

A PML-Based Nonreflective Boundary for Free Surface Fluid Animation

ANDREAS SÖDERSTRÖM and MATTS KARLSSON

Linköping University

and

KEN MUSETH

DreamWorks Animation and Linköping University

This article presents a novel nonreflective boundary condition for the free surface incompressible Euler and Navier-Stokes equations. Boundaries of this type are very useful when, for example, simulating water flow around a ship moving over a wide ocean. Normally waves generated by the ship will reflect off of the boundaries of the simulation domain and as these reflected waves return towards the ship they will cause undesired interference patterns. By employing a Perfectly Matched Layer (PML) approach we have derived a boundary condition that absorbs incoming waves and thus efficiently prevents these undesired wave reflections. To solve the resulting boundary equations we present a fast and stable algorithm based on the stable fluids approach. Through numerical experiments we then show that our boundaries are significantly more effective than simpler reflection preventing techniques. We also provide a thorough analysis of the parameters involved in our boundary formulation and show how they effect wave absorption efficiency.

Categories and Subject Descriptors: G.1.7 [Numerical Analysis]: Ordinary Differential Equations; I.6.4 [Simulation and Modeling]: Model Validation and Analysis; I.6.5 [Simulation and Modeling]: Model Development; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; J.2 [Physical Sciences and Engineering]: Physics

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Computational fluid dynamics, free surface, stable fluids, Euler equations, Navier-Stokes equations, nonreflecting boundary condition, perfectly matched layer

ACM Reference Format:

Söderström, A., Karlsson, M., and Museth, K. 2010. A PML-based nonreflective boundary for free surface fluid animation. *ACM Trans. Graph.* 29, 5, Article 136 (October 2010), 17 pages. DOI = 10.1145/1857907.1857912 <http://doi.acm.org/10.1145/1857907.1857912>

1. INTRODUCTION

In recent years there has been an increase in movie visual effects based on Computational Fluid Dynamics (CFD). The most common CFD effects are computer generated fire and smoke but high-quality water animations have also appeared in several blockbuster movies. At the core of these effects are typically the incompressible Navier-Stokes equations. However, for some phenomena, like water, the effect of viscosity is sometimes ignored and the Euler equations are solved instead. Though many methods exist for solving the Navier-Stokes equations [Monaghan 1988; He and Luo 1997; Zhu and Bridson 2005] among others, grid-based Eulerian solvers tend to be very popular when high quality results are desired. CFD calculations in general, and Eulerian solvers in particular are, however, very computationally expensive. Consequently it is desirable to limit the volume in which the simulation takes place, that is, the simulation domain. Using a small domain can, however, cause its own problems. One of these, undesired wave reflection, is the focus of this article.

Consider as an example a ship moving over a wide ocean; the ship will generate waves as it pushes through the water and these waves will travel outwards away from the ship. Close to the vessel we want a realistic fluid simulation that accurately captures the physics of this scenario: the waves breaking around the bow, for example. This requires a fairly accurate and thus typically slow simulation method. In order to complete the simulation within a reasonable timeframe we need to limit our simulation domain to the close surroundings of the ship. However, the waves generated by the ship will eventually reach and reflect off of the simulation domain boundaries. These reflected waves can easily return to the region of interest close to the ship causing wave patterns (i.e., interference) that should not exist for a lone vessel on an open ocean. An example of such a scenario is depicted in Figure 1. Note the distinctly different behavior of the fluid along the boundaries of the “walled in” reference simulation (left) and the “true” open ocean simulation (right).

In physics this type of problem is often encountered for compressible fluid simulations, for example, when simulating airflow

Authors' addresses: A. Söderström (corresponding author), M. Karlsson, K. Museth, Linköping University, SE-581 83 Linköping, Sweden; email: andreas.soderstrom@bredband.net.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 0730-0301/2010/10-ART136 \$10.00 DOI 10.1145/1857907.1857912 <http://doi.acm.org/10.1145/1857907.1857912>



Fig. 1. An example of the difference between a walled-in fluid simulation (left) and an open ocean simulation (right).

around an aircraft as is done in Hein et al. [2007] and Özyörük [2009] among many others. The typical solution is to employ non-reflecting far-field boundaries instead of simple walls at the edge of the simulation domain. One of the best known realizations of such a nonreflecting boundary is the Perfectly Matched Layer (PML) approach [Berenger 1994; Richards et al. 2004; Johnson 2007]. In physics one often deals with a continuous compressible medium covering the entire simulation domain: wind, weather, and acoustic simulations, for example. However, in this article we will instead focus on developing a PML-based, nonreflecting boundary condition for visual-effects-oriented problems like the ship on open water. This includes deriving a PML boundary formulation for the incompressible Navier-Stokes equations as well as properly treating the dynamic fluid surface now present. We also aim to construct a method for solving the resulting equations that is fast and stable enough to be useful for visual effects.

In Section 4 of this article we derive such a boundary condition by extending the work of Hu et al. [2008] to incompressible fluids in the presence of an external force field. In Section 5 we then describe how to apply the resulting equations to free surface fluid animation. This includes the steps necessary to handle the fluid surface as well as how to properly deal with the transition from the regular simulation domain to the reflection preventing boundary region. In Section 6 we present an algorithm for solving the relevant equations based on the popular stable fluids approach [Stam 1999; Foster and Fedkiw 2001; Fedkiw et al. 2001; Enright et al. 2002]. Section 7 presents two additional methods for achieving low reflection boundaries: explicit and implicit dampening. In Section 8 we provide numerical examples aimed towards measuring and validating the effectiveness of our PML boundary using explicit and implicit dampening as comparisons. Finally we provide concluding remarks in Section 9.

2. PREVIOUS WORK

The first perfectly matched layer approach was introduced by Berenger in 1994 for computational electromagnetics [Berenger 1994]. Berenger used a split-variable formulation that was later shown to be dynamically stable but only weakly well-posed [Abarbanel and Gottlieb 1997]. Since then it has been shown that the PML method is equivalent to a complex change of variables in the frequency domain and that the PML equations can be formulated in unsplit physical variables [Chew and Weedon 1994; Turkel and Yefet 1998; Zhao and Cangellaris 1996]. The PML method was first applied to computational fluid dynamics and computational aeroacoustics starting with the linearized Euler equations in Hu [1996]. In Hu [2001b] it was later found that a necessary condition for the PML method to work in this scenario was that all physical waves have consistent phase and group velocities. This conclusion was also reached independently in Bécache et al. [2003]. In response to this issue several new PML formulations have been developed. In

Hu [2001a] a stable PML method for the linearized Euler equations in the presence of a uniform mean-flow was proposed. Equivalent methods have also been presented in Bécache et al. [2004] and Hagstrom and Nazarov [2002, 2003]. Recently the PML method has also been applied to the nonlinear Euler [Hu 2006] and Navier-Stokes equations [Hagstrom et al. 2005; Hu et al. 2008].

It is interesting to note that the fundamental idea of deriving absorbing boundary conditions by means of complex scaling of coordinates, which forms the very basis of the PML, for example, Eq. (5), originates outside the field of computational fluid dynamics. While it is virtually impossible to track its exact origin, this idea of complex scaling has existed in quantum dynamics for a long time. Specifically, it has successfully been applied to the study of atomic and molecular dynamics in Reinhardt [1982], Neuhasuer and Baer [1989], Museth and Leforestier [1996] among others.

In the fields of computer graphics and fluid animation, simple open boundaries for incompressible gas and smoke simulation have been widely used for some time. Examples include Stam [1999] and Fedkiw et al. [2001]. Compressible gas and smoke simulation is not widely used in computer graphics, however, PML boundaries concerning this problem have been well studied in the field of computational acoustics as presented before. To the extent of our knowledge nontrivial open boundaries for free surface incompressible flow were first presented in Söderström and Museth [2009] which constitutes preliminary results of the current article.

3. CONTRIBUTIONS

Our approach for deriving a nonreflective boundary condition for the Navier-Stokes equations is primarily based on previous work by Hu et al. [2008]. However, we make several novel contributions. First of all we show how to extend PML-type boundaries to incompressible fluids in the presence of a free surface. We also derive the equations necessary to handle conservative forces, like gravity, in the boundary domain. Furthermore we present a novel algorithm for solving the resulting equations based on the stable fluids approach. Though our algorithm is not unconditionally stable, we describe a specialized time integration scheme that makes the stability condition independent of the boundary, allowing for fast and stable simulations regardless of the amount of wave dampening applied. Finally we present a thorough analysis of the effect of the various parameters involved in our boundary model as well as describing how to use both explicit and implicit geometry to represent the boundary itself.

4. DERIVING A PML BOUNDARY CONDITION FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

In this section we will derive a formulation of the Navier-Stokes equations that will be able to efficiently absorb any waves present in the solution. The goal is to obtain wave dampening equations that are matched to the behavior of the Navier-Stokes equations, thus achieving better dampening results than simpler techniques. We will deploy these matched equations in a boundary layer neighboring a region where the regular Navier-Stokes equations are solved, thus obtaining a boundary capable of preventing reflections. As we will see, this approach results in directional dampening of the Navier-Stokes momentum equations. We will also see that even though we are dealing with an incompressible fluid, our approach leads to artificial sources and sinks in the boundary regions. This pseudocompressible behavior allows our method to efficiently absorb incoming waves by temporarily absorbing mass as well as energy.

We start the derivation of our boundary condition by considering the Navier-Stokes equations for incompressible flow, commonly written as

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \frac{\mathbf{f}}{\rho} + \nu \nabla^2 \mathbf{v} - \frac{\nabla p}{\rho}, \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (2)$$

where $\mathbf{v} = (v_x, v_y, v_z)$ is the fluid velocity vector field, p represents pressure, \mathbf{f} denotes external forces, ν is the kinematic viscosity, and ρ is the density of the fluid. Eqs. (2) and (1) can also be written in conservation form, resulting in the equation

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}_1}{\partial x} + \frac{\partial \mathbf{F}_2}{\partial y} + \frac{\partial \mathbf{F}_3}{\partial z} = \mathbf{f}, \quad (3)$$

where

$$\mathbf{F}_1 = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p - \tau_{xx} \\ \rho v_x v_y - \tau_{yx} \\ \rho v_x v_z - \tau_{zx} \end{pmatrix}, \quad \mathbf{F}_2 = \begin{pmatrix} \rho v_y \\ \rho v_x v_y - \tau_{xy} \\ \rho v_y^2 + p - \tau_{yy} \\ \rho v_y v_z - \tau_{zy} \end{pmatrix},$$

$$\mathbf{F}_3 = \begin{pmatrix} \rho v_z \\ \rho v_x v_z - \tau_{xz} \\ \rho v_y v_z - \tau_{yz} \\ \rho v_z^2 + p - \tau_{zz} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} 0 \\ f_x \\ f_y \\ f_z \end{pmatrix}$$

and

$$\tau_{xx} = 2\rho\nu \frac{\partial v_x}{\partial x_1}, \quad \tau_{yy} = 2\rho\nu \frac{\partial v_y}{\partial x_2}, \quad \tau_{zz} = 2\rho\nu \frac{\partial v_z}{\partial x_3},$$

$$\tau_{xy} = \rho\nu \left(\frac{\partial v_x}{\partial x_2} + \frac{\partial v_y}{\partial x_1} \right), \quad \tau_{xz} = \rho\nu \left(\frac{\partial v_x}{\partial x_3} + \frac{\partial v_z}{\partial x_1} \right),$$

$$\tau_{yz} = \rho\nu \left(\frac{\partial v_y}{\partial x_3} + \frac{\partial v_z}{\partial x_2} \right).$$

Note that since we have assumed incompressibility we also know that $\rho(x, y, z, t) = \rho_0$ where ρ_0 is the initial density field.

Our derivation that follows will be fairly math intensive, however, many of the calculations will be similar for different vector components and/or spatial directions. Thus in order to reduce the number of equations and the size of each equation we will throughout this article use the following index notation: The index $\alpha \in \{0, 1, 2, 3\}$ will be used to enumerate vector components and the index $\beta \in \{1, 2, 3\}$ denotes spatial directions. Thus the notation $q_{\alpha,\beta}$ will describe 12 different variables, one for each vector component and direction. The notation $x_\beta \equiv \{x_1, x_2, x_3\}$ represents the three spatial variables commonly referred to as $x, y,$ and z . Using this notation the incompressible Navier-Stokes equations in conservation form can be written as

$$\frac{\partial u_\alpha}{\partial t} + \sum_\beta \frac{\partial F_{\alpha,\beta}}{\partial x_\beta} = f_\alpha. \quad (4)$$

Since the index $\alpha = \{0, 1, 2, 3\}$ Eq. (4) actually describes four equations: one for each vector component. The notation $F_{\alpha,\beta}$ denotes vector component α of flux vector β , thus, for example, $F_{1,3} = \rho v_x v_z - \tau_{xz}$. This notation is essentially equivalent to the index notation typically used for matrices and vectors and in this context it may be helpful to think of $F_{\alpha,\beta}$ as a matrix where each flux vector constitutes one column. It is important to understand that Eqs. (3) and (4) are equivalent since we will make heavy use of this type of notation throughout the article.

The derivation to follow is based on the work of Hu et al. [2008], however, we will make a novel contribution in order to handle the presence of external forces. For the scope of this article we will also

limit ourselves to PML boundary conditions without the presence of a mean-flow in the boundary region.

Our approach to obtaining a PML boundary formulation for the incompressible Navier-Stokes equations is based on applying a change of variables in the frequency domain. This approach has successfully been used to realize PML-type boundaries in Hu et al. [2008], Berenger [1994], Richards et al. [2004], and Hu [2001a] among others. The transformation typically used is

$$x \rightarrow x + \frac{i}{\omega} \int_{x_0}^{x_1} \sigma(x) dx, \quad (5)$$

where ω is the frequency, $\sigma(x)$ is a dampening function to be determined later ($\sigma(x) \geq 0 \forall x$), and x_0 is the position of the boundary between the regular Navier-Stokes equations and the PML boundary region. We will refer to this as the *inner boundary* as opposed to the solid wall *outer boundaries* surrounding the simulation domain. The width of the PML boundary region is $|x_1 - x_0|$. To give an intuition for the behavior of this transformation, consider a complex wave component of the form

$$e^{i(kx - \omega t)}. \quad (6)$$

If we apply (5) to (6) we obtain

$$e^{i(kx - \omega t)} e^{-\frac{k}{\omega} \int_{x_0}^{x_1} \sigma_x dx}. \quad (7)$$

As can be seen the second term of Eq. (7) represents exponential decay, assuming $\frac{k}{\omega} \geq 0$ and $\sigma_x \geq 0$, that is, we can achieve exponential dampening of wave components moving through the region where the transformation (5) is used. We note that there are some stability issues involved in guaranteeing $\frac{k}{\omega} \geq 0$ when a mean-flow is present, a scenario well described in Hu [2008] among others. However, since we assume that we have no mean-flow in the boundary region this is not an issue we will address in this article.

We will now derive the PML formulation for Eq. (4). We start by writing the integral Eq. (5) as its differential equivalent

$$\frac{\partial}{\partial x_\beta} \rightarrow \frac{1}{1 + i \frac{\sigma_\beta}{\omega}} \frac{\partial}{\partial x_\beta}. \quad (8)$$

Since Eq. (8) operates on spatial differentials it will prove convenient to introduce the vectors

$$\mathbf{e}_\beta = \frac{\partial \mathbf{v}}{\partial x_\beta}. \quad (9)$$

We may now consider the flux vectors \mathbf{F}_β as functions of \mathbf{u} and \mathbf{e}_β . At this point we transform Eqs. (4) and (9) to the frequency domain and apply the transformation (8). This yields the equations

$$-i\omega \tilde{u}_\alpha + \sum_\beta \frac{1}{1 + i \frac{\sigma_\beta}{\omega}} \frac{\partial \tilde{F}_{\alpha,\beta}}{\partial x_\beta} = \tilde{f}_\alpha, \quad (10)$$

$$\tilde{\mathbf{e}}_\beta = \frac{1}{1 + i \frac{\sigma_\beta}{\omega}} \frac{\partial \tilde{\mathbf{v}}}{\partial x_\beta}, \quad (11)$$

where we have used the symbol $\tilde{\cdot}$ to denote the frequency transform. Note that Eq. (10) assumes that all wave components can be written in the form $e^{i(g(x) - \omega t)}$ for some arbitrary function g . Our next step is to transform these equations back to the spatial domain. However, due to the presence of an external force term on the right-hand side of Eq. (10) we can no longer directly follow the approach of Hu et al. [2008]. At this point we note that though complex external force interactions may be present in the regular simulation domain this will typically be some distance from the actual walls of the domain. Based on this we make the assumption that all external forces *in the*

PML boundary region are conservative. This obviously includes the force of gravity but could potentially also be used for other types of interactions. Any conservative force \mathbf{f} can be written in the form

$$\mathbf{f} = -\nabla U, \quad (12)$$

where U is the potential field associated with \mathbf{f} . By applying Eq. (12) to Eq. (4) we can rewrite Eq. (10) as

$$-i\omega\tilde{u}_\alpha + \sum_\beta \frac{1}{1+i\frac{\sigma_\beta}{\omega}} \frac{\partial(\tilde{F}_{\alpha,\beta} + \delta_{\alpha,\beta}\tilde{U})}{\partial x_\beta} = 0, \quad (13)$$

where $\delta_{\alpha,\beta}$ is the Kronecker delta symbol, that is, $\delta_{\alpha,\beta} = 1$ if $\alpha = \beta$ and 0 otherwise. Eqs. (11) and (13) are now in a form that can be transformed back to their spatial representation by employing the split variable approach described in Berenger [1994], and Hu [1995] among others. We start by introducing the auxiliary variables $q_{\alpha,\beta}$ which satisfies the equations

$$u_\alpha = \sum_\beta q_{\alpha,\beta} \quad (14)$$

$$-i\omega\tilde{q}_{\alpha,\beta} + \frac{1}{1+i\frac{\sigma_\beta}{\omega}} \frac{\partial\tilde{F}_{\alpha,\beta} + \delta_{\alpha,\beta}\tilde{U}}{\partial x_\beta} = 0. \quad (15)$$

Here the variables $q_{\alpha,\beta}$ can be interpreted as a vector density. It will soon be evident that these potentially time-dependant density components results in the pseudocompressible behavior we described in the beginning of this section. Consequently the variables $q_{2,\beta}$, $q_{3,\beta}$, and $q_{4,\beta}$ can be interpreted as components of a velocity tensor field. We will later see that this tensor velocity allows for directional dampening of the Navier-Stokes momentum equation. The relation between the $q_{\alpha,\beta}$ tensor and the components $\{v_x, v_y, v_z\}$ of the velocity vector \mathbf{v} can be obtained by tensor contraction of $q_{\alpha,\beta}$ along the directional index β , that is, Eq. (14). The split of Eq. (13) allows us to multiply the left- and right-hand side of Eqs. (11) and (15) with $1+i\frac{\sigma_\beta}{\omega}$. This yields

$$(-i\omega + \sigma_\beta)\tilde{q}_{\alpha,\beta} + \frac{\partial\tilde{F}_{\alpha,\beta} + \delta_{\alpha,\beta}\tilde{U}}{\partial x_\beta} = 0 \quad (16)$$

$$\left(1+i\frac{\sigma_\beta}{\omega}\right)\tilde{\mathbf{e}}_\beta = \frac{\partial\tilde{\mathbf{v}}}{\partial x_\beta}. \quad (17)$$

To transform Eq. (17) into a partial differential equation we now introduce the auxiliary variables \mathbf{r}_β , such that

$$\frac{\partial\mathbf{r}_\beta}{\partial t} = \mathbf{e}_\beta. \quad (18)$$

After applying Eq. (18) to (17) we finally perform the transformation back to the spatial domain and thus obtain the complete set of equations describing the PML boundary condition for the incompressible Navier-Stokes equations in the presence of a conservative external force.

$$\frac{\partial q_{\alpha,\beta}}{\partial t} + \sigma_\beta q_{\alpha,\beta} + \frac{\partial F_{\alpha,\beta} + \delta_{\alpha,\beta}U}{\partial x_\beta} = 0 \quad (19)$$

$$\frac{\partial\mathbf{r}_\beta}{\partial t} + \sigma_\beta\mathbf{r}_\beta - \frac{\partial\mathbf{v}}{\partial x_\beta} = 0 \quad (20)$$

$$\mathbf{e}_\beta = \frac{\partial\mathbf{v}}{\partial x_\beta} - \sigma_\beta\mathbf{r}_\beta \quad (21)$$

In addition to these equations we also have Eq. (14) describing the relation between $q_{\alpha,\beta}$ and u_α .

The preceding set of equations can also be written in a somewhat simpler and more familiar form by combining (19), (20), (21), and (14). We have

$$\frac{\partial\mathbf{u}}{\partial t} + \frac{\partial\mathbf{F}_1}{\partial x_1} + \frac{\partial\mathbf{F}_2}{\partial x_2} + \frac{\partial\mathbf{F}_3}{\partial x_3} + \sigma_1\mathbf{q}_1 + \sigma_2\mathbf{q}_2 + \sigma_3\mathbf{q}_3 = \mathbf{f} \quad (22)$$

or

$$\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{\mathbf{f}}{\rho} + (\nabla \cdot \boldsymbol{\tau})^T - \frac{\nabla p}{\rho} - \sigma_1\mathbf{A}\mathbf{q}_1 - \sigma_2\mathbf{A}\mathbf{q}_2 - \sigma_3\mathbf{A}\mathbf{q}_3 \quad (23)$$

$$\nabla \cdot \mathbf{v} = -\sigma_1\mathbf{B}\mathbf{q}_1 - \sigma_2\mathbf{B}\mathbf{q}_2 - \sigma_3\mathbf{B}\mathbf{q}_3 \quad (24)$$

where

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{B} = (1 \ 0 \ 0 \ 0). \quad (25)$$

As can be seen, the four equations in (3) have become twelve equations for the $q_{\alpha,\beta}$ fields, nine equations for the \mathbf{r}_β fields, and nine equations for the \mathbf{e}_β fields for a grand total of 30 equations. Additionally we have the three equations in (14) relating $q_{\alpha,\beta}$ and \mathbf{v} . This obviously increases the computational complexity when solving Eqs. (19) through (21), however, as we will show in Section 6 we can solve these equations approximatly. This will reduce the number of equations and unknowns to 12, thus significantly reducing the computational complexity and memory overhead of our method. Still, the task of solving the system of equations presented earlier may appear daunting. However, the solver presented in Section 6 will prove to be rather similar to the familiar stable fluids method, showing that PML boundaries can be employed without much complexity added to one of the common solution methods used in computer graphics.

5. FREE SURFACE FLUID SIMULATION WITH PML BOUNDARIES

Solving Eqs. (19) through (21) in the compressible case, without the presence of a free surface or external forces has been done in Hu et al. [2008] using the viscous aeroacoustic approach [Li and Gao 2005, 2008] together with a dispersion-relation-preserving scheme [Tam and Webb 1993] and a high-order Runge-Kutta scheme [Hu et al. 1996]. However, we now seek a way to solve these equations that is more appropriate for visual effects, that is, a scheme for *incompressible* fluids that is stable, fast, and accurate enough to yield visually pleasing results. Furthermore we want to be able to handle external forces and the presence of a free surface. Shortly we will outline our algorithm for solving Eqs. (19) through (21) under these conditions.

5.1 Boundary Conditions for the Velocity Tensor

Next we shall derive appropriate Dirichlet boundary conditions for the auxiliary variables $q_{\alpha,\beta}$ at solid boundaries. The Dirichlet boundary condition for the velocity field \mathbf{v} at a solid boundary reads

$$\mathbf{v} \cdot \mathbf{n}_s = \mathbf{v}_s \cdot \mathbf{n}_s. \quad (26)$$

This simply states that the fluid velocity \mathbf{v} along the boundary surface normal \mathbf{n}_s should equal the velocity of the boundary \mathbf{v}_s along the same normal. Thus we can extrapolate this condition to the velocity-related components of the tensor field $q_{\alpha,\beta}$, that is, the

components where $\alpha > 0$, leading to the equation

$$q_{\alpha,\beta} n_\alpha^s = q_{\alpha,\beta}^s n_\alpha^s, \quad \alpha > 0, \quad (27)$$

where n_α^s are the components of the boundary surface normal and $q_{\alpha,\beta}^s$ is the tensor velocity of the boundary. Here we simply state that for the fluid tensor field neighboring a solid surface the vector associated with the plane to which n_α^s is a normal component equals the corresponding vector for the solid boundary. For the velocity tensor $\mathbf{T} = q_{\alpha,\beta}$, $\alpha > 0$ we can write this condition as

$$\mathbf{n}_s^T \mathbf{T} = \mathbf{n}_s^T \mathbf{T}_s. \quad (28)$$

Here \mathbf{n}_s^T denotes the transposed solid surface normal.

Due to the presence of a free surface we also need to apply an appropriate boundary condition at the fluid/air interface. In this case we will use a continuative boundary condition, that is, we assume that the motion of the medium surrounding our fluid is completely governed by the motion of the fluid itself. Thus we can apply the tensor equivalent of the velocity extension algorithm described in Osher and Fedkiw [2002] among others. This amounts to solving the equation

$$\frac{\partial q_{\alpha,\beta}}{\partial \tau} + \mathbf{n} \cdot \nabla q_{\alpha,\beta} = 0, \quad (29)$$

where \mathbf{n} is the fluid surface normal and τ is the fictious time used to propagate the solution to a steady state. The solution to this equation will guarantee that $q_{\alpha,\beta}$ outside the fluid is constant along the direction of the fluid surface normal.

5.2 The Dampening Function σ

The transition from the regular simulation domain to the PML boundary region can be achieved in several ways. The simplest is to assume that the dampening function σ is a step function. However, this is problematic since Eq. (21) will lead to terms of the form $\frac{\partial \sigma_{\beta} v_{\alpha,\beta}}{\partial x_{\beta}}$ which makes the choice of a discontinuous σ function inappropriate. Furthermore we suspect that a smooth transition will provide better numerical results. In Richards et al. [2004] a transfer function of the form

$$\sigma(x) = \frac{x^\gamma}{\epsilon} \quad (30)$$

was used. The optimal choice of γ was found to be problem dependant, however, a choice of $\gamma \approx 2$ generally provided good results. However, due to the exponential nature of (7) we suspect that a function more related to exponential growth might provide better results. Thus we also introduce two trigonometric-based dampening functions. The first is a smooth step function of height 1 and width ϵ .

$$\sigma(x) = \begin{cases} 0 & x < 0 \\ 1 & x > \epsilon \\ \frac{x}{\epsilon} - \frac{1}{\pi} \sin\left(\frac{2\pi x}{\epsilon}\right) & x \in [0, \epsilon] \end{cases} \quad (31)$$

In Section 8.1 we will evaluate this choice of function through numerical experiments. We will also investigate the ‘‘half step’’ function

$$\sigma(x) = \begin{cases} 0 & x < 0 \\ 1 & x > \epsilon \\ \frac{x}{\epsilon} - \frac{1}{\pi} \sin\left(\frac{\pi x}{\epsilon}\right) & x \in [0, \epsilon] \end{cases} \quad (32)$$

as well as the function (30) with $\gamma = 2$.

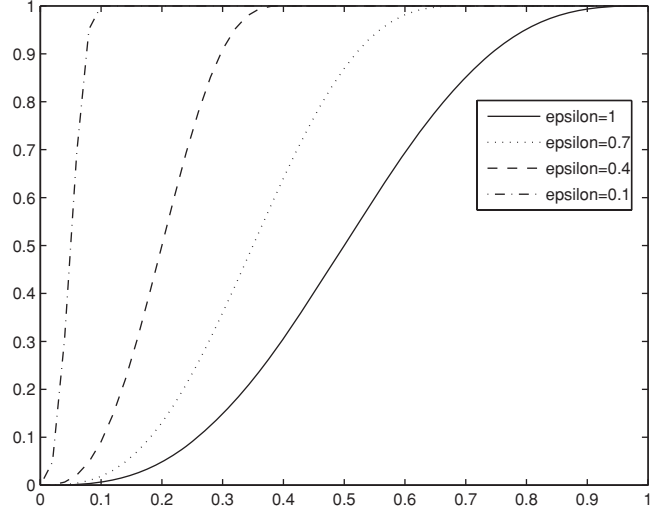


Fig. 2. Smooth step dampening function for different values of ϵ .

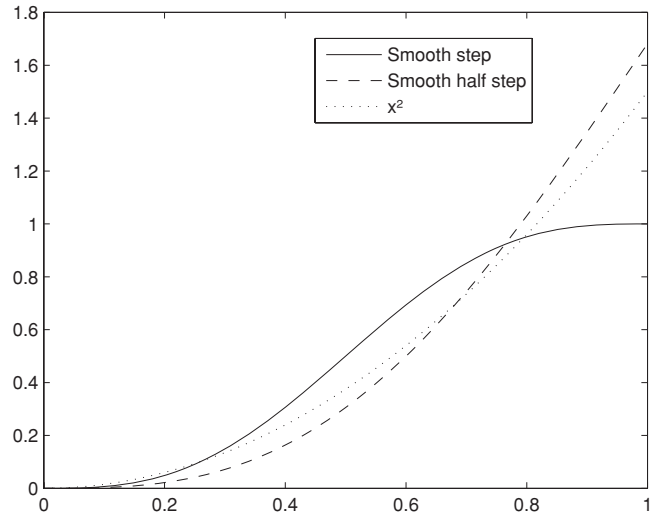


Fig. 3. Comparison between the different transfer functions described in Section 5.2. For the step functions $\epsilon = 1.0$ has been used. All functions have been normalized such that $\int_0^1 \sigma(x) dx = 0.5$.

The shape of function (31) for different values of ϵ can be seen in Figure 2 and a comparison between the three types of transfer function mentioned above is shown in Figure 3.

5.3 Estimating Maximum Dampening

Eq. (7) indicates that our PML boundary condition can theoretically yield an exponentially decaying solution in the boundary region. Thus we assume that the height of an incoming surface wave will be dampened exponentially as it travels through the domain. Based on this we approximate the amount of dampening needed to reduce

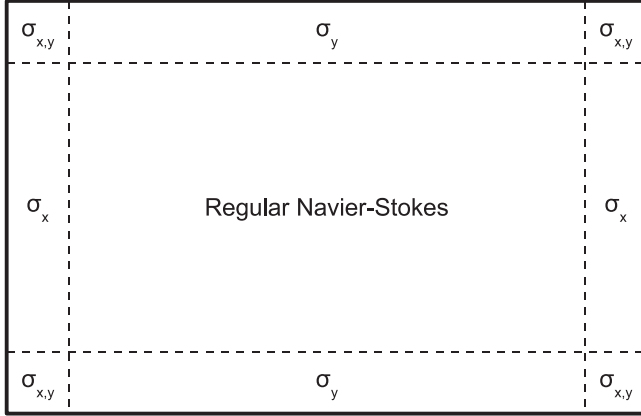


Fig. 4. An example of boundary zones. The dashed lines are the inner boundaries and the thick outer rectangle is the outer boundaries, that is, the actual physical walls of the simulation domain. In the area in the middle we solve the regular, undamped Navier-Stokes equations.

reflections by a factor D through the equation

$$1 - D = e^{-\frac{k}{\omega} \int_{x_0}^{x_1} \sigma_{max} \sigma(x) dx} \Rightarrow, \quad (33)$$

$$\sigma_{max} = -\frac{\ln(1 - D)}{\frac{k}{\omega} \int_{x_0}^{x_1} \sigma(x) dx}, \quad D \in [0, 1], \quad (34)$$

where L is the width of the relevant boundary zone and D is the desired dampening given as a fraction. $D = 0.9$ means that we desire the boundary to reduce the incoming wave amplitude by 90%. In order to simplify our scheme we want to avoid calculating the fraction $\frac{k}{\omega}$. Thus we will assume

$$\frac{k}{\omega} = 1, \quad (35)$$

and instead try to estimate the optimal dampening D through experiments in Section 8.1. This approach essentially assumes that we have a nondispersive medium, which typically is not true. Thus the optimal σ_{max} will be problem dependant under this assumption. However, in Section 8 we will show that a σ_{max} estimated for one scenario can be used with good dampening results in a multitude of similar scenarios.

5.4 Defining the Boundary Regions

For the PML boundary condition to work we need to define buffer zones close to the actual solid wall boundaries. Inside these buffer zones we apply the modified Navier-Stokes equations (19) through (21) derived earlier, thus achieving absorption of waves entering this region. Outside the buffer zones we solve the regular, unmodified Navier-Stokes equations. In order to define these regions we present two methods: an explicit axis-aligned boundary and an implicit boundary for general, complex shapes.

Axis-aligned boundaries. The first method puts emphasis on the directional nature of the PML boundary representation derived before. The $\sigma = \{\sigma_1, \sigma_2, \sigma_3\}$ vector defines the dampening in each Cartesian direction and by placing planes in these directions we can define regions in which only one σ component is nonzero. An example of this can be seen in Figure 4. Note that the corners of the simulation domain will have two nonzero σ components (in 2D).

Level set boundaries. For complex boundary shapes we will use level sets [Adalsteinsson and Sethian 1995; Osher and Fedkiw 2002] to represent the PML boundary region. For compact and efficient storage of boundaries regardless of their complexity we employ the compressed tubular level set framework of Nielsen et al. [2007]. Using level set boundaries we can calculate normals at any point in the boundary domain. This allows us to calculate the amount of dampening applied at a point \mathbf{p} through

$$\sigma = \sigma \mathbf{n}, \quad (36)$$

where σ is given by Eq. (31) and \mathbf{n} is the normal of the level curve that intersects \mathbf{p} .

6. SOLVING THE WAVE ABSORBING NAVIER-STOKES EQUATIONS

In this section we will show how to solve our modified Navier-Stokes equations in the PML boundary region. Our approach will be based on the popular stable fluids method and we will also use the regular stable fluids algorithm to solve for the velocity field outside of our PML boundaries. Note that in the boundary region we will typically deal with the auxiliary variables $q_{\alpha,\beta}$ which can be seen as a tensor velocity field for $\alpha > 0$ and the $q_{0,\beta}$ components can be interpreted as a vector density. The regular velocity field \mathbf{v} can at any time be obtained through Eq. (14). By making use of the operator splitting approach of the stable fluids method our scheme for updating the auxiliary tensor field $\mathbf{Q} \equiv q_{\alpha,\beta}$ one timestep will be

$$\mathbf{Q}_t \xrightarrow{\text{dampen \& advect}} \mathbf{Q}_1 \xrightarrow{\text{viscosity}} \mathbf{Q}_2 \xrightarrow{\text{forces}} \mathbf{Q}_3 \xrightarrow{\text{project}} \mathbf{Q}_{t+\Delta t},$$

where \mathbf{Q}_t is the solution at the beginning of our timestep and $\mathbf{Q}_{t+\Delta t}$ is the new solution to the boundary equations at time $t + \Delta t$.

In order to facilitate this approach we start by noting that Eq. (19) can be written as

$$\begin{cases} \alpha = 0 : \\ \frac{\partial q_{\alpha,\beta}}{\partial t} + \sigma_\beta q_{\alpha,\beta} + \frac{\partial \mathbf{v}_\beta}{\partial x_\beta} = 0 \\ \alpha = 1, 2, 3 : \\ \frac{\partial q_{\alpha,\beta}}{\partial t} + \sigma_\beta q_{\alpha,\beta} + \frac{\partial \mathbf{v}_\alpha \mathbf{v}_\beta}{\partial x_\beta} + \delta_{\alpha,\beta} \frac{1}{\rho} \frac{\partial U_\beta}{\partial x_\beta} - \frac{\partial \tau_{\alpha,\beta}}{\partial x_\beta} + \delta_{\alpha,\beta} \frac{1}{\rho} \frac{\partial p}{\partial x_\beta} = 0. \end{cases} \quad (37)$$

The first row of (37), that is, $\alpha = 0$ corresponds to the conservation of mass and describes the pseudo-compressible nature of our boundaries. This equation will be tightly coupled with the projection step. The remaining three equations of (37), that is, $\alpha = \{1, 2, 3\}$ correspond to the conservation of momentum. By introducing the index $\lambda = \{1, 2, 3\}$ and using the operator splitting approach we can identify the integration steps of the stable fluids algorithm as follows.

—Dampened self-advection:

$$\frac{\partial q_{\lambda,\beta}}{\partial t} + \sigma_\beta q_{\lambda,\beta} + \frac{\partial v_\lambda v_\beta}{\partial x_\beta} = 0$$

—Viscosity:

$$\frac{\partial q_{\lambda,\beta}}{\partial t} - \frac{\partial \tau_{\lambda,\beta}}{\partial x_\beta} = 0$$

—External forces:

$$\frac{\partial q_{\lambda,\beta}}{\partial t} + \delta_{\lambda,\beta} \frac{\partial U}{\partial x_\beta} = 0$$

—Projection:

$$\frac{\partial q_{\lambda,\beta}}{\partial t} + \delta_{\lambda,\beta} \frac{1}{\rho} \frac{\partial p}{\partial x_\beta} = 0 \quad \text{and} \quad \frac{\partial q_{0,\beta}}{\partial t} + \frac{\partial \mathbf{v}_\beta}{\partial x_\beta} = 0$$

We will now go through each of these equations and describe how they can be solved.

6.1 The Dampened Self-Advection Term

The first step of our solution algorithm is to obtain the field \mathbf{Q}_1 from \mathbf{Q}_t by accounting for dampening and self-advection. This amounts to solving the equation

$$\frac{\partial q_{\lambda,\beta}}{\partial t} + \sigma_\beta q_{\lambda,\beta} + \frac{\partial v_\lambda v_\beta}{\partial x_\beta} = 0. \quad (38)$$

We also solve the equation

$$\frac{\partial q_{0,\beta}}{\partial t} + \sigma_\beta q_{0,\beta} + \frac{\partial v_\beta}{\partial x_\beta} = 0, \quad (39)$$

that is, the first equation of (37) during this step. We have as of yet not been able to construct an unconditionally stable integration scheme for Eqs. (38) and (39) and thus for the scope of this article we will focus on solving Eqs. (38) and (39) through explicit integration. However, we will derive an explicit integration scheme with a stability condition that is independent of the dampening σ_β , hence allowing for fast integration even for very large σ_{max} . In Section 8.1.4 we will show that the reflection preventing properties of our boundaries are virtually unaffected by the use of this scheme.

Our first step is to assume that v_β is constant during each timestep, that is, the same assumption employed by the stable fluids method. This allows us to treat Eq. (38) as dampened advection, as opposed to dampened self-advection. Inspired by the analysis in Baldauf [2008] we will solve this equation using third-order TVD Runge-Kutta [Shu and Osher 1988] together with fifth-order WENO upwinding [Liu et al. 1994]. Note that though the scheme to follow is derived for Eq. (38) the equivalent method can be used for solving Eq. (39). We now make the assumption that $\frac{\partial v_\lambda v_\beta}{\partial x_\beta}$ is constant during our timestep. By making this assumption we can reduce Eq. (38) to a first-order ODE to which the solution is

$$q_{\lambda,\beta} = -\frac{C}{\sigma_\beta} + \left(q_{0,\lambda,\beta} + \frac{C}{\sigma_\beta} \right) e^{-\sigma_\beta t}, \quad (40)$$

where $C = \frac{\partial v_\lambda v_\beta}{\partial x_\beta}$ and $q_{0,\lambda,\beta}$ is the value of $q_{\lambda,\beta}$ at the beginning of the current timestep. Eq. (40) can now be seen as a model approximating Eq. (38) for small timesteps. Thus we propose to use this model instead of Eq. (38) when integrating the self-advection term in the boundary region. Note that Eq. (40) is also well behaved for $\sigma_\beta \rightarrow 0$.

Assuming that $q_{\lambda,\beta}$ follows the model in Eq. (40), and σ_β is time independent we next see that

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = \frac{\partial}{\partial t} \left(-\frac{C}{\sigma_\beta} + \left(q_{0,\lambda,\beta} + \frac{C}{\sigma_\beta} \right) e^{-\sigma_\beta t} \right) \Rightarrow \quad (41)$$

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = -q_{0,\lambda,\beta} \sigma_\beta e^{-\sigma_\beta t} - C e^{-\sigma_\beta t} \quad (42)$$

which when discretized can be integrated by parts using the operator splitting approach employed by the stable fluids method. The scheme for updating $q_{\lambda,\beta}$ will be

$$q_{\lambda,\beta}^0 \rightarrow q_{\lambda,\beta}^1 \rightarrow q_{\lambda,\beta}^2,$$

where $q_{\lambda,\beta}^0$ is the initial q field and $q_{\lambda,\beta}^2$ is the resulting field after solving Eq. (42). The field $q_{\lambda,\beta}^1$ can be calculated from $q_{\lambda,\beta}^0$ by solving

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = -q_{0,\lambda,\beta} \sigma_\beta e^{-\sigma_\beta t} \quad (43)$$

and $q_{\lambda,\beta}^2$ can then be calculated from $q_{\lambda,\beta}^1$ by solving

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = -C e^{-\sigma_\beta t}. \quad (44)$$

The assumption of pure advection allows us to write Eq. (44) as

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = -v_\beta e^{-\sigma_\beta t} \frac{\partial v_\lambda}{\partial x_\beta} \quad (45)$$

which when discretized has a CFL stability condition limited by $|v_\beta e^{-\sigma_\beta t}|$. Since we know that $\sigma_\beta \geq 0$ and $t \geq 0$ this condition will at worst be equal to that of the undampened advection. Thus, using the model in Eq. (40) the stability of Eq. (44) is not adversely affected by the size of the dampening coefficient σ_β .

In order to investigate the stability of Eq. (43) we assume that it will be solved using simple first-order Euler integration. This gives us

$$q_{\lambda,\beta}^{t+\Delta t} = (1 - \Delta t \sigma_\beta e^{-\sigma_\beta \Delta t}) q_{0,\lambda,\beta}, \quad (46)$$

where $q_{\lambda,\beta}^{t+\Delta t}$ is our approximate solution to (43) at time $t + \Delta t$ and $q_{0,\lambda,\beta}$ is the solution at time t . We now note that

$$\sigma_\beta \Delta t e^{-\sigma_\beta \Delta t} \in [0, e^{-1}], \quad \forall \sigma_\beta \geq 0, \quad \forall \Delta t \geq 0. \quad (47)$$

Performing Von Neumann stability analysis of Eq. (46) readily shows that it is unconditionally stable since $\Delta t \geq 0$ and $\sigma_\beta \geq 0$. However, note that the combined scheme still has a CFL stability condition due to Eq. (44). As a result we conclude that by integrating Eq. (40) instead of (38) we have obtained a numerical scheme with a stability condition that is not adversely affected by the size of σ_β . The preceding reasoning is valid for first-order time discretization. However, it follows that any higher-order scheme that can be described as a convex combination of such first-order integration steps, for example the TVD Runge-Kutta scheme, will also have a CFL stability condition equivalent to that of this first-order method. For the specific stability conditions of such higher-order schemes consider for example the work by Gottlieb and Shu [1998].

6.2 The Viscosity Term

In order to obtain the field \mathbf{Q}_2 from \mathbf{Q}_1 we need to account for the effect of viscosity through the equation

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = \frac{\partial \tau_{\lambda,\beta}}{\partial x_\beta} \quad (48)$$

which, assuming incompressibility and constant viscosity, can be written as

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = \nu \frac{\partial e_{\lambda,\beta}}{\partial x_\beta}. \quad (49)$$

For incompressible flow this equation is dominantly parabolic and thus numerically troublesome. Though we can solve Eq. (49) through explicit integration we consider this too slow to be practical. The stable fluids approach solves this problem through implicit integration and thus we attempt to do the same. By substituting $e_{\lambda,\beta}$

from Eq. (21) we arrive at the following equation describing the effect of viscosity in the PML boundary.

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = \nu \frac{\partial \left(\frac{\partial v_\lambda}{\partial x_\beta} - \sigma_\beta r_{\lambda,\beta} \right)}{\partial x_\beta} \quad (50)$$

Using Eq. (14) we can rewrite this as

$$\frac{\partial \mathbf{v}}{\partial t} = \nu \nabla^2 \mathbf{v} - \nu \frac{\partial \sigma_1 \mathbf{r}_1}{\partial \bar{x}_1} - \nu \frac{\partial \sigma_2 \mathbf{r}_2}{\partial \bar{x}_2} - \nu \frac{\partial \sigma_3 \mathbf{r}_3}{\partial \bar{x}_3} \quad (51)$$

which tells us that this term consists of the familiar velocity diffusion term and a set of correction terms containing the auxiliary tensor field \mathbf{r}_β . In order to solve for this term in a fast and efficient manner we will make the assumption that the contribution of the correction terms in Eq. (51) is small. That this can be the case is supported by the experimental results in Section 8.2. Thus, for the scope of this article, we will focus only on the diffusive part of Eq. (49) and drop the auxiliary variables \mathbf{r}_β entirely. This greatly reduces the number of auxiliary variables needed by our scheme and leaves us with the equation

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = \nu \frac{\partial^2 v_\lambda}{\partial x_\beta^2} \quad (52)$$

which we solve through implicit integration using the conjugate gradient method.

6.3 External Force Term

At this point we can proceed to calculate the field \mathbf{Q}_3 from \mathbf{Q}_2 by solving the equation

$$\frac{\partial q_{\lambda,\beta}}{\partial t} + \delta_{\lambda,\beta} \frac{\partial U}{\partial x_\beta} = 0. \quad (53)$$

To solve Eq. (53) we use first-order Euler integration if U is time independent and third-order TVD Runge-Kutta otherwise.

6.4 Pressure Correction

As the final step of our solution algorithm we want to account for the effects of internal pressure. In order to do so we employ a modified version of the projection method of Chorin [1968]. First we apply Eq. (14) to calculate the velocity field \mathbf{u} from the tensor field \mathbf{Q}_3 . At this point we note that the right-hand side of Eq. (24) is not zero, implying that the auxiliary variables $q_{0,\beta}$ act as sources and sinks in the boundary region. Thus we interpret them as such and modify the standard projection step with these artificial sources/sinks added to the pressure Poisson equation. This will account for the pseudocompressible nature of our boundaries. The modified projection equation becomes

$$\frac{\Delta t}{\rho} \nabla^2 p = \nabla \mathbf{v} + \sigma_1 \mathbf{B} \mathbf{q}_1 + \sigma_2 \mathbf{B} \mathbf{q}_2 + \sigma_3 \mathbf{B} \mathbf{q}_3 \quad (54)$$

and its solution gives us the field p which can be interpreted as pressure. We can now obtain our final tensor field $\mathbf{Q}_{t+\Delta}$ from \mathbf{Q}_3 by solving the equation

$$\frac{\partial q_{\lambda,\beta}}{\partial t} = -\delta_{\lambda,\beta} \frac{1}{\rho} \frac{\partial p}{\partial x_\beta}. \quad (55)$$

7. ADDITIONAL REFLECTION REDUCING TECHNIQUES

In order to test the performance of our wave absorbing boundary we will compare its effectiveness against two additional reflection

preventing techniques which we denote respectively “explicit dampening” and “implicit dampening” [Richards et al. 2004]. Both these methods are well known in the field of physics but to the extent of our knowledge neither of them has previously been applied to free surface incompressible flow.

7.1 Explicit Dampening

As with our PML-based method this technique also relies on boundary zones in which alternative physics is applied in order to achieve dampening of wave reflections. Explicit dampening is a very simple wave dampening method that works by direct modification of the velocity field. Though high-quality results can be obtained through this method large boundary zones are typically required for it to be effective [Richards et al. 2004]. The explicit dampening boundary condition can be formulated as

$$\mathbf{u}_{new} = \mathbf{v} - (\mathbf{v} - \mathbf{v}_{target})\sigma, \quad (56)$$

where \mathbf{v} is the velocity field, \mathbf{v}_{target} is the target velocity at the solid boundary, and σ is a spatially varying function that is zero outside the boundary region and one close to the actual physical boundaries, of the simulation domain. Eq. (56) is applied at every iteration of the fluid solver and achieves dampening by simply reducing the velocity vectors in the boundary region. In Richards et al. [2004] a transfer function of the form

$$\sigma = \left(\frac{d}{L} \right)^\gamma \quad (57)$$

was used for dampening the compressible Euler equations. In Eq. (57) d is the distance from the outer boundary (the wall) and L is the width of the dampening region. The optimal value for γ is problem dependant. However, as demonstrated in Richards et al. [2004], $\gamma \in [2, 2.5]$ appears to be an overall good choice and thus we use this transfer function with $\gamma = 2$ in our tests. A significant problem with the explicit dampening method is to determine the optimal target velocity \mathbf{v}_{target} . Ideally this velocity at the outer boundary should be such that a reflected wave is completely dampened out the moment before it reaches the inner boundary again. However, due to the unpredictable dampening performance of the explicit method such a target velocity is nontrivial to estimate in general. Thus we find that the best choice for a general scenario is $\mathbf{v}_{target} = \mathbf{0}$, that is, that all wave motion at the *outer boundary* has been completely dampened out. Thus our explicit dampening formulation becomes

$$\mathbf{u}_{new} = \mathbf{u} - \mathbf{u} \left(\frac{d}{L} \right)^2. \quad (58)$$

7.2 Implicit Dampening

We will also make use of implicit dampening, another fairly simple technique for reducing wave reflection. In this case we add a dampening term to the Navier-Stokes equations and solve the resulting equations using the stable fluids method. This method requires fairly small timesteps for large values of σ and can hence be rather slow. The Navier-Stokes equations for implicit dampening in the boundary region reads

$$\nabla \cdot \mathbf{v} = 0 \quad (59)$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \frac{\mathbf{f}}{\rho} + \nu \nabla^2 \mathbf{v} - \frac{\nabla p}{\rho} - \sigma \mathbf{v}. \quad (60)$$

For these equations we will use the same σ function as for our PML method, that is, Eq. (31).

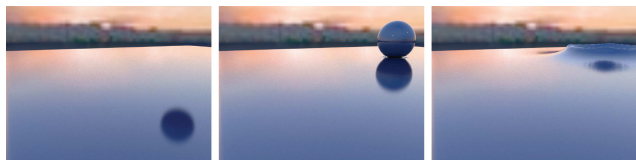


Fig. 5. A rendering depicting the Pebble in the Pond scenario at times 0.1, 0.6, and 0.8 seconds into the simulation.

8. RESULTS

In order to test the behavior and performance of our PML based boundary condition we have run a series of simulations corresponding to graphics scenarios where wave reflection is undesirable. In order to measure the effectiveness of the nonreflective boundaries we look at the total kinetic energy of the simulation

$$\int_{Vol} \frac{1}{2} \mathbf{u} \cdot \mathbf{u} dV \quad (61)$$

as well as the flux

$$\int_{\delta Vol} \mathbf{u} \cdot \mathbf{n}_{\delta} Vol dA \quad (62)$$

through the planes separating the simulation domain from the damping zones near the boundary. We also use visualizations of the offset of the water surface from its equilibrium state in order to better show the wave patterns that develop on the fluid surface. The color map used for the surface offset visualizations are provided to the right of these figures and measures the height of the fluid surface in voxels.

8.1 The Pebble in the Pond

Our first test simulation corresponds to a drop of water hitting the surface of a pond (see Figure 5), generating waves that propagate outwards in rings. This setup is fairly simple and controlled and thus we will use it to investigate the effect of several of the parameters involved in the PML boundary formulation. For all tests in this section a simulation domain of $165 \times 70 \times 165$ voxels was used.

8.1.1 Transfer Function Comparison. When moving from the regular simulation region to the boundary region we need to change σ_{β} from 0 to σ_{max} . This can be done through a simple step function but we suspect that a smoother transition will yield better results. The transfer functions tested can be seen in Figures 2 and 3. In the figures that follow we have described the step width ϵ as a fraction of the boundary width. The damping percentage is calculated through Eq. (34) with $\frac{k}{\omega} = 1$, thus 99% damping equals $\sigma_{max} \approx 77$. As can be seen from the results displayed in Figure 6 the sharpness of our smooth step function does make a difference. For $\epsilon = 0.1$ we see a distinct interference pattern after around 8 seconds. Due to the construction of our simulation this deviation from a pure decaying oscillation must be caused by wave reflection. We can also see that though good results are obtained for $\epsilon > 0.4$ the best result seems to be obtained for $\epsilon = 1$ and thus we conclude that this is the best shape of the “smooth step” transfer function.

Since a sharp transfer function has a larger integration area and thus results in more aggressive damping for a given σ_{max} we need to test the effect of different σ_{max} on the smooth step function with low values of ϵ . This in order to verify that the differences in Figure 6 are due only to the shape of the transfer function and not the increased integration area. As can be seen from the results in Figure 7 even at $\sigma_{max} = 38$, that is, roughly half of what is

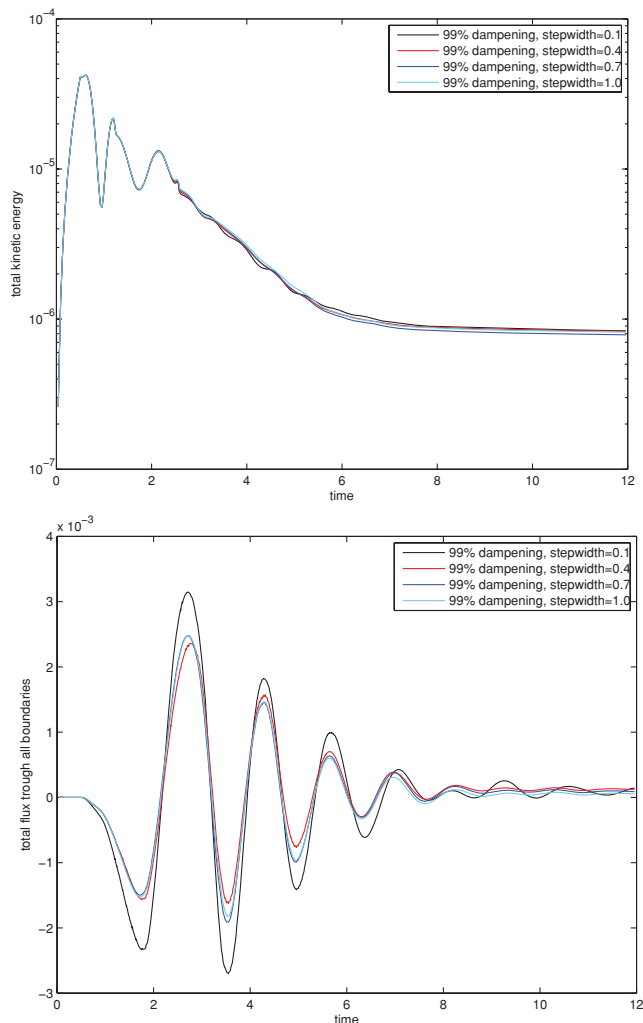


Fig. 6. Energy and flux profiles for varying shape of the “smooth step” boundary transfer function. The step width (i.e., ϵ) is given as a fraction of the boundary width which is 6% of the width of the simulation domain (i.e., 10 voxels) for this simulation.

used in Figure 6 we see a distinct interference pattern. This is consistent with the previous claim that a smooth transfer function is the most appropriate. Note that since the difference in area between the smooth step at $\epsilon = 1$ and $\epsilon = 0.1$ is approximately 2 the amount of damping provided by the smooth step with $\epsilon = 0.1$ and $\sigma_{max} = 38$ should roughly equal that of the smooth step with $\epsilon = 1$ and $\sigma_{max} = 77$.

We now proceed to investigate the effectiveness of the two additional transfer functions presented in Section 5.2, namely Eqs. 30 and 32. In order to make this a fair comparison we have normalized all functions so that the integral of the damping function across the boundary is the same as for our step function. This should ensure that the amount of damping is the same and any differences are only caused by the shape of the function. As can be seen from the results in Figure 8 all transfer function shapes tested do a good job at preventing reflections. The symmetric step function does, however, absorb less kinetic energy while still preventing reflection as well as the other two shapes. Thus we conclude that this shape is the most

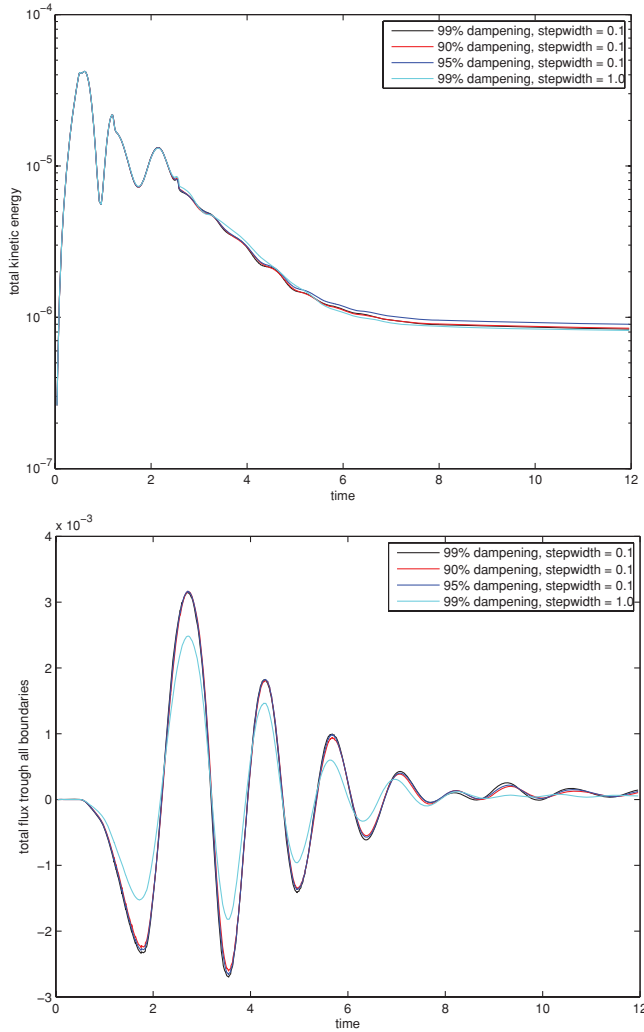


Fig. 7. This graph shows the effect of different σ_{max} on an aggressive transfer function (stepwidth = 0.1). A smooth step (stepwidth = 1.0) is provided as a reference. 90%, 95%, and 99% dampening correspond to $\sigma_{max} = 38, 50,$ and 77 in this test.

effective. However, from Figures 6 and 8 it is evident that as long as the function is fairly smooth it is effective for preventing reflection.

8.1.2 Effect of Different σ_{max} . We now proceed to investigate the effect of different σ_{max} on the smoothest step function (stepwidth = 1.0). As can be seen in Figure 9 we have almost no wave reflections for all our tested values of σ_{max} . However, if we look closely we can see that the result for 99% dampening is better than for both 95% and 99.9%. Thus we conclude that for this simulation an optimum exists close to $\sigma_{max} = 77$, that is, 99% dampening and we will use this setting henceforth.

8.1.3 Effect of Boundary Width. Next we will investigate the importance of the width of the dampening region. The results of this test can be seen in Figure 10. Here we see the expected result that the performance of our boundary is better the wider the boundary region is. We also note that a width of 6%, that is, 10 voxels seems to be sufficient and even 3% is enough to prevent most of the wave reflection.

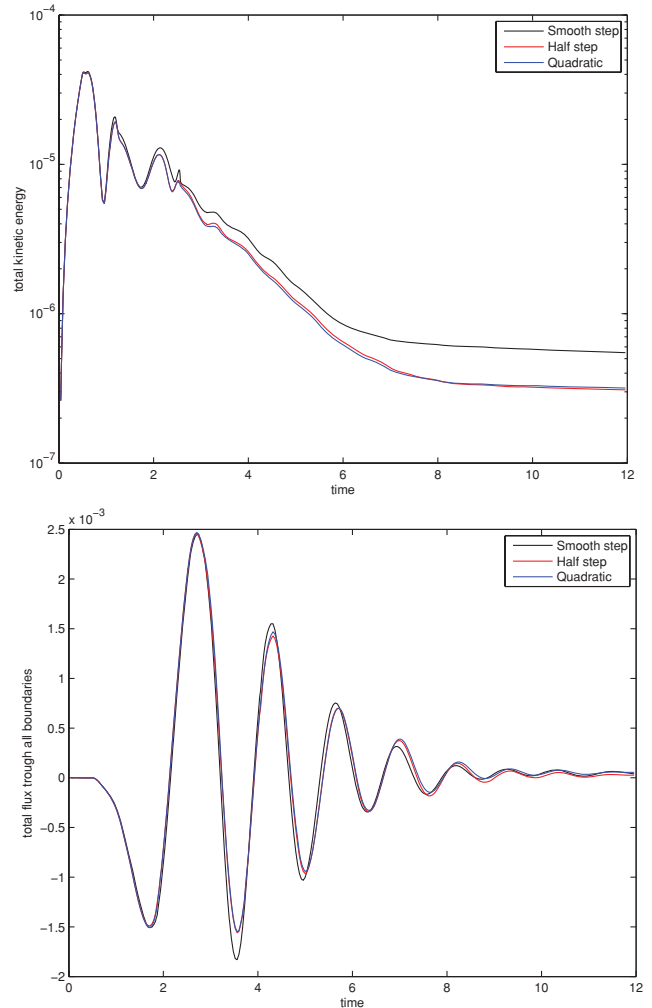


Fig. 8. Comparison between our three different types of σ function: The smooth step (31), the half step (32), and the quadratic function (30). For all tests $\sigma_{max} = 77$ and the boundary width is 6%.

8.1.4 Effect of PML Stabilization. In Section 6.1 we have derived a method to stabilize the time integration for large values of σ . We now investigate the effect of this approach on dampening performance and stability. For this test we will not adapt our timestepping to the requirements of our boundary condition. Instead we will only use the timestepping constraint associated with the undamped simulation, that is, $\sigma = 0$. As can be seen in Figure 11 this does cause an instability for the aggressive case of a narrow boundary and a desired reflection of 1% of the incoming wave amplitude. However, as we enable our time integration stabilization, without any changes to the timestepping condition itself, the simulation is once again stable. We also see that the shape of the flux plots in Figure 11 show an exponentially decaying oscillation for all stable test cases, indicating no detectable reflection. However, the aggressive dampening of the 3% boundary (with proper timestepping) evidently dissipates a lot of energy from the simulation. We suspect that this may be caused by increased numerical dissipation due to the increased number of iterations needed to obtain a stable solution for this case. Another possible explanation is that the large amount

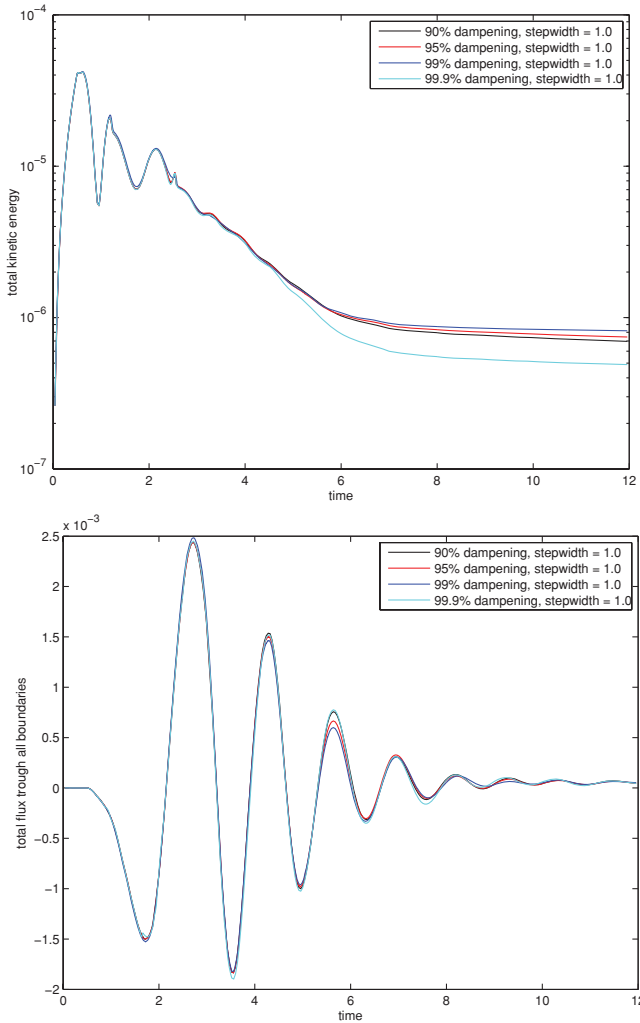


Fig. 9. This graph shows the effect of different σ_{max} on a smooth transfer function (stepwidth = 1.0). 90%, 95%, 99%, and 99.9% damping corresponds to $\sigma_{max} = 38, 50, 77,$ and 115 in this test.

of dampening necessary to obtain 1% reflection for this extreme case is responsible for removing excessive amounts of energy from the rest of the simulation domain. In either case the results shown in Figure 11, indicate that for very aggressive dampening our PML stabilization scheme is in some cases preferable to proper timestepping not only because of low computational cost but also with regard to visual quality of the simulation.

8.1.5 Boundary Effectiveness. Finally we compare the effectiveness of our three reflection preventing methods: explicit dampening, implicit dampening, and our PML-based wave absorbing approach. For these tests we have used a 6% boundary thickness and our boundaries once again use the time integration stabilization described in Section 6.1. As can be seen from the energy and flux graphs in Figure 13 the superior performance of our method is evident under these conditions. This result is further supported when looking at the fluid surface offset visualizations, that is, wave pattern visualizations, provided in Figures 14 to 17. We see that all methods except our PML-based method exhibit clearly visible

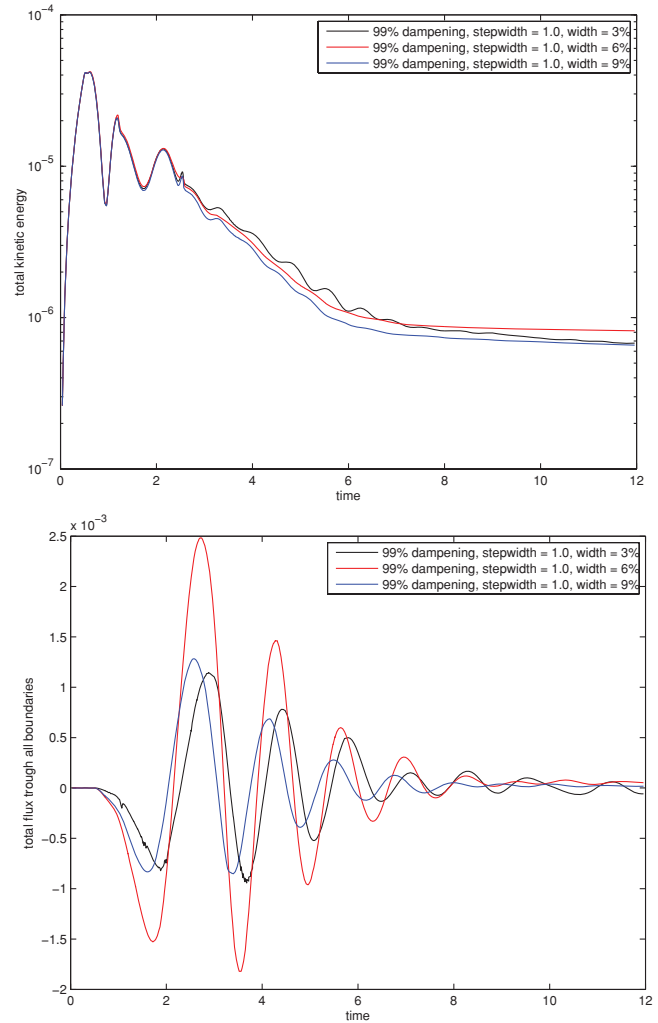


Fig. 10. Effectiveness of dampening for different width of the boundary. Boundary width is given as a percentage of the width of the simulation domain. 3% corresponds to 5 voxels.

interference patterns caused by undesired wave reflections from the physical walls of the simulation domain. When looking at the graphs in Figures 10 and 12 it is also interesting to note that our boundary condition at half the boundary width, that is, 3%, is still more effective than both explicit and implicit dampening in this test.

8.1.6 Speed Comparison. In the boundary domain our method needs up to 12 vector components to represent the velocity field instead of the regular 3. Since the computations performed on each component are very similar to that of the undamped solver an approximate upper bound for the computational cost of our PML condition is 4x the cost when our boundaries are not present. This cost corresponds to the case where every voxel in the simulation domain is a boundary voxel. However, as is evident from the results in, for example, Figure 11, our boundaries are efficient using boundary widths of only 3–6% of the width of the simulation domain. Thus the real-world cost of using our method is significantly lower than the estimated upper bound.

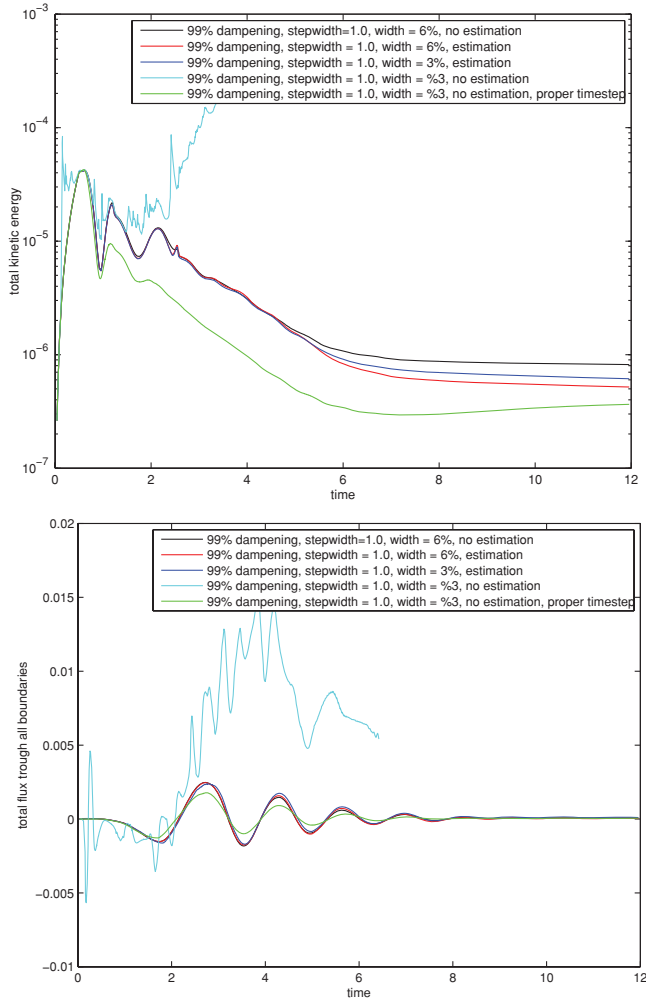


Fig. 11. This graph shows the effects of the PML stabilization. We use a boundary of width 3% and 6% (5 and 10 voxels) and using unaided explicit integration as a reference.

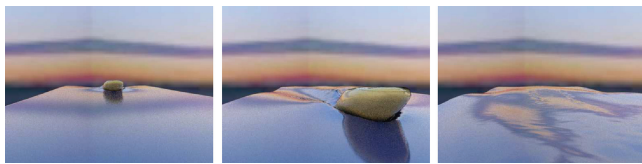


Fig. 12. A rendering depicting the Speedboat scenario at times 1.0, 2.2, and 3.4 seconds into the simulation.

In order to measure the computational cost of our boundaries we have performed the “Pebble in the Pond” simulation using explicit damping, implicit damping, and our PML-based method. The simulation with PML boundaries uses the time integration stabilization presented in Section 6.1. As a reference we have also performed the simulation using an undamped stable fluids solver. Note that in this case we have assumed that all σ_β components can be nonzero. If the directional boundaries described in section 5.4 are used only one σ_β component is nonzero with the exception of

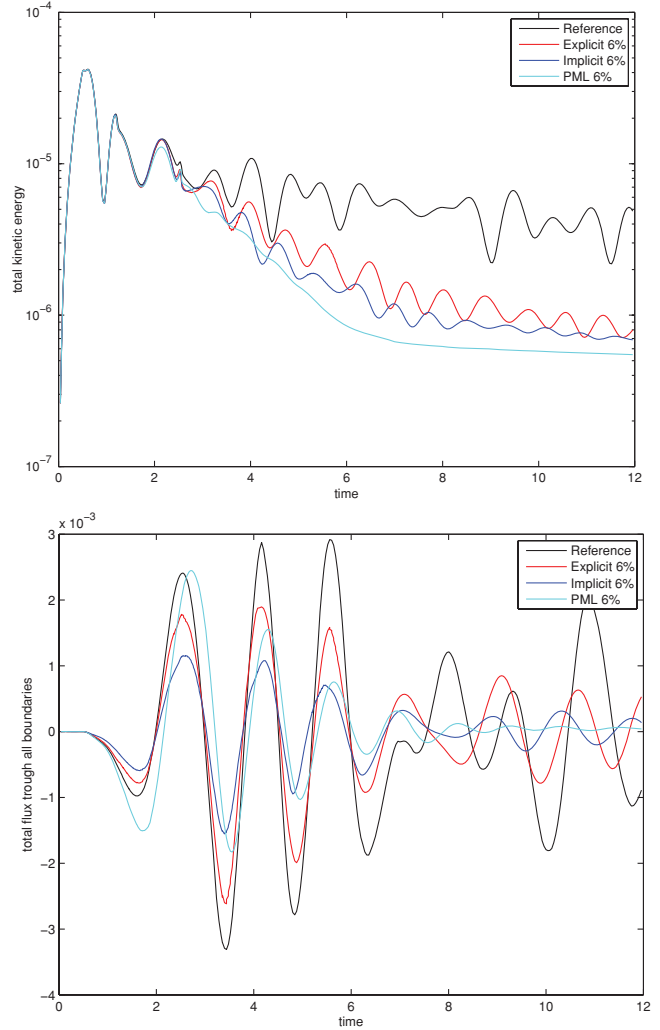


Fig. 13. Comparative results between an undamped reference simulation and the three types of reflection preventing boundary conditions mentioned in this article. The boundary width is 10 voxels and for implicit and PML damping $\sigma = 77$ is used.

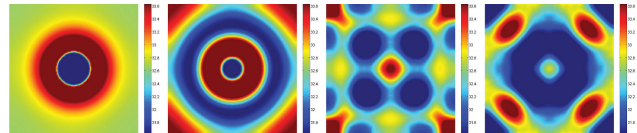


Fig. 14. Surface offset visualization for the “Pebble in the Pond” reference simulation at time 1.1, 2.4, 7.2, and 10 seconds, respectively.

the corners. This allows for further optimizations which can potentially reduce the cost of our boundaries further. As can be seen from the results presented in Table I the computational cost for our boundary in the unoptimized case is approximately 1.75x that of the undamped simulation. The cost of using implicit damping is fairly high due to the additional number of iterations necessary to ensure stability when a damping term is present. The negligible cost of applying explicit damping is also evident; the explicitly

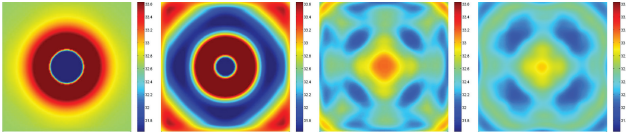


Fig. 15. Surface offset visualization for the “Pebble in the Pond” simulation using explicit dampening. The images shows the simulation at time 1.1, 2.4, 7.2, and 10 seconds, respectively.

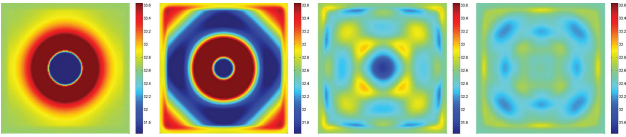


Fig. 16. Surface offset visualization for the “Pebble in the Pond” simulation using implicit dampening. The images shows the simulation at time 1.1, 2.4, 7.2, and 10 seconds, respectively.

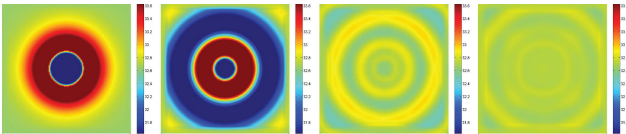


Fig. 17. Surface offset visualization for the “Pebble in the Pond” simulation using our PML-based wave absorbing boundaries. The images show the simulation at time 1.1, 2.4, 7.2, and 10 seconds, respectively.

dampened simulation actually takes less time than the undampened reference simulation. This is caused by the explicit method being as fast as the reference simulation but due to the dampening of waves we can take progressively larger timesteps, resulting in a slightly faster simulation.

The negligible cost of the explicit dampening method calls for further investigation. We could significantly increase the width of the boundary region for the explicit method before we reach the computational cost of our PML approach. In Figure 18 we have shown the results obtained when this is done. The average time per frame for the explicit simulation is now 184 seconds and the boundary width is 50 voxels. The size of the undampened central region for both simulations are equal. As can be seen from Figure 18 the explicit method works much better with these wide boundaries, however, in spite of boundary regions covering nearly 60% of the fluid surface the explicit method is still unable to obtain the reflection preventing qualities of our boundaries. This is even more evident if one studies the surface offset visualizations in Figure 19.

8.2 The Pebble in the Pond - Viscous

This test is designed to show the performance of our PML-based boundary for viscous fluids. The setup is the same as for Section 8.1 before: a drop of water falling into a pond. In this case we imagine the fluid being cold motor oil ($\nu = 8.1$ Stokes). As can be seen from Figure 20 our method maintains its effectiveness also for this scenario. Thus we conclude that our method can be used as a reflection-preventing technique for viscous fluids in spite of our approximation in Section 6.2.

Table I. Time Required for Completing One Frame of the “Pebble in the Pond” Simulation Using Different Dampening Methods

	Reference	Explicit	Implicit	PML
Time/frame (s)	90	86	137	156
Relative time	1.00	0.96	1.54	1.74

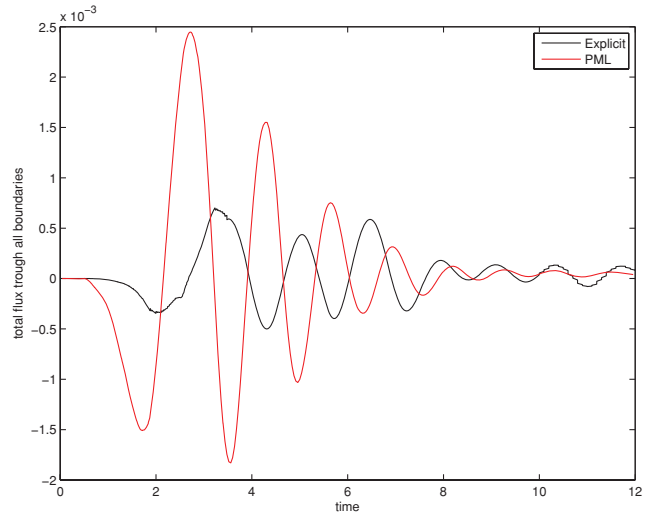
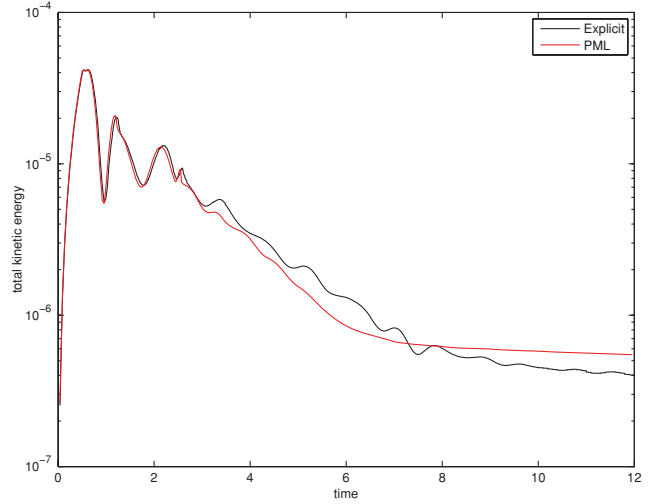


Fig. 18. Comparison between explicit and PML boundaries at roughly the same computational cost. The boundary widths are 50 voxels for the explicit simulation and 10 voxels for the PML dampened simulation. The average time per frame is 184 seconds for the explicit method and 156 seconds for the PML simulation. Only the boundary widths are different; the size of the undampened region is equal in both simulations.

8.3 Speedboat

The speedboat simulation (see Figure 12) is intended as an example of a nontrivial use of our PML boundary condition. Here we have used the level set boundary approach in Section 5.4 in order to create a smooth and nontrivial boundary shape. For this simulation we have created a bottle-like boundary that allows the boat to enter and leave the simulation domain through a narrow opening. An

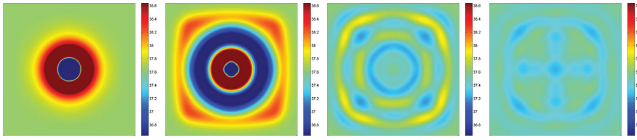


Fig. 19. Surface offset visualization for the “Pebble in the Pond” simulation using wide explicit boundaries. The total area of the boundary region corresponds to roughly 60% if the entire simulation domain. The images shows the simulation at time 1.1, 2.4, 7.2, and 10 seconds, respectively.

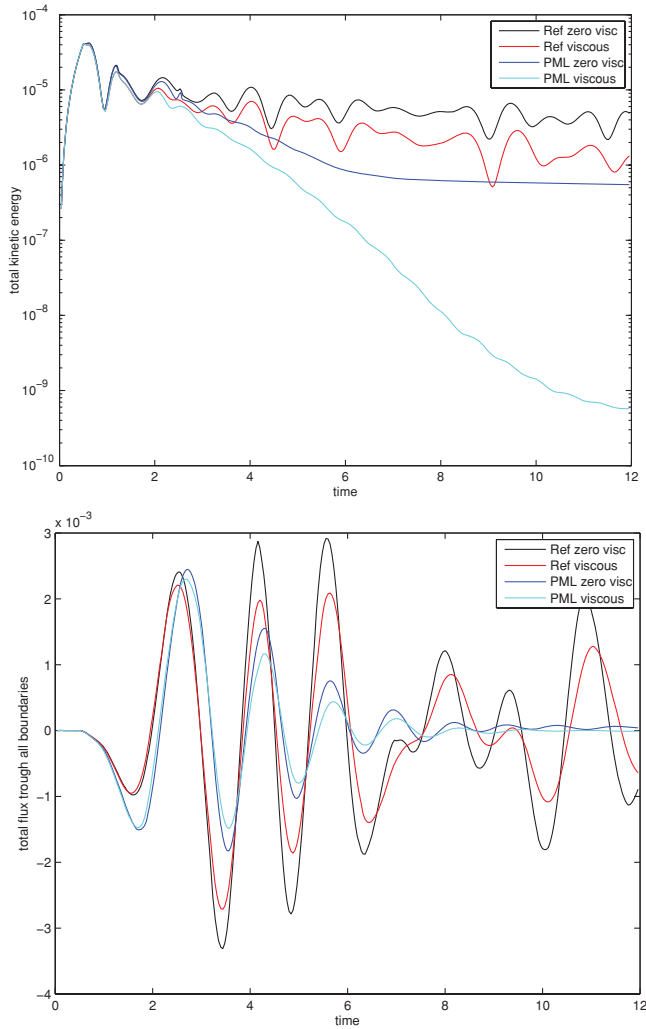


Fig. 20. This graph shows the behavior of our wave absorbing boundary for a viscous fluid with $\nu = 8.1$ Stokes. As a reference the same simulation with ≈ 0 viscosity is also provided.

illustration of this boundary shape is provided in Figure 21. The kinetic energy diagram for this simulation can be seen in Figure 22 and a visualization of the fluid surface offset at different points in time is presented in Figure 23. As can be seen in Figure 23 the reference simulation exhibits the telltale undesired interference pattern long after the boat has passed. This is not the case when our boundary condition is employed, thus showing its effectiveness

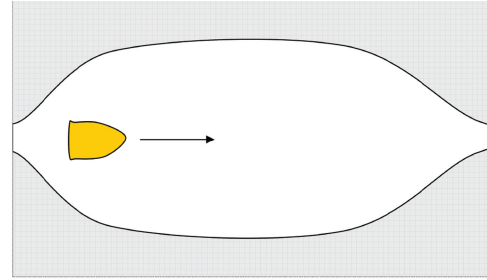


Fig. 21. Illustration of the nonreflective boundary shape used for the “Speedboat” simulation.

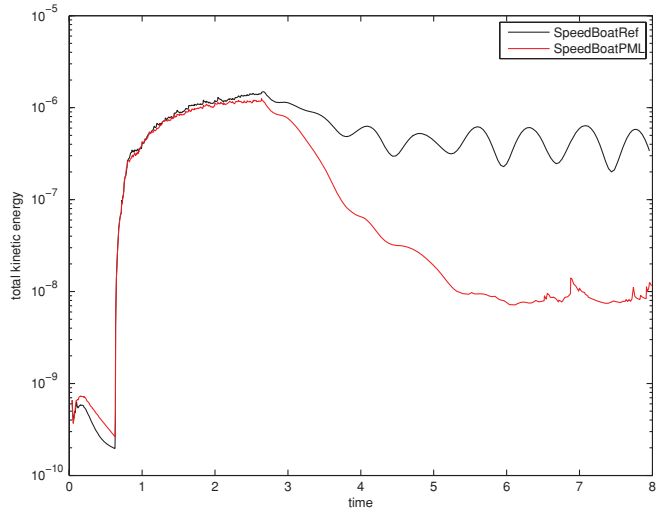


Fig. 22. Kinetic energy graph for the “Speedboat” simulation.

in absorbing any waves leaving the simulation domain also for this scenario.

8.4 Crazy Boat

This simulation (see Figure 24) is intended to show that complex fluid interactions are possible when using our PML-based boundaries. Here a boat moves around in a lake making tight turns in a flower-like pattern. As can be seen from the kinetic energy graph in Figure 25 and even more clearly from the surface offset visualizations in Figure 26 our boundaries once again do a good job at removing reflections. Thus we conclude that our method works well also for nontrivial, visual effects/oriented fluid simulations.

9. CONCLUSIONS

We have presented a novel PML-based wave absorbing boundary condition for the free surface incompressible Euler and Navier-Stokes equations in the presence of external forces. We have also presented a novel algorithm for solving the resulting equations in the boundary region based on the stable fluids approach. We have shown that our boundary condition can efficiently absorb incoming waves causing very little reflection in the process and that this ability also extends to complex boundary shapes and nontrivial fluid-solid interactions. Furthermore we have shown that our method provides

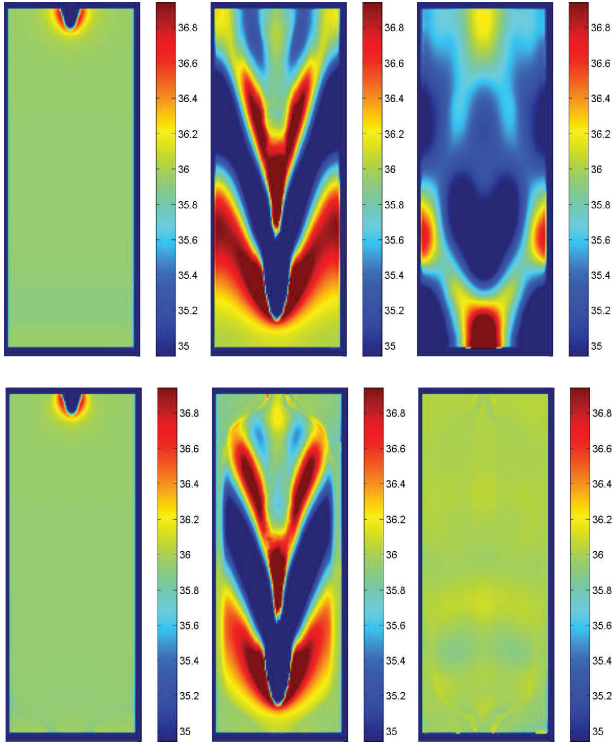


Fig. 23. Water surface offset visualization of the “Speedboat” simulation. The first three images (top row) show the reference simulation after 0.8, 2.5, and 7.8 seconds, respectively. The three images on the second row show the same simulation with a PML wave absorbing layer present.



Fig. 24. A rendering depicting the Crazy Boat scenario at times 0.4, 2.1, and 5.4 seconds into the simulation.

significantly better results than both explicit and implicit dampening, sometimes requiring boundary widths of only a couple of voxels in order to be effective (see Figure 10). In order to solve our boundary equations in a fast and stable manner we have had to make a number of approximations. However, our results show that in spite of this our method performs very well as a nonreflecting boundary. Our results also show that the transition from the regular simulation domain to the boundary region needs to be smooth. However, the exact shape of the transfer function has proven to be of lesser importance; all the functions tested performed well with the exception of a step function. We have also shown that our specialized time integration scheme for the dampened self-advection term results in a stability condition that is not dependant on the amount of dampening applied, thus making our method fast even for very aggressive boundaries. Finally we have shown that the computational cost associated with our boundaries is fairly low, typically 1.75 times that of the undampened reference simulation. This cost, though not insignificant, has still proven smaller than the cost of

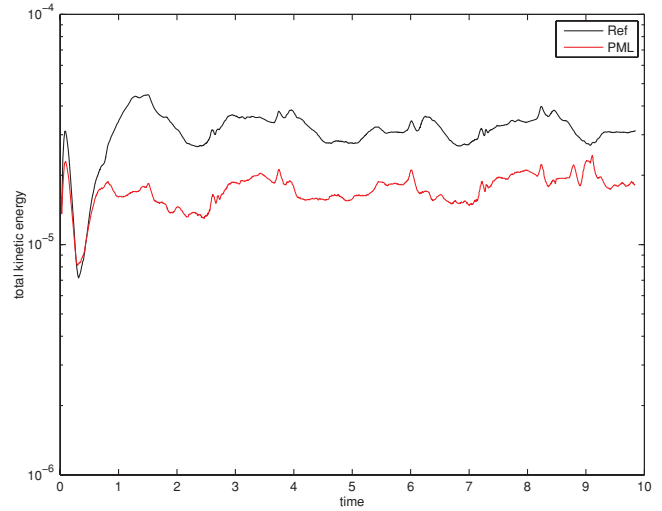


Fig. 25. Kinetic energy graph from the “Crazy Boat” simulation.

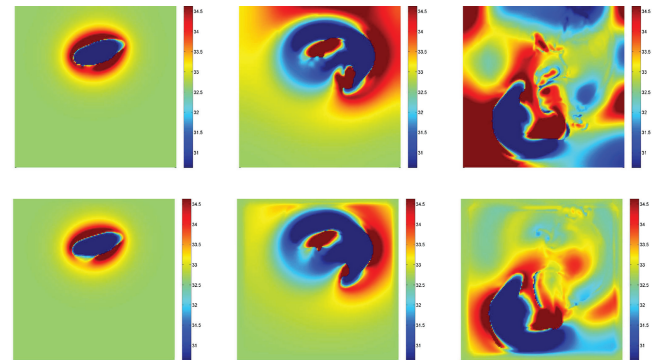


Fig. 26. Surface offset visualizations from the “Crazy Boat” simulation at time 0.1, 0.8, and 2.8 seconds, respectively. The top three images show the reference simulation and the bottom three show the same simulation using our PML-based boundaries.

achieving equivalent results using explicit and implicit dampening as is evident in Section 8.1.6. It should be noted that the dampening regions surrounding the simulation domain behave in an unphysical manner, which can be surmised from, for example Figures 16, 17 and 26. This is to be expected since these regions represent a wave absorbing medium and not actual fluid. From a visual effects perspective it is thus prudent that these regions are removed from the final rendering of the simulation using for example Constructive Solid Geometry (CSG) operations.

We note that there are several areas in which our method can be improved. First of all the algorithm described in Section 6, though based on the stable fluids approach, is not unconditionally stable. Due to our explicit integration of the self-advection term a CFL stability condition must be considered. An unconditionally stable method for solving this term in the boundary region would potentially allow for faster stable integration and thus even faster computations. We have also observed a tendency for the $q_{0,\beta}$ variables not to return completely to zero when the fluid returns to a stationary state, thus resulting in a small net flux into or out of the boundary

region. For all our tests this effect has been too small to be seen by the naked eye, however, it is an indication that our integration scheme for these variables can be further improved. Another potential area for improvement is the integration of the viscosity term. Though our tests in Section 8.2 indicate that our boundaries work well also for viscous fluids an unconditionally stable method for solving Eq. (50) without approximations could potentially provide even better results. It is also important to note that we have assumed that all forces in the boundary region are conservative. In practice this works well since the only force we need to incorporate everywhere by necessity is the force of gravity. This is further helped by the fact that our boundaries do not need to be very wide to be effective. If more general force interactions are desired we expect that arbitrary external forces can be integrated into our method by splitting Eq. (10) as is. However, this will add additional complexity to the derivation in Section 4 and the algorithm in Section 6. Thus we consider arbitrary external forces in the boundary region to be beyond the scope of this article. In this article we also do not consider the case of a mean-flow in the boundary domain. However, we believe that the method described in Hu et al. [2008] for this scenario can be applied to our work as well. Finally we wish to note that though our boundary condition is PML based the resulting equations are not formally perfectly matched to the Navier-Stokes equations, primarily due to the splitting approach used to transform Eq. (13) back to its spatial representation. Thus to be accurate our boundary condition should be considered “well matched” to the Navier-Stokes equations.

The aforesaid limitations allow for further improvement of our method. However, we are still able to efficiently realize boundaries that are practically reflection free. Though our boundaries are rather involved, much of the complexity can be hidden from the end user. Only two parameters are necessary during deployment: desired reflection (in percent) and desired boundary width. As a result we believe that our method can be deployed in an intuitive and user-friendly way. We also strongly believe that this article and the work it presents will provide the visual effects community with a new and highly useful tool in the struggle for better and ever more realistic visual effects.

ACKNOWLEDGMENTS

The authors would like to thank the other members of the Graphics Group at Linköping University for sharing various software libraries - in particular M. Nielsen for his optimization of the DT-Grid [Nielsen and Museth 2006]. A. Söderström would also like to thank the Department of Management and Engineering at Linköping University for contributing office space and resources during this project. K. Museth would like to thank Digital Domain and Dream-Works Animation for supporting the continued PhD supervision of A. Söderström. Finally the authors would like to thank the reviewers for their insightful comments and suggestions for improvement of this work.

REFERENCES

- ABARBANEL, S. AND GOTTLIEB, D. 1997. A mathematical analysis of the pml method. *J. Comput. Phys.* 134, 2, 357–363.
- ADALSTEINSSON, D. AND SETHIAN, J. A. 1995. A fast level set method for propagating interfaces. *J. Comput. Phys.* 118, 2, 269–277.
- BALDAUF, M. 2008. Stability analysis for linear discretisations of the advection equation with runge-kutta time integration. *J. Comput. Phys.* 227, 13, 6638–6659.
- BCACHE, E., FAUQUEUX, S., AND JOLY, P. 2003. Stability of perfectly matched layers, group velocities and anisotropic waves. *J. Comput. Phys.* 188, 2, 399–433.
- BCACHE, E., DHIA, A.-S. B.-B., AND LEGENDRE, G. 2004. Perfectly matched layers for the convected helmholtz equation. *SIAM J. Numer. Anal.* 42, 1, 409–433.
- BERENGER, J.-P. 1994. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.* 114, 2, 185–200.
- BCACHE, E., FAUQUEUX, S., AND JOLY, P. 2003. Stability of perfectly matched layers, group velocities and anisotropic waves. *J. Comput. Phys.* 188, 2, 399–433.
- CHEW, W. C. AND WEEDON, W. H. 1994. A 3-d perfectly matched medium from modified maxwell’s equations with stretched coordinates. *Microwave Opt. Tech. Lett.* 7, 599–604.
- CHORIN, A. 1968. Numerical solution of navier-stokes equations. *Math. Comp.* 22, 745–762.
- ENRIGHT, D., FEDKIW, R., FERZIGER, J., AND MITCHELL, I. 2002. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* 183, 1, 83–116.
- FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of ACM SIGGRAPH Conference*. 15–22.
- FOSTER, N. AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of ACM SIGGRAPH Conference*. ACM Press, 23–30.
- GOTTLIEB, S. AND SHU, C.-W. 1998. Total variation diminishing runge-kutta schemes. *Math. Comput.* 67, 221, 73–85.
- HAGSTROM, T., GOODRICH, J., NAZAROV, I., AND DODSON, C. 2005. High-order methods and boundary conditions for simulating subsonic flows. In *Proceedings of the 11th AIAA/CEAS Aeroacoustics Conference*. AIAA–2005–2869.
- HAGSTROM, T. AND NAZAROV, I. 2002. Absorbing layers and radiation boundary conditions for jet flow simulations. In *Proceedings of the 8th AIAA/CEAS Aeroacoustics Conference and Exhibit*. 2002–2606.
- HAGSTROM, T. AND NAZAROV, I. 2003. Perfectly matched layers and radiation boundary conditions for shear flow calculations. In *Proceedings of the 9th AIAA/CEAS Aeroacoustics Conference and Exhibit*. 2003–3298.
- HE, X. AND LUO, L.-S. 1997. Lattice boltzmann model for the incompressible navierstokes equation. *J. Statist. Phys.* 88, 3, 927–944.
- HEIN, S., HOHAGE, T., KOCH, W., AND SCHBERL, J. 2007. Acoustic resonances in a high-lift configuration. *J. Fluid Mech.* 582, -1, 179–202.
- HU, F. 2006. On the construction of pml absorbing boundary condition for the non-linear euler equations. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*. AIAA–2006–798.
- HU, F. Q. 1995. On absorbing boundary conditions for linearized euler equations by a perfectly matched layer. *J. Comput. Phys.* 129, 201–219.
- HU, F. Q. 1996. On absorbing boundary conditions for linearized euler equations by a perfectly matched layer. *J. Comput. Phys.* 129, 1, 201–219.
- HU, F. Q. 2001a. A stable, perfectly matched layer for linearized euler equations in unsplit physical variables. *J. Comput. Phys.* 173, 2, 455–480.
- HU, F. Q. 2001b. A stable, perfectly matched layer for linearized euler equations in unsplit physical variables. *J. Comput. Phys.* 173, 2, 455 – 480.
- HU, F. Q. 2008. Development of pml absorbing boundary conditions for computational aeroacoustics: A progress review. *Comput. Fluids* 37, 4, 336–348.
- HU, F. Q., HUSSAINI, M. Y., AND MANTHEY, J. L. 1996. Low-dissipation and low-dispersion runge-kutta schemes for computational acoustics. *J. Comput. Phys.* 124, 1, 177–191.
- HU, F. Q., LI, X. D., AND LIN, D. K. 2008. Absorbing boundary conditions for nonlinear euler and navier-stokes equations based on the perfectly matched layer technique. *J. Comput. Phys.* 227, 9, 4398–4424.

- JOHNSON, S. G. 2007. Notes on perfectly matched layers. <http://www-math.mit.edu/~stevenj/18.369/pml.pdf>
- LI, X. D. AND GAO, J. H. 2005. Numerical simulation of the generation mechanism of axisymmetric supersonic jet screech tones. *Phys. Fluids* 17, 8, 085105.
- LI, X. D. AND GAO, J. H. 2008. Numerical simulation of the three-dimensional screech phenomenon from a circular jet. *Phys. Fluids* 20, 3, 035101.
- LIU, X., OSHER, S., AND CHAN, T. 1994. Weighted essentially nonoscillatory schemes. *J. Comput. Phys.* 115, 200–212.
- MONAGHAN, J. 1988. An introduction to sph. *Comput. Phys. Comm.* 48, 89–96.
- MUSETH, K. AND LEFORESTIER, C. 1996. On the direct complex scaling of matrix elements expressed in a discrete variable representation: Application to molecular resonances. *J. Chem. Phys.* 104, 18, 7008–7014.
- NEUHASUER, D. AND BAER, M. 1989. The time-dependent schrödinger equation: Application of absorbing boundary conditions. *J. Chem. Phys.* 90, 8, 4351–4355.
- NIELSEN, M. B. AND MUSETH, K. 2006. Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets. *J. Sci. Comput.* 26, 3, 261–299.
- NIELSEN, M. B., NILSSON, O., SÖDERSTRÖM, A., AND MUSETH, K. 2007. Out-of-core and compressed level set methods. *ACM Trans. Graph.* 26, 4, 16.
- OSHER, S. AND FEDKIW, R. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, Berlin.
- ÖZYÖRÜK, Y. 2009. Numerical prediction of aft radiation of turbofan tones through exhaust jets. *J. Sound Vibrat.* 325, 1-2, 122–144.
- REINHARDT, W. P. 1982. Complex coordinates in the theory of atomic and molecular structure and dynamics. *Ann. Rev. Phys. Chem.* 33, 223–255.
- RICHARDS, S. K., ZHANG, X., CHEN, X. X., AND NELSON, P. A. 2004. The evaluation of non-reflecting boundary conditions for duct acoustic computation. *J. Sound Vibrat.* 270, 3, 539–557.
- SHU, C. AND OSHER, S. 1988. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.* 77, 439–471.
- SÖDERSTRÖM, A. AND MUSETH, K. 2009. Nonreflective boundary conditions for incompressible free surface fluids. In *Proceedings of the SIGGRAPH'09 Talks*. ACM, New York 1–1.
- STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH Conference*. 121–128.
- TAM, C. K. AND WEBB, J. C. 1993. Dispersion-relation-preserving finite difference schemes for computational acoustics. *J. Comput. Phys.* 107, 2, 262–281.
- TURKEL, E. AND YEFET, A. 1998. Absorbing pml boundary layers for wave-like equations. *Appl. Numer. Math.* 27, 4, 533–557.
- ZHAO, L. AND CANGELLARIS, A. 1996. Gt-pml: generalized theory of perfectly matched layers and its application to the reflectionless truncation of finite-difference time-domain grids. *IEEE Trans. Microw. Theory Tech.* 44, 12, 2555–2563.
- ZHU, Y. AND BRIDSON, R. 2005. Animating sand as a fluid. In *Proceedings of the ACM SIGGRAPH Papers*. ACM, New York. 965–972.

Received August 2009; revised April 2010; accepted August 2010