $YZ\beta$ Discontinuity Capturing for Advection-Dominated Processes with Application to Arterial Drug Delivery”, *International Journal for Numerical Methods in Fluids*, **54**, pp. 593-608.
- [19] Franca, L. P. and Frey, S. L., 1992, “Stabilized Finite Element Methods: ii. The incompressible Navier-Stokes equations”, *Computer Methods in Applied Mechanics and Engineering*, **99**, pp. 209-233.
- [20] Elias, R. N., Martins, M. A. D. and Coutinho, A. L. G. A., 2006, “Parallel Edge-Based Solution of Viscoplastic Flows with the SUPG/PSPG Formulation”. *Computational Mechanics*, **38**, pp. 365–381.
- [21] Valli, A. M. P, Carey, G. F. and Coutinho, A. L. G. A., 2005, “Control Strategies for Timestep Selection in FE Simulation of Incompressible Flows and Coupled Reaction-Convection-Diffusion Processes”. *Int. J. Num. Meth. in Fluids*, **47**, pp. 201-231.
- [22] Elias, R. N., Coutinho, A. L. G. A and Martins, M. A. D., 2006, “Inexact Newton-Type Methods for the Solution of Steady Incompressible Viscoplastic Flows with the SUPG/PSPG Finite Element Formulation”. *Computer Methods in Applied Mechanics in Engineering*; **195**, pp. 3145-3167.
- [23] Coutinho, A. L. G. A., Martins, M. A. D., Sydenstricker, R. M. and Elias, R. N., 2006, “Performance Comparison of Data-Reordering Algorithms for Sparse Matrix-Vector Multiplication in Edge-Based Unstructured Grid Computations”. *International Journal for Numerical Methods in Engineering*, **66**, pp. 431-460.
- [24] Martins, M. A. D., Elias, R. N. and Coutinho, A. L. G. A., 2007, “Edgepack: A Parallel Vertex and Node Reordering Package for Optimizing Edge-Based Computations in Unstructured Grids”. *Lecture Notes in Computer Science*; **4395**, pp. 292-304.
- [25] Elias, R. N, Martins, M. A. D and Coutinho, A. L. G. A., 2005, “Parallel Edge-Based Inexact Newton Solution of Steady Incompressible 3D Navier-Stokes Equations”, *Lecture Notes in Computer Science*, **3648**, pp. 1237-1245.
- [26] Journée, J.M.J. and Massie, W.W., 2001. Offshore Hydromechanics. *Delft University of Technology*.
- [27] Hughes, S. A., 1993 “Physical Models and Laboratory Techniques in Coastal Engineering”, *Advanced Series in Ocean Engineerin, World Scientific*.
- [28] Dean, R.G. and Dalrymple, R. A., 1984 “Water Wave Mechanics for Engineers and Scientists”, *Prentice-Hall*.
- [29] Sarpkaya, T. and Isaacson, M., 1981, “Mechanics of Wave Forces on Offshore Structures”, Van Nostrand Reinhold.
- [30] WAMIT, Inc., “Wamit 6.1 User Manual”.