Renato N. Elias
e-mail: renato@nacad.ufrj.br

Milton A. Gonçalves, Jr.
e-mail: milton@nacad.ufrj.br

Alvaro L. G. A. Coutinho
e-mail: alvaro@nacad.ufrj.br

High Performance Computing Centre,
Federal University of Rio de Janeiro,
P.O. Box 68506,
Rio de Janeiro, RJ 21945-970, Brazil

Paulo T. T. Esperança
Department of Naval and Ocean Engineering,
Federal University of Rio de Janeiro,
P.O. Box 68508,
Rio de Janeiro, RJ 21945-970, Brazil
e-mail: ptarso@peno.coppe.ufrj.br

Marcos A. D. Martins
e-mail: marcos.martins@petrobras.com.br

Marcos D. A. S. Ferreira
e-mail: marcos.donato@petrobras.com.br

PETROBRAS Research Center,
Avenida Horacio Macedo 850,
Rio de Janeiro, Brazil

Computational Techniques for Stabilized Edge-Based Finite Element Simulation of Nonlinear Free-Surface Flows

Free-surface flows occur in several problems in hydrodynamics, such as fuel or water sloshing in tanks, waves breaking in ships, offshore platforms, harbors, and coastal areas. The computation of such highly nonlinear flows is challenging, since free-surfaces commonly present merging, fragmentation, and breaking parts, leading to the use of interface-capturing Eulerian approaches. In such methods the surface between two fluids is captured by the use of a marking function, which is transported in a flow field. In this work we discuss computational techniques for efficient implementation of 3D incompressible streamline-upwind/Petrov–Galerkin (SUPG)/pressure-stabilizing/Petrov–Galerkin finite element methods to cope with free-surface problems with the volume-of-fluid method (Elias, and Coutinho, 2007, "Stabilized Edge-Based Finite Element Simulation of Free-Surface Flows," Int. J. Numer. Methods Fluids, ***54****, pp. 965–993). The pure advection equation for the scalar marking function was 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 of the interface. We introduce parallel edge-based data structures, a parallel dynamic deactivation algorithm to solve the marking function equation only in a small region around the interface. The implementation is targeted to distributed memory systems with cache-based processors. The performance and accuracy of the proposed solution method is tested in the simulation of the water impact on a square cylinder and in the propagation of a solitary wave.* [DOI: 10.1115/1.3124136]

1 Introduction

Free-surface flows occur in many hydrodynamics problems. Sloshing of liquids in tanks, wave breaking in ships, offshore platforms, harbors, coastal areas, 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 (DSD/SST) finite element formulation proposed by Tezduyar et al. [1–3] is a mesh based example of the 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. presented in Ref. [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 Ref. [7] by Takizawa et al. In this work the authors enhanced the constrained interpolation profile (CIP) method for solving hyperbolic equations with a meshless Soroban grid. The resulting formulation was used to treat fluid-object and fluid-structure interactions 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 postprocessing 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

Contributed by the Ocean Offshore and Arctic Engineering Division of ASME for publication in the JOURNAL OF OFFSHORE MECHANICS AND ARCTIC ENGINEERING. Manuscript received August 16, 2008; final manuscript received February 26, 2009; published online September 8, 2009. Review conducted by Solomon Yim. Paper presented at the ASME 27th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2008), Estoril, Portugal, June 15–20, 2008.

(VOF) scheme, first 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. The enhanced-discretization interface-capturing method, first proposed by Tezduyar et al. in Ref. [9] and the work of Lohner et al. presented in Ref. [10], are both examples of unstructured grid formulations based on the finite element method to solve free-surface flows using a VOF marking function. Level set methods [11] 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.

In this work we use our VOF edge-based stabilized finite element solver [12] to deal with complex free-surface problems. 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 to improve performance; support to message passing and shared memory parallel programming models; and large eddy simulation extensions using a classical Smagorinsky model.

The remainder of this paper is organized as follows. Section 2 presents the incompressible flow and interface-capturing governing equations, respectively, Sec. 3 summarizes the solution procedures employed, and Sec. 4 shows results obtained for the simulation of the water impact on a square cylinder and in the propagation of a solitary wave. We compare our results with available experimental and analytical results showing that the present scheme is fast, simple, and accurate. The final remarks and conclusions are summarized in Sec. 7.

2 Governing Equations

2.1 Incompressible Fluid Flow. Let $\Omega \subset \mathbb{R}^{n_{\mathrm{SD}}}$ 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:

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

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

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

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

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

$$\mathbf{T} = 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) \tag{4}$$

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

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \tfrac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T) \tag{5}$$

In the present work a large eddy simulation (LES) approach to turbulence is considered by the use of a classic Smagorinsky turbulence model [13]. 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

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

where C_S is the Smagorinsky constant, ranging from 0.1 to 0.2.

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

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

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

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, $S_{\mathbf{u}}^h$, $V_{\mathbf{u}}^h$, S_p^h, and $V_p^h = S_p^h$. The finite element formulation of Eqs. (1) and (2) using SUPG and PSPG stabilizations for incompressible fluid flows can be written (see Ref. [1]) as follows. Find $\mathbf{u}^h \in S_{\mathbf{u}}^h$ and $p^h \in S_p^h$ such that $\forall \mathbf{w}^h \in V_{\mathbf{u}}^h$ and $\forall q^h \in V_p^h$

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

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

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

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

In Eq. (9), the last summation is the least-squares incompressibility constraint term [3] added to prevent spurious oscillations at high Reynolds number flows. The LSIC stabilization parameter is

$$\tau_{\mathrm{LSIC}} = \frac{\|\mathbf{u}^h\|h}{2} \tag{11}$$

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

2.2 Interface Capturing. The VOF method was first proposed by Hirt and Nichols [8] for finite difference schemes. The idea of the method is to define a scalar marking function over the computational domain in such a manner that its value at a certain point and instant indicates the fraction of the fluids involved. Thus, a scalar marking function can be employed to capture the position of the interface between the fluids by simply using the fluid's 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

$$\frac{\partial \phi}{\partial t} + \boldsymbol{\nabla} \cdot (\mathbf{u}\phi) = 0 \tag{12}$$

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

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

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

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

The finite element formulation of Eq. (12) can be written as follows. Find $\phi^h \in S^h_\phi$, such that, $\forall w^h \in V^h_\phi$

$$\int_\Omega w^h\left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\phi^h\right)d\Omega + \int_\Omega w^h(\boldsymbol{\nabla}\cdot\mathbf{u}^h)\phi^h d\Omega + \sum_{e=1}^{n_{\text{el}}}\int_{\Omega^e}\tau_{\text{SUPG}}\mathbf{u}^h\cdot\boldsymbol{\nabla}w^h\left(\frac{\partial \phi^h}{\partial t} + \mathbf{u}^h\cdot\boldsymbol{\nabla}\phi^h + (\boldsymbol{\nabla}\cdot\mathbf{u}^h)\phi^h\right)d\Omega^e + \sum_{e=1}^{n_{\text{el}}}\int_{\Omega^e}\delta(\phi^h)\,\boldsymbol{\nabla}\,w^h\cdot\boldsymbol{\nabla}\phi^h d\Omega = 0 \tag{15}$$

where S^h_ϕ and V^h_ϕ are standard test and weight finite element spaces. The first two integrals represent the Galerkin formulation of Eq. (12), while the first elementwise 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 [14]. The evaluation of τ_{SUPG} and δ stabilization terms follow the definitions described in Refs. [14,15], respectively. The discretization of Eq. (15) leads us to a nonlinear 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 Ref. [16], where δ is computed as

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

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

$$R^e(\phi^h) = \frac{\partial \phi^h}{\partial t} + \boldsymbol{\nabla}\cdot(\mathbf{u}^h\phi^h) \tag{17}$$

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 [14]. In order to avoid nonphysical results, values lying outside the range [0,1] were truncated with the following function:

$$\phi^h = \min[\max[\phi^h,0],1] \tag{18}$$

3 Solution Procedure

The computational solution kernels consist of predictor multicorrector time integration schemes, as described in Refs. [15,17] 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 in the Navier–Stokes equation are treated by an inexact Newton-generalized minimum residual (GMRES) algorithm, as described in Ref. [18]. 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 toward 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 is employed 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 Ref. [19]) 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 and to diminish floating point operation counts (flops) and memory requirements, as described in Refs. [20,21] 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 [22]. All computational kernels are parallelized, as detailed in Ref. [23]. Thus, to enable these computational enhancements, the mesh is preprocessed. First, it was partitioned into several subdomains by METIS [24], and the nodes, edges, and elements within the subdomains are reordered for improving data locality by EdgePack. It is worth noticing, however, that all preprocessing took a negligible fraction of the whole solution time.

4 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 et al. [25], 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, and undulations in the solution of the marking function advection. Level set methods suffer when the marking function lose 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 Refs. [12,25] 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

$$u_n = \left|\mathbf{u}\cdot\frac{\nabla\phi}{\|\nabla\phi\|}\right| \tag{19}$$

The amount computed from Eq. (19) guarantees that the mass correction will act mainly in regions where the interface moves 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 Ref. [12], Elias and Coutinho evaluated different test problems with the help of the mass preservation algorithm, showing its efficiency.

5 Parallel Dynamic Deactivation

We use here another computational artifact to further improve the overall efficiency of the present free-surface solver, the parallel dynamic deactivation (PDD) technique for solving the marking function. This technique is an extension of the dynamic deactivation (DD) procedure, which is an algorithm that restricts the computation on regions were a defined gradient is found. It was first

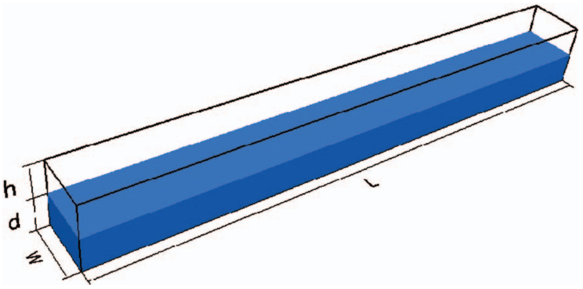


Fig. 1 Schematic view of computational domain. *L*=200 m, *h*=10 m, *d*=10 m, and *w*=0.5 m.

presented by Lohner and Camelli in Ref. [26] for contaminant transport problems. In Ref. [12] the original DD scheme was extended to the parallel computation of free-surface flows. Since the marking function employed on VOF methods presents steep gradients, the dynamic deactivation algorithm catches and restricts the computations only on regions around the interface.

Consequently, most of the computational effort that would be necessary to solve the interface transport over the whole domain is considerably saved. Moreover, a buffer zone around the interface is built to assure that the interface is kept within the enabled region in each time step. It is important to emphasize that although the computational costs associated to the transport problem are recognizably lesser than those spent by the Navier–Stokes solution, a similar approach can be employed to restrict the computations only on regions filled by the aimed fluid during the incompressible flow solution phase. In the present context the set of active elements initially selected by the DD algorithm is based on the following criteria:

$$\|\nabla\phi\|^e \geq \eta\|\nabla\phi\|_m \tag{20}$$

where $\|\nabla\phi\|^e$ is the Euclidean norm of the element gradient solution, $\|\nabla\phi\|_m$ is the average gradient norm computed for the whole computational grid, and $\eta\in[0,1]$ is a parameter, which controls how the element selection must work.

It is worthwhile to mention that even producing unbalanced partitions, the PDD still produces computational gains since it can drastically reduce the overall effort (computation and communication) to solve the problem. Moreover, the PDD does not spend any further effort in repartitioning, renumbering, and redistributing the entities over the processors. In fact, the PDD partitions are produced and controlled by the use of lists of active entities, which restrict the main loops of the solver. For more details, please see Ref. [12]

6 Test Problems

6.1 Solitary Wave. In the first problem we present simulation results of solitary wave propagation [27]. This problem consists of an elevation of free-surface that suffers an initial "push" and then propagates itself. This push, that drives the wave initially, is introduced in the problem by velocity initial conditions. The geometry of computational domain is presented in Fig. 1.

The initial conditions are provided by Laitone's approximation [28], given by

$$u_x^0 = \sqrt{gd}\frac{H}{d}\sec h^2\left[\sqrt{\frac{3H}{4d^3}}x\right] \tag{21}$$

$$u_y^0 = \sqrt{3gd}\left(\frac{H}{d}\right)^{3/2}\left(\frac{y}{d}\right)\sec h^2\left[\sqrt{\frac{3H}{4d^3}}x\right]\tan h\left[\sqrt{\frac{3H}{4d^3}}x\right] \tag{22}$$

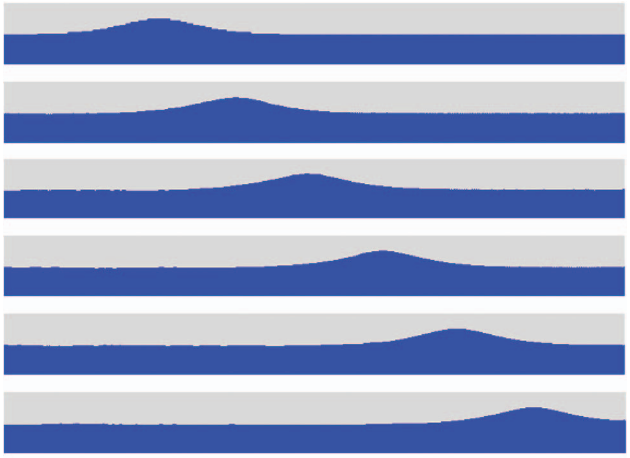


Fig. 2 Solitary wave propagation from free-surface elevation at *t*=0 s, *t*=2 s, *t*=4 s, *t*=6 s, *t*=8 s, and *t*=10 s.

$$\eta = d + H\sec h^2\left[\sqrt{\frac{3H}{4d^3}}x\right] \tag{23}$$

where g and d are the gravity acceleration and water depth, H and η are the initial amplitude of solitary wave and free-surface elevation, x and y are spatial coordinates, and u_x^0 and u_y^0 are the initial velocities in the x and y directions, respectively. For this example, H was taken as 5 m. The unstructured finite element mesh has 42,168 nodes and 102,705 tetrahedral elements, and it is more refined around the air/water interface. The time step was fixed in 0.01 s, and the simulation time is 10 s. The maximum inexact-Newton linear tolerance for GMRES was set to 10^{-1} for the Navier–Stokes solver and to 10^{-3} for the marking function equation. For nonlinear loops, both relative residual and relative step increment are used as stopping criteria and were set to 10^{-3}. With these data, that Navier–Stokes inexact-Newton flow solver converges in six or seven iterations in the initial simulation stages, but after that, four iterations are enough. For the marking function, convergence to the solution occurs in five or six iterations during almost the entire simulation.

Figure 2 shows the solitary wave propagation, generated from the initial condition given by Eqs. (21)–(23) during the first 10 s of simulation.

The wave celerity, the pulse velocity in the x direction, is given by

$$c = \sqrt{gd\left(1+\frac{H}{d}\right)} \tag{24}$$

According to the problem data, the wave celerity computed by Eq. (24) is $c=12.12$ m/s, which compares well with the simulation, since the pulse's displacement is approximately 120 m in 10 s, as shown in Fig. 2. The numerical wave celerity is equal to 12.03 m/s, which is in good agreement with the analytical value.

6.2 Wave Impact With a Tall Structure. In this problem we simulate the interaction of a dam break wave with a tall structure. This problem has been experimentally studied at the University of Washington to model the effects due to Tsunami waves. It consists of a dam break wave, which starts to flow due to gravity, and reaches inertia before hitting a tall structure. It has been also numerically studied by authors, such as Gesteira and Dalrymple [29], Wu [30], and Raad [31], by a variety of methods. The collected experimental data included the net force on the cylinder and the fluid velocity in a fixed position, 0.146 m upstream of the center of the structure and 0.026 m of the floor of the tank. Forces

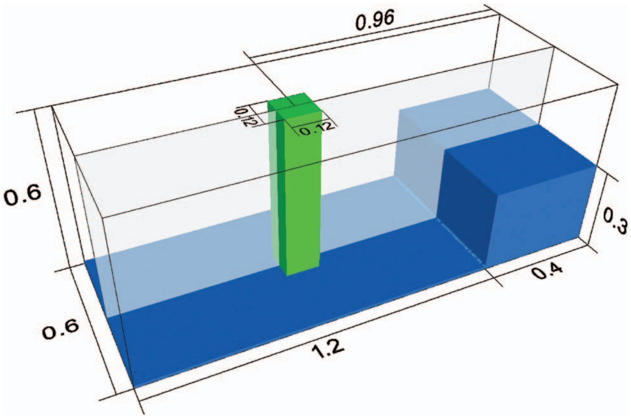


Fig. 3 Wave impact with a tall structure—model description

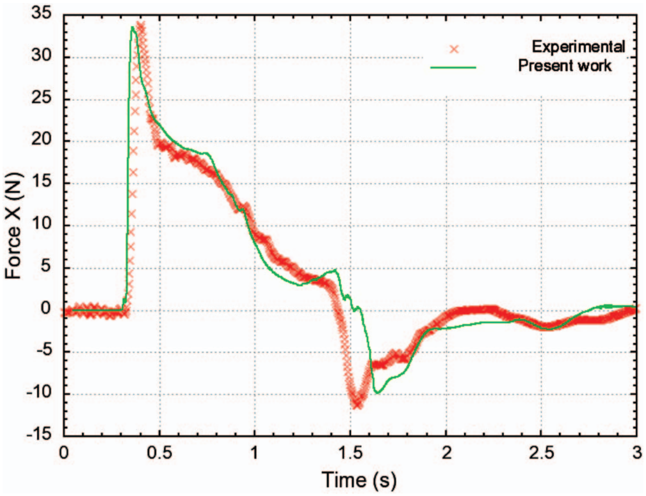


Fig. 4 Impact force on the column in the mainstream direction

were measured with a load cell and velocities with a laser-doppler velocimetry (LDV) system.

The model shown in Fig. 3 comprises a box, representing a tank with dimensions $1.6\times0.6\times0.6\ \text{m}^3$ ($L\times W\times H$), where a water column of $0.4\times0.6\times0.3\ \text{m}^3$ initially rests at the tank extreme.

An obstacle, representing a rigid column, of $0.12\times0.12\times0.6\ \text{m}^3$ is placed 0.96 m from origin at the center line of the tank. Following Wu [30] we considered only half the width of the tank according to the symmetry plane illustrated in Fig. 3.

A thin layer of 1 cm of water at the bottom of the tank was also modeled, which corresponds to a small leakage due to a deficient drainage observed in the experimental set up [29,30]. The fluid properties are $\rho_{\text{water}}=1000.0\ \text{kg/m}^3$, $\mu_{\text{water}}=0.001\ \text{kg/(m s)}$, $\rho_{\text{air}}=1.0\ \text{kg/m}^3$, and $\mu_{\text{air}}=0.001\ \text{kg/(m s)}$. The air viscosity is considered equal to the water in order to avoid unstable regions in the air phase.

The problem was ran in parallel (MPI) in a SGI Altix 450 system employing 8 of its 32 Itanium-2 cores (1.6 GHz). For 3 s of simulation, about 13 h of CPU time was spent. The finite element mesh was built with 1,600,137 tetrahedra, 1,943,367 edges, and 279,107 nodes, which, after applying the boundary conditions, resulted in 1,057,725 velocity equations and 362,907 pressure equations. The maximum inexact-Newton linear tolerance was set to 10^{-1}, while the linear tolerance for the marking function equation was 10^{-3}. The nonlinear loops were stopped after the relative residual, as well as the relative step increment decreased by three orders of magnitude. In addition, a fixed time step of 0.0001 was employed.

Figure 4 shows the history of the hydrodynamic force exerted by the fluid when flowing around the column. The maximum computed force was 33.57 N at instant 0.36 s, while the experimental value reported is 33.90 N at instant 0.4 s.

It is important to note that a small lag of 0.2 s is reported for the gate to finish its opening in the experimental set up. The computed results for the force profile are in good agreement with the experimental data. A small lag of 0.10 s can be observed for the moment that the wave returns and hits the column at the second time. Moreover, the computed force value of −9.87 N was smaller than the experimental value (−11.33 N).

Figure 5 shows the velocity in the x direction measured by a LDV gauge placed 14.6 cm upstream of the center of the column and 2.6 cm of the tank's floor (75.4 cm, 31 cm, and 2.6 cm related to the absolute tank coordinates). Since we are using an unstructured grid, the velocity values were extracted from the nearest node. As can be seen in Fig. 5, the computed velocity time history close to the gauge follows the global behavior presented by the experiment.

The maximum water loss was around 0.06%, which is one order of magnitude smaller than the nonlinear stopping criteria. In other words, material losses are negligible.

Figure 6 shows snapshots of the simulation for the instant when the dam break wave collides with the structure (Fig. 6(*a*)), the moment when the water flow starts to revert (Fig. 6(*b*)) and when the second collision occurs (Fig. 6(*c*)).

Figure 7 shows the history of active edges per processor due to the parallel dynamic deactivation algorithm. As may be noticed, the PDD artifact, as employed in this work, produces an unbalanced work distribution, however, it is important to realize that there is an expressive reduction in computational effort. Note in Fig. 7 that, in the worst case, processor 1 (P1) had its computational effort reduced by 15% when compared with the computations being performed for the full model. The PDD algorithm, restricting computations to a narrow band around the free-surface, has a positive impact on the overall performance.

7 Conclusions

In this work we have discussed the use of several computational techniques for the parallel edge-based solution of free-surface flows on unstructured grids. The unsteady three-dimensional Navier–Stokes equations were discretized with a SUPG/PSPG stabilized finite element formulation, including a least-squares in-

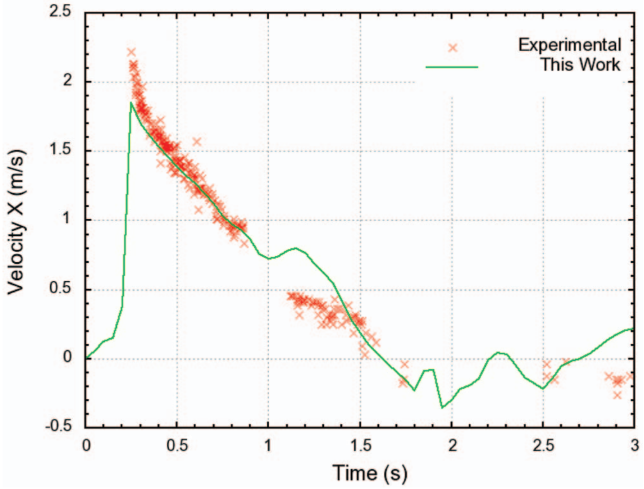
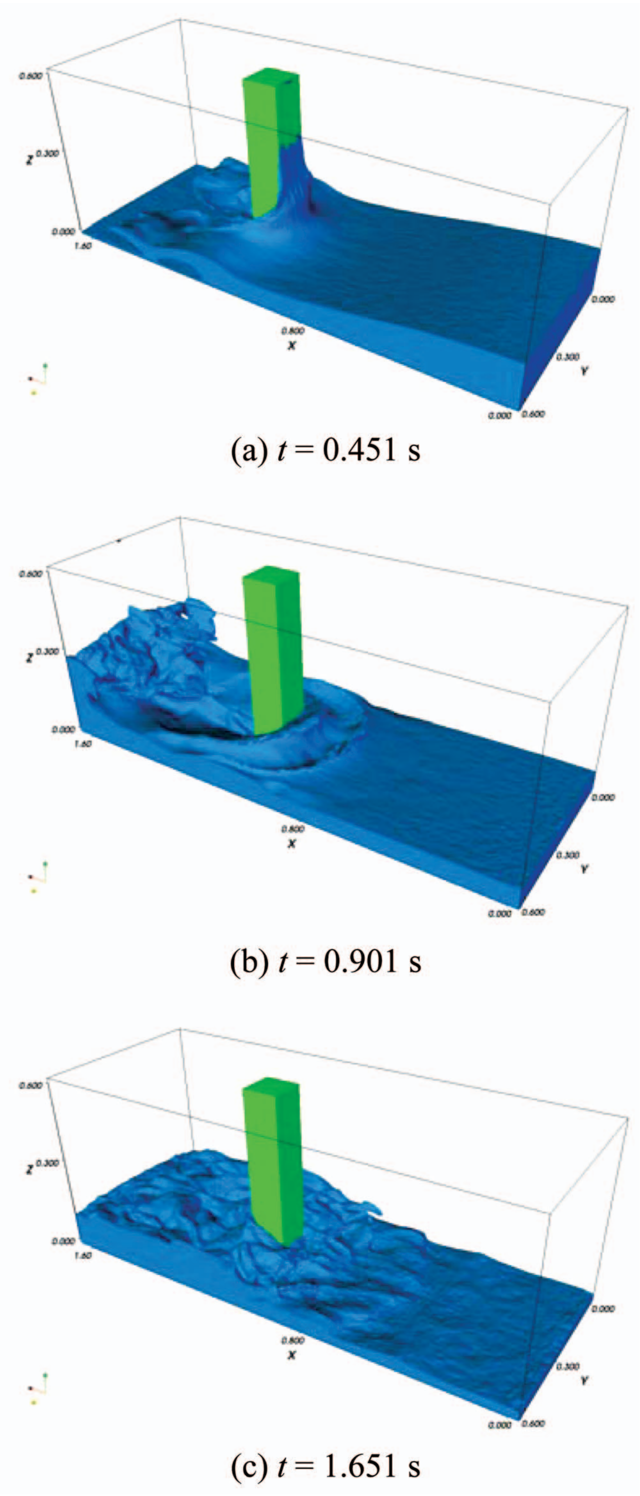


Fig. 5 Velocity in the *x* direction



(a) $t = 0.451$ s

(b) $t = 0.901$ s

(c) $t = 1.651$ s

Fig. 6 Snapshots for the simulation of the wave impact with a tall structure: (*a*) $t=0.451$ s, (*b*) $t=0.901$ s, and (*c*) $t=1.651$ s

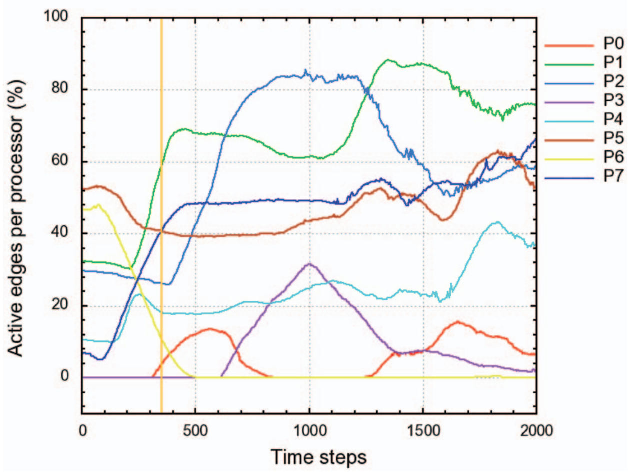


Fig. 7 Parallel dynamic deactivation region of active entities

compressibility constraint term and a classical Smagorinsky model. The main characteristics of this flow solver are implicit time marching scheme with adaptive time stepping control; advanced inexact-Newton solvers; edge-based data structures to save memory and to improve performance; support to message passing; and shared memory parallel programming models. Into this flow solver we introduced VOF extensions to track the evolving free-surface. The pure advection equation for the scalar marking function was 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 of the interface. This guarantees that the mass correction will act mainly in regions where the interface moves faster, while keeping the stationary regions untouched. This procedure is accurate, provided the volume of the fluid phases is computed correctly. We introduced here another computational artifact to further improve the overall efficiency of the present free-surface solver, the PDD technique for solving the marking function. This technique restricts the computation to regions where a defined gradient is found. Since the marking function employed on VOF methods presents steep gradients, the dynamic deactivation algorithm catches and restricts the computations only on regions around the interface. In the PDD extension, each processor estimates its own number of finite element entities enabled for computation, and the loops are constrained to the number of active entities (nodes, edges, and elements). Consequently, the overall effort (computation and communication) to solve the problem is drastically reduced.

We validated the whole solution procedure simulating a solitary wave and the water impact on a square cylinder. Our simulation was able to reproduce the solitary wave shape and celerity. In problem of the interaction of a dam break wave with a tall structure, good agreement was observed between our results and the available experimental results for the time histories of the impact force on the column and the measured water velocity. We noticed in this case that the global mass conservation procedure is very effective, keeping the volume loss below 1%. In this more challenging application, the free-surface presents a complex behavior, involving fragmentation, merging, and turbulence effects.

Acknowledgment

The authors would like to thank the financial support of the PETROBRAS Research Center and MCT/CNPq, The Brazilian Council for Scientific Research. The Center for Parallel Computations (NACAD) at the Federal University of Rio de Janeiro provided the computational resources for this research. We are also grateful to Professor Harry Yeh of UW-Seattle who kindly provided us the experimental data for the problem of wave impact with a tall structure.

References

[1] Tezduyar, T. E., and Osawa, Y., 2000, "Finite Element Stabilization Parameters Computed From Element Matrices and Vectors," Comput. Methods Appl. Mech. Eng., **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. Eng., **95**, pp. 221–242.

[3] Tezduyar, T. E., 2001, "Finite Element Methods for Flow Problems With Moving Boundaries and Interfaces," Arch. Comput. Methods Eng., **8**, pp. 83–130.

[4] Koshizuka, S., Tamako, H., and Oka, Y., 1995, "A Particle Method for Incompressible Viscous Flow With Fluid Fragmentation," Comput. Fluid Dyn. J., **4**, pp. 29–46.

[5] Violeau, D., and Issa, R., 2007, "Numerical Modelling of Complex Turbulent Free-Surface Flows With the SPH Method: An Overview," Int. J. Numer. Methods Fluids, **53**, pp. 277–304.

[6] Del Pin, F., Idelsohn, S., Oñate, E., and Aubry, R., 2007, "The ALE/Lagrangian Particle Finite Element Method: A New Approach to Computation of Free-Surface Flows and Fluid-Object Interactions," Comput. Fluids, **36**, pp. 27–38.

[7] Takizawa, K., Yabe, T., Tsugawa, Y., Tezduyar, T. E., and Mizoe, H., 2007, "Computation of Free-Surface Flows and Fluid-Object Interactions With the CIP Method Based on Adaptive Meshless Soroban Grids," Comput. Mech., **40**, pp. 167–183.

[8] Hirt, C. W., and Nichols, B. D., 1981, "Volume of Fluid (VOF) Methods for the Dynamics of Free Boundaries," J. Comput. Phys., **39**, pp. 201–225.

[9] Tezduyar, T. E., Aliabadi, S., and Behr, M., 1997, "Enhanced-Discretization Interface-Capturing Technique," Proceedings of the ISAC'97 High Performance Computing on Multiphase Flows, Y. Matsumoto and A. Prosperetti, eds., Vol. 1–6.

[10] Lohner, R., Yang, C., and Onate, E., 2006, "On the Simulation of Flows With Violent Free-Surface Motion," Comput. Methods Appl. Mech. Eng., **195**, pp. 5597–5620.

[11] 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.

[12] Elias, R. N., and Coutinho, A. L. G. A., 2007, "Stabilized Edge-Based Finite Element Simulation of Free-Surface Flows," Int. J. Numer. Methods Fluids, **54**, pp. 965–993.

[13] Smagorinsky, J., 1963, "General Circulation Experiments With Primitive Equations, Part I: The Basic Experiment," Mon. Weather Rev., **91**, pp. 99–152.

[14] Galeão, A. C., and do Carmo, E. G. D., 1988, "A Consistent Approximate Upwind Petrov-Galerkin Method for Convection-Dominated Problems," Comput. Methods Appl. Mech. Eng., **68**, pp. 83–95.

[15] 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. Eng., **32**, pp. 199–259.

[16] Bazilevs, Y., Calo, V. M., Tezduyar, T. E., and Hughes, T. J. R., 2007, "$YZ\beta$ Discontinuity Capturing for Advection-Dominated Processes With Application to Arterial Drug Delivery," Int. J. Numer. Methods Fluids, **54**, pp. 593–608.

[17] Franca, L. P., and Frey, S. L., 1992, "Stabilized Finite Element Methods: II. The Incompressible Navier-Stokes Equations," Comput. Methods Appl. Mech. Eng., **99**, pp. 209–233.

[18] 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," Comput. Mech., **38**, pp. 365–381.

[19] 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. Numer. Methods Fluids, **47**, pp. 201–231.

[20] 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," Comput. Methods Appl. Mech. Eng., **195**, pp. 3145–3167.

[21] 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," Int. J. Numer. Methods Eng., **66**, pp. 431–460.

[22] 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," Lect. Notes Comput. Sci., **4395**, pp. 292–304.

[23] 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," Lect. Notes Comput. Sci., **3648**, pp. 1237–1245.

[24] Karypis, G., and Kumar, V., 1998, "METIS, A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices," University of Minnesota, Minneapolis, http://glaros.dtc.umn.edu/gkhome/views/metis.

[25] Lohner, R., Yang, C., and Onate, E., 2006, "On the Simulation of Flows With Violent Free-Surface Motion," Comput. Methods Appl. Mech. Eng., **195**, pp. 5597–5620.

[26] Lohner, R., and Camelli, F., 2004, "Dynamic Deactivation for Advection-Dominated Contaminant Transport," Commun. Numer. Methods Eng., **20**, pp. 639–646.

[27] Radovitzky, R., and Ortiz, M., 1998, "Lagrangian Finite Element Analysis of Newtonian Fluid Flows," Int. J. Numer. Methods Eng., **43**, pp. 607–619.

[28] Laitone, E. V., 1960, "The Second Approximation to Cnoidal and Solitary Waves," J. Fluid Mech., **9**, pp. 430–444.

[29] Gómez-Gesteira, M., and Dalrymple, R. A., 2004, "Using a Three-Dimensional Smoothed Particle Hydrodynamics Method for Wave Impact on a Tall Structure," J. Waterway, Port, Coastal, Ocean Eng., **130**, pp. 63–69.

[30] Wu, T. R., 2004, "A Numerical Study of Three-Dimensional Breaking Waves and Turbulence Effects," Ph.D. thesis, Cornell University, Ithaca, New York.

[31] Raad, P., "Mitigation of Local Tsunami Effects," http://engr.smu.edu/waves/project.html.