INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:1451–1468
Published online 29 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.813

Discontinuous boundary implementation for the shallow water equations

Shintaro Bunya[1,*,†], Joannes J. Westerink[2] and Shinobu Yoshimura[1]

[1] *Institute of Environmental Studies, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-0033, Japan*
[2] *Department of Civil Engineering and Geological Sciences, University of Notre Dame, Notre Dame, IN 46556, U.S.A.*

SUMMARY

Quasi-bubble finite element approximations to the shallow water equations are investigated focusing on implementations of the surface elevation boundary condition. We first demonstrate by numerical results that the conventional implementation of the boundary condition degrades the accuracy of the velocity solution. It is also shown that the degraded velocity leads to a critical instability if the advection term is present in the momentum equation. Then we propose an alternative implementation for the boundary condition. We refer to this alternative implementation as a discontinuous boundary (DB) implementation because it introduces at each boundary node two independent mass–flux values that result in a discontinuity at the boundary. Numerical results show that the proposed DB implementation is consistent, stabilizes the quasi-bubble scheme, and leads to second-order accuracy at the surface elevation specified boundary. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: quasi bubble-function; shallow water equations; finite elements; boundary condition

1. INTRODUCTION

The 2D shallow water equations (SWE) are obtained from the Navier–Stokes equations by assuming a hydrostatic pressure distribution and a uniform velocity profile in the vertical direction. The SWE are a very efficient formulation to describe a class of fluid dynamics phenomena to which the shallow-water long-wave assumption is applicable. That class of phenomena includes storm surges, tidal fluctuations, tsunami waves, and forces acting on offshore structures. As seen in the Navier–Stokes equations, primitive approaches such as the standard Galerkin finite element formulation are not accurate for the SWE because 2δ-scale spurious oscillations arise. While many finite element formulations have been investigated in order to eliminate spurious oscillation [1, 2], the quasi bubble-function approach, which was

[*] Correspondence to: S. Bunya, Institute of Environmental Studies, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-0033, Japan.
[†] E-mail: bunya@save.k.u-tokyo.ac.jp

Received 24 May 2004
Revised 14 October 2004
Accepted 15 October 2004

introduced by Mewis and Holz [3] to the SWE and extensively used in the Telemac model [4, 5] developed by Electricité de France (EDF), has many advantageous properties. Mewis and Holz showed numerical results and demonstrated that the quasi-bubble scheme eliminates spurious oscillations without requiring any non-physical damping. In their calculations, the advection term was not included. Atkinson *et al.* [6] compared the quasi-bubble scheme with the generalized wave continuity equation (GWCE) scheme which is another extensively exercised SWE solution with very low numerical damping. They demonstrated by dispersion error analyses with linearized harmonic formulations that the quasi-bubble scheme has a very accurate monotonic dispersion relationship, which theoretically indicates the potential of the quasi-bubble scheme to eliminate spurious oscillations. We implemented the quasi-bubble scheme for the SWE with the advection term expecting smooth solutions. However, our numerical results showed non-physical oscillations. It was observed that the quasi-bubble scheme leads to a significant instability when surface elevation is specified as an essential boundary condition. We supposed that the advection term in the momentum equation might cause the spurious oscillations at the boundary. In fact, when the advection term was dropped from the momentum equation, we did not see any undesirable oscillations, and the calculation was very stable.

Kolar *et al.* [7] derived a unique finite element formulation using generalized functions, and suggested an alternative implementation of the boundary conditions for the SWE. Their formulation demonstrates that boundary conditions should be implemented by treating normal fluxes as natural conditions with the flux interpreted as external to the computational domain. They applied the alternative implementation to the GWCE in order to improve the mass-conserving properties. The alternative implementation allows us to have a pair of velocity values at each open boundary node; one of them is a velocity defined for the internal domain, and the other is a velocity defined for the external domain.

Extending their idea to the momentum equation, we derived another alternative implementation of the boundary conditions, which utilizes the external velocity to augment the order of accuracy of the momentum equation. Since two velocity values defined from the interior or exterior domain can have different values, we refer to this implementation as a discontinuous boundary (DB) implementation. Adopting the DB implementation and a modification described later, the quasi-bubble scheme performed very stably and provided very smooth solutions.

In this article, we first demonstrate how the conventional implementation causes an instability in conjunction with the advection term and surface-elevation essential boundary conditions. Then the details of the DB implementation are introduced, followed by numerical experiments that support the consistency, stability, and high-order accuracy of the proposed formulation.

2. GOVERNING EQUATIONS AND NUMERICAL SCHEMES

2.1. Shallow water equations

2D SWE are written as

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (H\mathbf{u}) = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \tau \mathbf{u} + f_{\mathrm{c}} \mathbf{k} \times \mathbf{u} + g \nabla \zeta - \frac{v}{H} \Delta (H\mathbf{u}) = \frac{1}{H} \mathbf{F} \tag{2}$$

where ζ is the surface elevation, $H = h + \zeta$ is the total water depth, h is the bathymetric depth, $\mathbf{u} = (u, v)$ is the horizontal velocity field, f_{c} is the Coriolis parameter, $\mathbf{k}$ is the local vertical vector, g is the gravitational acceleration, τ is the bottom friction coefficient, and v is the depth averaged turbulent viscosity. Surface wind stress, variable atmospheric pressure, and tidal potential are expressed through the body force $\mathbf{F}$. Equation (1) is the continuity equation in primitive form, and Equation (2) is the momentum equation in non-conservative form. In the following discussion, we use the 1D SWE deduced from Equations (1) and (2). In 1D form, the Coriolis term is dropped. For further simplification, the diffusion term and the body force term are excluded from the momentum equation. Thus we deal with the following 1D form of the shallow water equations:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial H u}{\partial x} = 0 \tag{3}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial \zeta}{\partial x} - \tau u \tag{4}$$

On a domain Ω let $(\cdot,\cdot)_\Omega$ denote the $L^2(\Omega)$ inner product. To distinguish boundary integrations we will use the notation $\langle\cdot,\cdot\rangle_{\partial\Omega}$. The weighted residual statements of the governing equations (3) and (4) are

$$\left(\frac{\partial \zeta}{\partial t} + \frac{\partial Q}{\partial x}, \phi_I\right)_\Omega = 0 \tag{5}$$

$$\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial \zeta}{\partial x} + \tau u, \psi_J\right)_\Omega = 0 \tag{6}$$

where ϕ_I and ψ_J are the globally defined bases of test functions, and $Q = Hu$. The upper-case subscripts, I and J, represent globally assigned node numbers, while lower-case subscripts, i, j, l, m, and n, which will be used later, represent local elemental node numbers.

Now we adopt the quasi-bubble scheme in order to obtain a finite element formulation. In the quasi-bubble scheme the standard linear bases are used as both interpolation functions for the surface elevation and the test functions for the continuity equation, while quasi bubble-function bases are used as both interpolation functions for velocity and the test functions for the momentum equation. The 1D standard linear bases ϕ_i and the quasi bubble-function bases ψ_i are depicted in Figure 1 and defined in local co-ordinates s as

$$\phi_1 = \tfrac{1}{2}(1 - s) \tag{7}$$

$$\phi_2 = \tfrac{1}{2}(1 + s) \tag{8}$$

$$\psi_1 = \begin{cases} -s & \text{for } s \in -1 < s < 0 \\ 0 & \text{for } s \in 0 < s < 1 \end{cases} \tag{9}$$

$$\psi_2 = \begin{cases} 1 + s & \text{for } s \in -1 < s < 0 \\ 1 - s & \text{for } s \in 0 < s < 1 \end{cases} \tag{10}$$

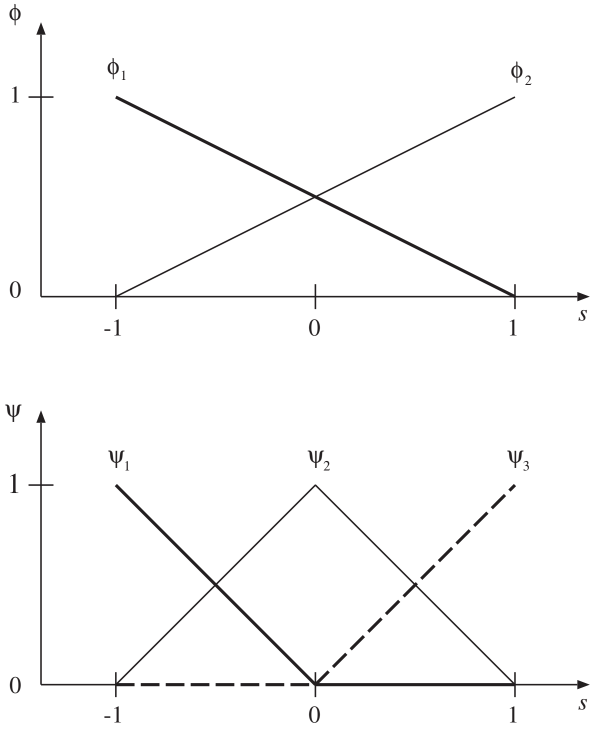

Figure 1. The 1D basis functions.

$$\psi_3 = \begin{cases} 0 & \text{for } s \in -1 < s < 0 \\ s & \text{for } s \in 0 < s < 1 \end{cases} \tag{11}$$

Functions are then piecewise interpolated by those basis functions and nodal values as

$$v = v_i^{(e)} \phi_i^{(e)} \tag{12}$$

$$w = w_j^{(e)} \psi_j^{(e)} \tag{13}$$

$$\zeta = \zeta_i^{(e)} \phi_i^{(e)} \tag{14}$$

$$u = u_j^{(e)} \psi_j^{(e)} \tag{15}$$

where the summation rule over $i \in \{1, 2\}$ and $j \in \{1, 2, 3\}$ is implied. In the above equations, v is the test function for surface elevation, and w is the test function for velocity. $v_i^{(e)}$, $w_i^{(e)}$, $\zeta_i^{(e)}$, and $u_i^{(e)}$ are nodal values of each function. $\phi_i^{(e)}$ and $\psi_i^{(e)}$ are elementwise-defined basis functions in the global co-ordinate x. Superscripts (e) indicate that the functions and nodal values are defined in or belong to element e. The summation rule over $e \in \{1, 2, \ldots, N-1\}$ is also implied. N represents the number of standard nodes.

Substituting the element-based discretizations into Equations (5) and (6) yields

$$\left(\frac{\partial \zeta_l^{(e)}}{\partial t} \phi_l^{(e)} + Q_m^{(e)} \frac{\partial \psi_m^{(e)}}{\partial x}, \phi_i^{(e)} \right)_{\Omega^{(e)}} = 0 \tag{16}$$

$$\left(\frac{\partial u_m^{(e)}}{\partial t} \psi_m^{(e)} + u_m^{(e)} u_n^{(e)} \psi_m^{(e)} \frac{\partial \psi_n^{(e)}}{\partial x} + g \zeta_l^{(e)} \frac{\partial \phi_l^{(e)}}{\partial x} + \tau u_m^{(e)} \psi_m^{(e)}, \psi_j^{(e)} \right)_{\Omega^{(e)}} = 0 \tag{17}$$

Then we have element-matrix formulations of the approximated shallow water equations:

$$\mathbf{ME}_{il}^{(e)} \frac{\partial \zeta_l^{(e)}}{\partial t} + \mathbf{B}_{im}^{(e)} Q_m^{(e)} = 0 \tag{18}$$

$$\mathbf{MU}_{jm}^{(e)} \frac{\partial u_m^{(e)}}{\partial t} + \mathbf{C}_{jm}^{(e)}(\mathbf{u}^{(e)}) u_m^{(e)} + g \mathbf{D}_{jl}^{(e)} \zeta_l^{(e)} + \tau \mathbf{MU}_{jm}^{(e)} u_m^{(e)} = 0 \tag{19}$$

where

$$\mathbf{ME}_{il}^{(e)} = (\phi_i^{(e)}, \phi_l^{(e)})_{\Omega^{(e)}} = \frac{L^{(e)}}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{20}$$

$$\mathbf{B}_{im}^{(e)} = \left(\phi_i^{(e)}, \frac{\partial \psi_m^{(e)}}{\partial x} \right)_{\Omega^{(e)}} = \frac{1}{4} \begin{bmatrix} -3 & 2 & 1 \\ -1 & -2 & 3 \end{bmatrix} \tag{21}$$

$$\mathbf{MU}_{jm}^{(e)} = (\psi_j^{(e)}, \psi_m^{(e)})_{\Omega^{(e)}} = \frac{L^{(e)}}{12} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix} \tag{22}$$

$$\begin{aligned} \mathbf{C}_{jm}^{(e)} &= (\psi_j^{(e)}, \psi_m^{(e)})_{\Omega^{(e)}} \frac{\mathrm{d}\psi_n^{(e)}}{\mathrm{d}x} u_n^{(e)} \\ &= \frac{1}{6} \left\{ \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} [-1 \;\; 1 \;\; 0] + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} [0 \;\; -1 \;\; 1] \right\} \begin{bmatrix} u_1^{(e)} \\ u_2^{(e)} \\ u_3^{(e)} \end{bmatrix} \end{aligned} \tag{23}$$

$$\mathbf{D}_{jl}^{(e)} = \left(\psi_j^{(e)}, \frac{\partial \phi_l^{(e)}}{\partial x} \right)_{\Omega^{(e)}} = \frac{1}{4} \begin{bmatrix} -1 & 1 \\ -2 & 2 \\ -1 & 1 \end{bmatrix} \tag{24}$$

As seen in Figure 1, the test functions at bubble nodes have a non-zero value within only one element that contains the centre node. In other words, the velocity at the centre node can be determined by the momentum equations associated with one element instead of solving globally assembled momentum equations. Thus the allocation of the bubble-function nodes permits a static condensation procedure [8]. Thanks to this procedure one can reduce the dimension of the simultaneous equations, and avoid a significant increase of computing time, which might occur due to the additional degrees of freedom of the bubble-function nodes. Mewis and Holz [3] reported that the quasi-bubble scheme augmented computation time up to approximately 20% as compared to standard linear elements, and it could be reduced further by efficient programming.

For the temporal discretization, the predictor multi-corrector method is employed using the first-order forward-difference method as a predictor and the Crank–Nicolson method as a corrector.

The test problem that will be presented later includes two kinds of boundary conditions; one of them is an open boundary condition, the other is a land boundary condition. The former, the open boundary condition, is given by specifying surface elevation at the boundary, and a velocity value is not specified. The latter, the land boundary condition, is given by velocity equal to zero in the normal direction to the boundary. In the following sections we specifically discuss implementations of the open boundary condition.

2.2. The conventional implementation of open boundary condition

The main focus of this paper is on the implementation of the open boundary condition. In this section we look into the conventional implementation, and we show an instability that is caused by the conventional implementation in conjunction with the quasi-bubble scheme.

In the conventional implementation the open boundary is set by dropping the continuity equation associated with the open boundary nodes at which the surface elevation is given. This procedure may reduce available information, which is described by the dropped continuity equation. We assert that the lack of information about mass continuity degrades the order of accuracy at the boundary node when one adopts the quasi bubble-function method and, possibly, other mixed methods as well.

We support this hypothesis by showing results of a test problem, which was investigated by Kolar *et al.* [7]. The test problem is depicted in Figure 2. Conditions for the problem

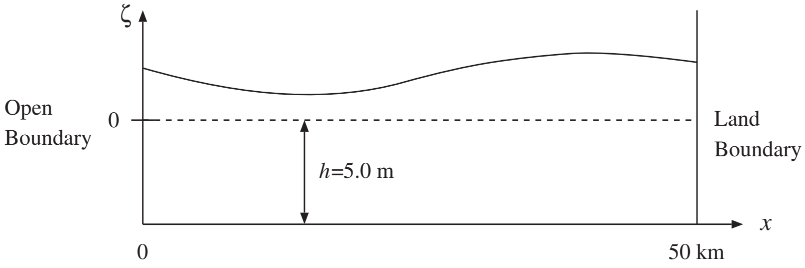

Figure 2. Test problem.

Table I. Conditions for the test problem.

Channel co-ordinates	$0 \leqslant x \leqslant 50$ km
Channel water depth	5 m
Bottom friction τ	$0.0001\ \mathrm{s}^{-1}$ (constant)
Simulation time	10 M_2-tidal cycles
Δt	Adjusted to maintain a Courant number $= 0.1$
Δx	2.5 km unless noted otherwise
$\lambda_{M_2}/\Delta x$	125 unless noted otherwise
Boundary conditions	$\zeta(0,t) = 1.0\ \sin(2\pi t/12.42\mathrm{h})$ m $u(50,t) = 0.0\ \mathrm{ms}^{-1}$
Initial conditions	Cold start: $\zeta(x,0) = u(x,0) = 0.0$

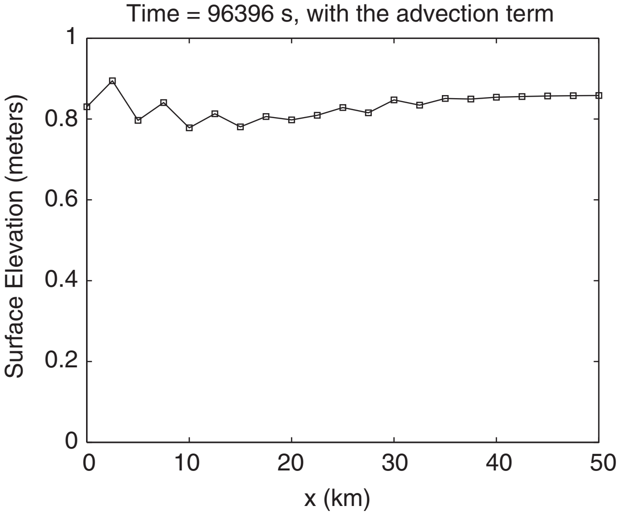

Figure 3. Snapshot of surface elevation at 96 396 s obtained with the conventional boundary condition implementation and with the advection term in the momentum equation.

are described in Table I. A land boundary is located at $x = 50$ km, and an open boundary is located at $x = 0$ km with an M_2 tide forcing of 1 m amplitude.

Snapshots of the results at 96 396 s are shown in Figures 3 and 4. As seen in these plots, the numerical solutions are degraded by unphysical or spurious oscillations. The calculation gets out of the range of double-precision at 98 538 s, 60 timesteps after the timestep of Figures 3 and 4. In order to determine the reason for the instability, we solved the same problem with the momentum equation in which the advection term is dropped. Figures 5 and 6 show snapshots at the same timestep as Figures 3 and 4. In this case the calculation is stable, and no apparent oscillations are observed. However, we note that the velocity shows a small bump at the first bubble-function node from the open boundary, as indicated in Figure 6. This bump may be related to the degraded accuracy at the open boundary node.

In order to determine the order of accuracy at each node, we apply Richardson Extrapolation to get an error estimate [9, 10]. The error estimate is obtained as follows. First we assume

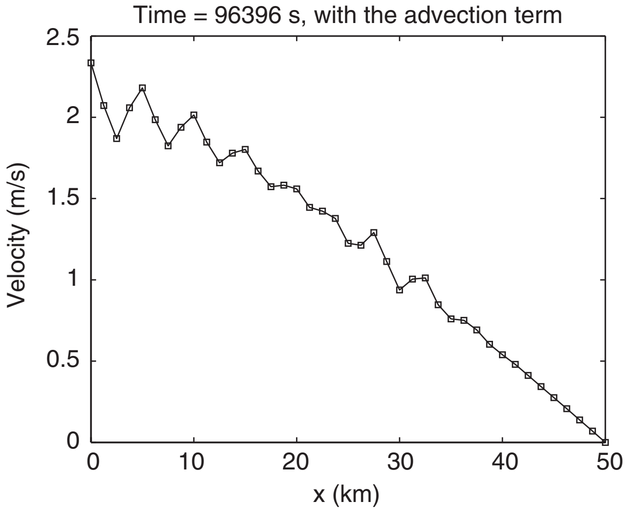

Figure 4. Snapshot of velocity at 96 396 s obtained with the conventional boundary condition implementation and with the advection term in the momentum equation.

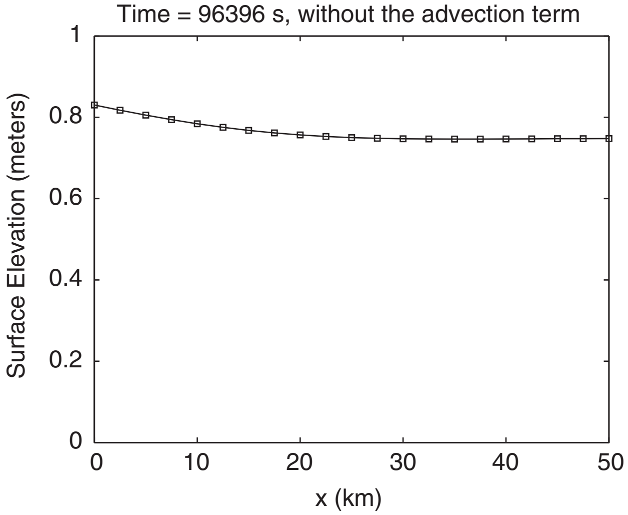

Figure 5. Snapshot of surface elevation at 96 396 s obtained with the conventional boundary condition implementation and without the advection term in the momentum equation.

that the exact value equals the approximate value plus given order errors, i.e.

$$y_e = y_h + ch^p + dh^{p+1} + eh^{p+2} + \cdots \tag{25}$$

where y_e is the exact solution, y_h is the approximate solution corresponding to spacing h. The coefficients c, d, and e depend on the derivatives in the interval. Assuming that the lowest

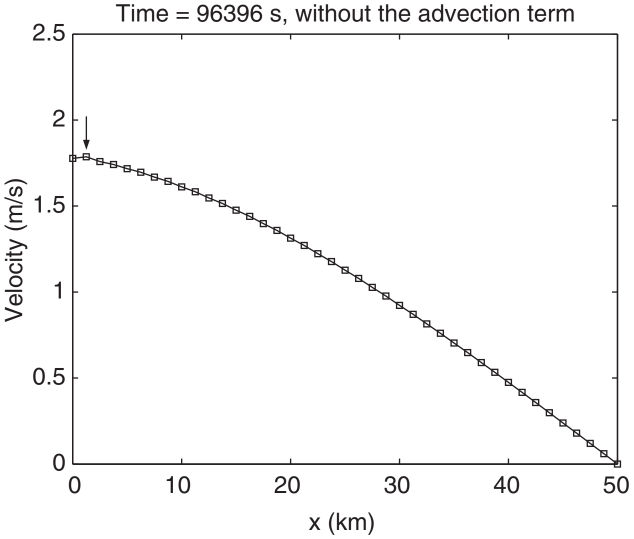

Figure 6. Snapshot of velocity at 96 396 s obtained with the conventional boundary condition implementation and without the advection term in the momentum equation.

order error term dominates subsequent terms in the error series, then for three different grid spacings h_1, h_2, and h_3, we have

$$\begin{aligned} y_e &= y_{h_1} + ch_1^p \\ &= y_{h_2} + ch_2^p \\ &= y_{h_3} + ch_3^p \end{aligned} \tag{26}$$

Given grid spacings h_i and the approximated solutions y_{h_i}, Equation (26) can be solved for p:

$$p = \ln\left(\frac{y_{h_2} - y_{h_1}}{y_{h_3} - y_{h_2}}\right) \Big/ \ln r \tag{27}$$

where we assumed that

$$\frac{h_1}{h_2} = \frac{h_2}{h_3} = r \tag{28}$$

In the following cases $r > 1$. We expect p to be equal to 2 because the quasi-bubble scheme adopts the piecewise linear interpolation. We then take a temporal average p_i' of p at each node i

$$p_i' = \sum_{t=0}^{N_T} p_i^t / N_T \tag{29}$$

where p_i^t is the estimated order of accuracy evaluated by Equation (27) at node i at timestep t, and N_T is the number of timesteps.

We calculated the same test problem without advection with three different grid spacings, $h_1 = 5000$, $h_2 = 1250$, and $h_3 = 312.5$m. The values of p_i' for surface elevation and velocity are

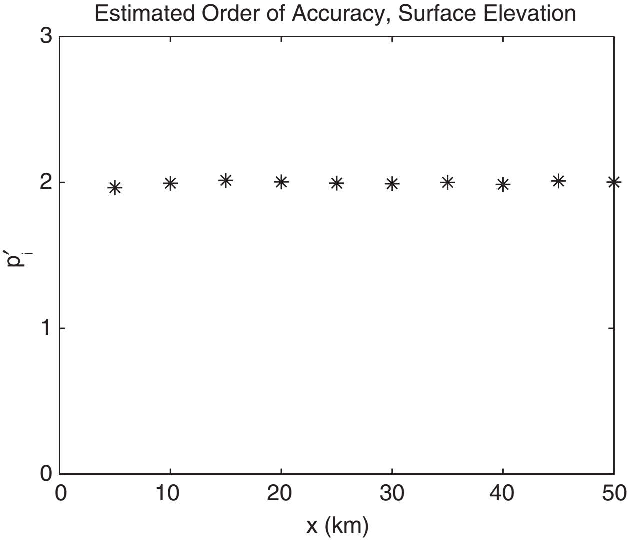

Figure 7. Estimated order of accuracy for surface elevation at each node for a conventional boundary condition implementation and without the advection term in the momentum equation.

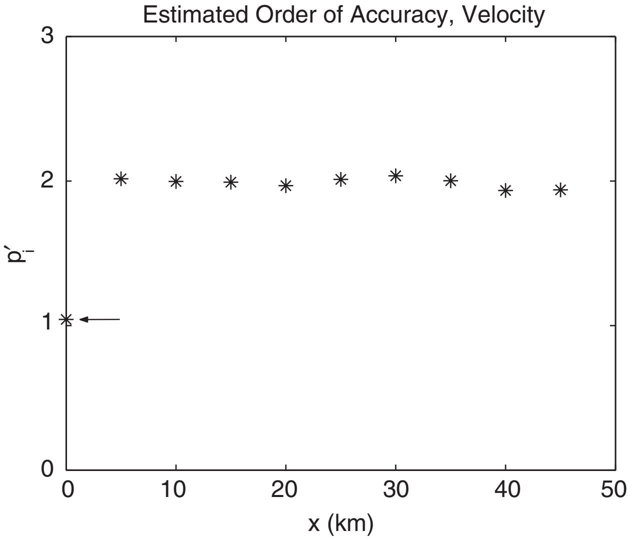

Figure 8. Estimated order of accuracy for velocity at each node for a conventional boundary condition implementation and without the advection term in the momentum equation.

shown in Figures 7 and 8, where p_i' are evaluated at each standard node in the coarsest grid. Nothing is plotted at $x = 0\,\mathrm{km}$ in Figure 7 or at $x = 50\,\mathrm{km}$ in Figure 8 because Equation (27) always gives zero at nodes where an essential boundary condition is given. What we observe in Figures 7 and 8 is that, while the surface elevation exhibits second-order accuracy over the whole domain, the velocity achieves a significantly lower order accuracy at $x = 0$ than at other nodes. We suggest that this degradation in accuracy of the velocity at the open boundary node originates in the implementation of the boundary condition, because the conventional implementation of essential conditions ignores one of the simultaneous equations by setting the test function at the boundary to zero. It may also be hypothesized that the quasi-bubble method, and probably other mixed methods as well, are more significantly affected by the lack of the condition because one nodal continuity equation is related to a larger number of nodal momentum equations than in other standard schemes. The cases presented here and the precedence considerations indicate that the lower-order accuracy in velocity at the open boundary must be remedied in some way to eliminate the instability arising from the advection term.

2.3. Alternative implementations

In standard Galerkin formulations, Green's theorem is applied to terms that include second or higher-order derivatives. On the other hand, in the unique formulation that was proposed by Kolar *et al.* [7], Green's theorem was applied to the mass flux term in the continuity equation, which is seen as the second term in Equation (5) in the 1D case. Applying Green's theorem, or integration by parts in 1D cases, to the second term of Equation (5), we have

$$\left(\frac{\partial\zeta}{\partial t},\phi_I\right)_\Omega - \left(Q,\frac{\mathrm{d}\phi_I}{\mathrm{d}x}\right)_\Omega + \langle Q,\phi_I\rangle_{\partial\Omega} = 0 \tag{30}$$

Kolar *et al.* suggested that the flux Q in the boundary integral term in Equation (30) should be interpreted as external to the computational domain. In other words, we have two different flux values at each boundary node; one of them is defined for the internal domain, and the other one is defined for the external domain. Denoting the flux Q for the external domain by $\hat{Q}$, and evaluating the boundary integral, Equation (30) yields

$$\left(\frac{\partial\zeta}{\partial t},\phi_I\right)_\Omega - \left(Q,\frac{\mathrm{d}\phi_I}{\mathrm{d}x}\right)_\Omega + \hat{Q}_N\phi_I(x_N) - \hat{Q}_0\phi_I(x_0) = 0 \tag{31}$$

where x_0 is the x co-ordinate at the left end of the domain, and x_N is the x co-ordinate at the right end of the domain. $\hat{Q}_0$ and $\hat{Q}_N$ represent the nodal values of the external flux $\hat{Q}$ at $x = x_0$ and $x = x_N$, respectively. In the following discussion, we assume that the left end of the domain is an open boundary, and the right end of the domain is a land boundary. In order to focus on the treatment of the open boundary, the land boundary is implemented in the conventional way. This is achieved by assuming $\hat{Q}_N = Q_N$, denoting the nodal value of the internal Q at the right end by Q_N. Equation (31) is then rewritten as

$$\left(\frac{\partial\zeta}{\partial t},\phi_I\right)_\Omega - \left(Q,\frac{\mathrm{d}\phi_I}{\mathrm{d}x}\right)_\Omega + Q_N\phi_I(x_N) - \hat{Q}_0\phi_I(x_0) = 0 \tag{32}$$

Further, for $I = 0$,

$$\left(\frac{\partial\zeta}{\partial t},\phi_0\right)_\Omega - \left(Q,\frac{\mathrm{d}\phi_0}{\mathrm{d}x}\right)_\Omega - \hat{Q}_0 = 0 \tag{33}$$

because $\phi_0(x_0)=1$ and $\phi_0(x_N)=0$. The external flux $\hat{Q}_0$ can be treated as a new unknown, because ζ_0 is known from the open boundary condition. Thus, the number of unknowns is consistent with the rank of the set of simultaneous equations.

The treatment of the boundary condition above is so far the same as Kolar *et al.* introduced in Reference [7]. They succeeded in dramatically improving the global mass conservation property with this unique implementation.

We extend their idea further to the momentum equation and utilize the newly obtained $\hat{Q}_0$ in order to get an accurate integration of the advection term. Applying integration by parts to the advection term in Equation (6), we have

$$\left(\frac{\partial u}{\partial t},\psi_J\right)_\Omega-\left(u,\frac{\partial}{\partial x}(u\psi_J)\right)_\Omega+\left(g\frac{\partial\zeta}{\partial x},\psi_J\right)_\Omega+(\tau u,\psi_J)_\Omega+\langle u,u\psi_J\rangle_{\partial\Omega}=0 \tag{34}$$

which leads to

$$\left(\frac{\partial u}{\partial t},\psi_J\right)_\Omega-\left(u\frac{\partial u}{\partial x},\psi_J\right)_\Omega-\left(u^2,\frac{\mathrm{d}\psi_J}{\mathrm{d}x}\right)_\Omega+\left(g\frac{\partial\zeta}{\partial x},\psi_J\right)_\Omega+(\tau u,\psi_J)_\Omega+\langle u^2,\psi_J\rangle_{\partial\Omega}=0 \tag{35}$$

Following a similar approach to that of Kolar *et al.* [7], it can be shown that u in the boundary integration term in Equation (35) should be viewed as external to the domain. Denoting u external by $\hat{u}$ and evaluating the boundary integration, Equation (35) is rewritten as

$$\left(\frac{\partial u}{\partial t},\psi_J\right)_\Omega-\left(u\frac{\partial u}{\partial x},\psi_J\right)_\Omega-\left(u^2,\frac{\mathrm{d}\psi_J}{\mathrm{d}x}\right)_\Omega+\left(g\frac{\partial\zeta}{\partial x},\psi_J\right)_\Omega$$
$$+(\tau u,\psi_J)_\Omega+u_N^2\psi_J(x_N)-\hat{u}_0^2\psi_J(x_0)=0 \tag{36}$$

where we assumed $\hat{u}_N=u_N$ for the same reason as we set $\hat{Q}_N$ to Q_N. $\hat{u}_0$ is obtained from

$$\hat{u}_0=\frac{\hat{Q}_0}{H_0} \tag{37}$$

where $H_0=\zeta_0+h_0$, and ζ_0 and h_0 are given by the boundary condition and the problem definition, respectively. For $J=0$, Equation (36) can be rewritten as

$$\left(\frac{\partial u}{\partial t},\psi_0\right)_\Omega-\left(u\frac{\partial u}{\partial x},\psi_0\right)_\Omega-\left(u^2,\frac{\mathrm{d}\psi_0}{\mathrm{d}x}\right)_\Omega+\left(g\frac{\partial\zeta}{\partial x},\psi_0\right)_\Omega+(\tau u,\psi_0)_\Omega-\hat{u}_0^2=0 \tag{38}$$

because $\psi_0(x_0)=1$ and $\psi_0(x_N)=0$. We expect that utilizing $\hat{Q}_0$ obtained from Equation (33), which is absent in the conventional implementation, improves the accuracy of the advection term. Since the flux Q and the velocity u are discontinuous at the boundary, we refer to this implementation as a discontinuous boundary (DB) implementation.

We first tested the DB with the problem described in Section 2.2 excluding the advection term in the momentum equation. The results show that the proposed implementation exhibits approximately second-order accuracy for velocity at $x=0$, which used to be first-order when we tested the conventional implementation. However, when we tested the DB implementation with the advection term, we encountered a kind of instability, which is different from what we saw in the conventional implementation. After several stabilizing techniques were tried in order to remedy this instability, we found that an upwinded definition of $\hat{u}_0$ leads to a stable scheme. Thus we upwind $\hat{u}_0$ as follows

$$\hat{u}_0 = \begin{cases} \hat{Q}_0/H_0 & \text{if } \hat{Q}_0 \geqslant 0 \\ Q_0/H_0 & \text{if } \hat{Q}_0 < 0 \end{cases} \tag{39}$$

A similar technique can be found in the context of the discontinuous Galerkin method [11]. Since $\hat{u}_0$ is upwinded along the streamline, we refer to the modified DB implementation as the stream-line upwind discontinuous boundary (SUDB) implementation.

Recalling the element-wise interpolations defined in Equations (12)–(15), Equation (36) yields an element-wise discretization, i.e.

$$\left(\frac{\partial u_m^{(e)}}{\partial t}\psi_m^{(e)}, \psi_j^{(e)}\right)_{\Omega^{(e)}} - \left(u_m^{(e)} u_n^{(e)} \psi_m^{(e)} \frac{\mathrm{d}\psi_n^{(e)}}{\mathrm{d}x}, \psi_j^{(e)}\right)_{\Omega^{(e)}} - \left(u_m^{(e)^2} \psi_m^{(e)}, \frac{\mathrm{d}\psi_j^{(e)}}{\mathrm{d}x}\right)_{\Omega^{(e)}}$$

$$+ \left(g\zeta_l^{(e)} \frac{\partial \phi_l^{(e)}}{\partial x}, \psi_j^{(e)}\right)_{\Omega^{(e)}} + (\tau u_m^{(e)} \psi_m^{(e)}, \psi_j^{(e)})_{\Omega^{(e)}} + u_N^2 \psi_J(x_N) - \hat{u}_0^2 \psi_J(x_0) \tag{40}$$

We then have element-matrix formulations of the approximated momentum equation in the SUDB implementation

$$\mathbf{MU}_{jm}^{(e)} \frac{\partial u_m^{(e)}}{\partial t} - \mathbf{C}_{jm}^{(e)}(\mathbf{u}^{(e)}) u_m^{(e)} - \mathbf{E}_{jm}^{(e)} u_m^{(e)^2} + g\mathbf{D}_{jl}^{(e)} \zeta_l^{(e)} \tag{41}$$

$$+ \tau \mathbf{MU}_{jm}^{(e)} u_m^{(e)} + u_N^2 \psi_J(x_N) - \hat{u}_0^2 \psi_J(x_0) = 0 \tag{42}$$

where $\mathbf{MU}_{jm}^{(e)}$, $\mathbf{C}_{jm}^{(e)}$, and $\mathbf{D}_{jl}^{(e)}$ are defined in Equations (22)–(24), respectively. $\mathbf{E}_{jm}^{(e)}$ is defined as

$$\mathbf{E}_{jm}^{(e)} = \left(\frac{\mathrm{d}\psi_j^{(e)}}{\mathrm{d}x}, \psi_m^{(e)}\right)_{\Omega^{(e)}} = \frac{1}{2}\begin{bmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \tag{43}$$

Some numerical results of the proposed implementation will be shown in the next section.

3. NUMERICAL VERIFICATION

The same problem described in Section 2.2 was tested using the proposed formulation including the advection term. Figures 9 and 10 are snapshots for surface elevation and velocity, respectively, at the same timestep as Figures 3–6. The solution is very smooth and there is no bump as in Figure 6. The orders of accuracy estimated by Equation (29) are shown in Figures 11 and 12. Solutions for three grid spacings, i.e. $h_1 = 5000$, $h_2 = 1250$, and $h_3 = 312.5$ in metres, were used to evaluate p_i'. We see some small perturbations in Figures 11 and 12.

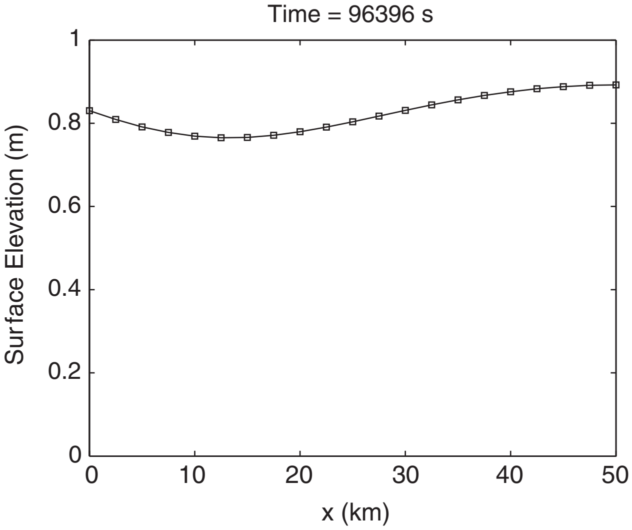

Figure 9. Snapshot of surface elevation at 96 396 s obtained with the SUDB implementation and with the advection term in the momentum equation.

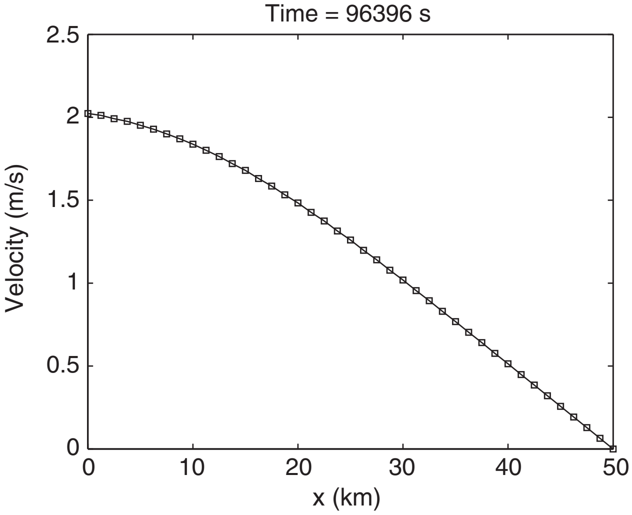

Figure 10. Snapshot of velocity at 96 396 s obtained with the SUDB implementation and with the advection term in the momentum equation.

This might be due to the advection term, which was not considered in the case shown in Figures 5 and 6. Despite the small perturbations in the convergence rates, the solutions are smooth, and we have not observed any instabilities with the proposed scheme. In addition, a significant improvement is observed for velocity at $x = 0$. The order of accuracy for velocity

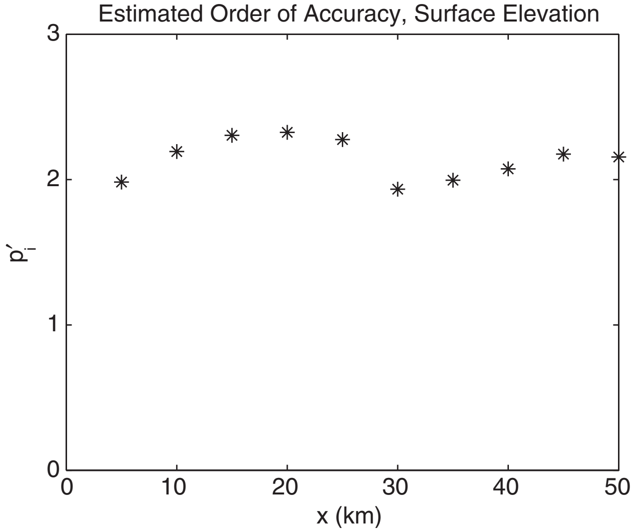

Figure 11. Estimated order of accuracy for surface elevation at each node for the SUDB implementation with the advection term in the momentum equation.

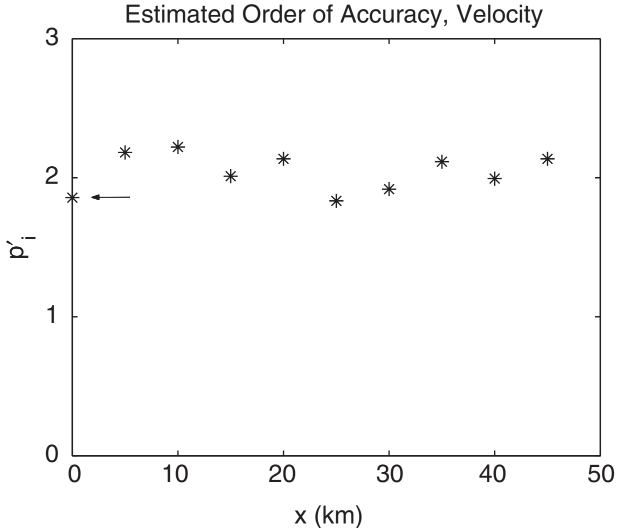

Figure 12. Estimated order of accuracy for velocity at each node for the SUDB implementation with the advection term in the momentum equation.

at $x = 0$ has improved to 1.86 from 1.04 in Figure 8. We suggest that the more accurate velocity value at the boundary stabilizes the scheme by giving a more accurate evaluation of the advection term.

In order to check the convergence rate and the consistency of the SUDB method, we took the difference between solutions calculated with various grid spacings and a solution obtained

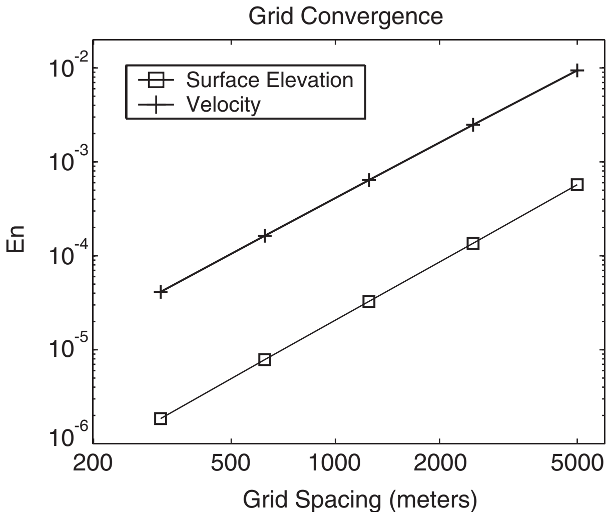

Figure 13. Grid convergence study over the domain for the SUDB implementation with the advection term in the momentum equation.

with a very fine grid. We assume that the solution obtained with the fine grid approximates the exact solution with a sufficiently small discrepancy. Letting h_{c} and h_{f} denote the grid spacings of the coarse and fine grids, respectively, and representing solutions with those grids by y_{c} and y_{f}, respectively, we define an error indicator for the coarse grid as

$$E_n = \frac{1}{N_T} \sum_{k=1}^{N_T} \frac{1}{N} \sqrt{\sum_{i=1}^{N} (|y_{\mathrm{c}} - y_{\mathrm{f}}|_{x_i,t_k})^2} \tag{44}$$

which represents a time-averaged root-mean-squared error. In Equation (44), N_T is the number of timesteps of the coarse grid, N is the number of nodes of the coarse grid, and y_{c} and y_{f} are either surface elevation or velocity. It is expected that E_n should decrease with the square of the grid spacing. We used five coarse grid spacings, $h_1 = 5000$, $h_2 = 2500$, $h_3 = 1250$, $h_4 = 625$, and $h_5 = 312.5$ m, to obtain y_{c}. The reference solution y_{f} is calculated with the fine grid spacing of 39.0625 m. The indicator E_n is plotted in Figure 13 for both surface elevation and velocity. The figure shows steady decreases, as we expected. The order of accuracy obtained from the slopes is 2.07 for surface elevation and 1.96 for velocity, which agrees well with the theoretical order of accuracy for linear finite elements.

Our remaining concern is to see whether the external flux $\hat{Q}_0$ approaches the internal flux Q_0 for higher resolution grids. We define another error indicator E_b as

$$E_b = \sum_{t=1}^{N_T} |\hat{Q}_0^t - Q_0^t|/N_T \tag{45}$$

which is the time-averaged difference between the external and internal flux. We used five grid spacings, $h_1 = 5000$, $h_2 = 2500$, $h_3 = 1250$, $h_4 = 625$, and $h_5 = 312.5$ m. The results are shown in Figure 14, which shows a steady convergence at the rate of 1.85. This rate approximately

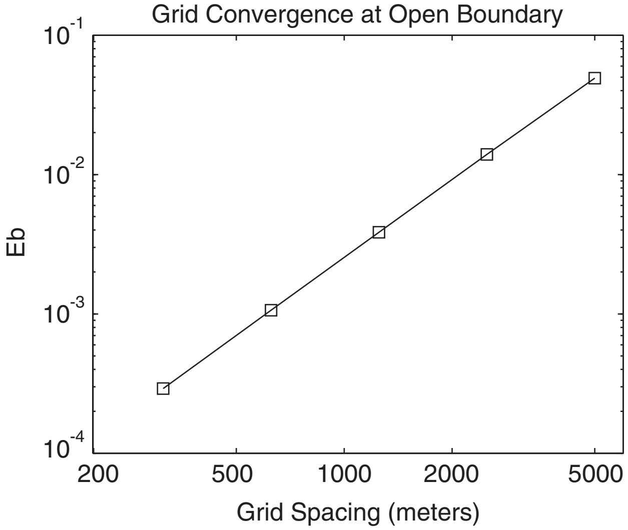

Figure 14. Grid convergence study for the duplication boundary flux values obtained with the SUDB implementation and with the advection term in the momentum equation.

agrees with the order of accuracy for velocity at the boundary node obtained by Equation (29) and seen in Figure 12.

These figures and convergence rates support the conclusion that the proposed method is consistent and approximately holds the theoretical order of accuracy expected for standard linear finite elements. Although the velocity at the boundary node shows a slightly lower order of accuracy, it has been significantly improved from what we obtained using the conventional implementation.

The most important improvement is that the proposed implementation eliminates spurious oscillations and stabilizes the quasi bubble-function scheme, which is unstable if the essential condition on surface elevation is implemented in the conventional way and advection is included.

4. CONCLUSIONS

The quasi-bubble scheme has an ideal monotonic dispersion property [6], which indicates that it should not generate non-physical or spurious oscillations. In addition, the use of the primitive form of the SWE and static condensation procedures make the scheme very efficient with respect to computational resources. However, our numerical results with the quasi-bubble scheme show that the conventional implementation of an essential surface-elevation boundary condition degrades the accuracy of the velocity at the boundary node, and causes the interior domain solution to deteriorate by introducing spurious oscillations. These spurious oscillations are destructive enough to make the scheme unstable. Our numerical experiments also show that the spurious oscillations occur in conjunction with the advection term. In order to remedy this instability, the SUDB implementation has been developed in this paper, which is obtained by extending the unique formulation introduced by Kolar *et al.* [7]. The flux from the external domain is defined by the continuity equation at the boundary node, which is usually dropped in the conventional implementation. The SUDB implementation enables us to utilize all conditions that can be obtained from the shallow water equations. The results of our numerical investigation show that, reformulating the momentum equation and using the flux from the external domain in the boundary integral derived from the advection term, the quasi-bubble scheme becomes stable and increases the order of accuracy at the boundary. The consistency of the proposed scheme is also confirmed by thorough numerical verification.

REFERENCES

1. Kolar RL, Westerink JJ. A look back at 20 years of gwc-based shallow water models. *Proceedings of the XIII International Conference on Computational Methods in Water Resources*, Calgary, Alberta, Canada, 2000; 899–906.
2. Westerink JJ, Gray WG. Progress in surface water modeling. *Reviews of Geophysics* (*Supplement*, *American Geophysical Union*) 1991; 210–217.
3. Mewis P, Holz K-P. A quasi bubble-function approach for shallow water waves. *Advances in Hydro-Science and Engineering* 1993; **1**:768–774.
4. Galland JC, Goutal N, Hervouet JM. Telemac: a new numerical model for solving the shallow water equations. *Advances in Water Resources* 1991; **14**:138–148.
5. Hervouet JM. On spurious oscillations in primitive shallow water equations. *The XIII International Conference on Computational Methods in Water Resources*, Calgary, Alberta, Canada, 2000; 929–936.
6. Atkinson JH, Westerink JJ, Hervouet JM. Similarities between the quasi-bubble and the generalized wave continuity equations to the shallow water equations. *International Journal for Numerical Methods in Fluids* 2004; **45**:689–714.
7. Kolar RL, Gray WG, Westerink JJ. Boundary conditions in shallow water models—an alternative implementation for finite element codes. *International Journal for Numerical Methods in Fluids* 1996; **22**:603–618.
8. Franca LP, Farhat C. On the limitation of bubble functions. *Computer Methods in Applied Mechanics and Engineering* 1994; **117**:225–230.
9. Roache PJ. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* 1994; **116**:405–413.
10. Roache PJ. *Verification and Validation in Computational Science and Engineering*, Chapter 5. Hermosa Publishers: Albuquerque, New Mexico, U.S.A., 1998.
11. Aizinger V, Dawson C. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources* 2002; **25**:67–84.