A moving boundary approach for one-dimensional free surface flows

Scott A. Yost *, Prasada Rao

Department of Civil Engineering, The University of Kentucky, 161 CE/KTC Bldg., Lexington, KY 40506-0281, USA

Received 15 August 1998; received in revised form 4 June 1999; accepted 23 June 1999

Abstract

Numerical analysis of open channel flows has largely confined to using fixed boundaries in the numerical codes. Given the size of the problem domain, using fixed boundaries often taxes the computational resources since the effect of the boundary is transferred on to its adjacent node in the first few iterations. In this work, we explore the performance of a moving boundary in conjunction with central differencing (classical Runge Kutta (RK)) and high resolute (Essentially Non-Oscillatory (ENO)) shock capturing schemes. The performance of this algorithm is tested with flows containing discontinuities. By correlating the velocity of the boundary to the location of the wave front, the results obtained indicate that no flow information is lost. Moving the boundary with time helps in constricting the size of flow domain and thus accelerating the solution to the desired transient state.

Keywords: Runge Kutta; ENO; Explicit; Moving boundary; Shocks; Open channels

1. Introduction

The past two decades saw the advent and wide spread application of numerical techniques for open channel flows. While the formulation, advantages and limitations of these approaches are well documented (Hirsch [1] and Chaudhry [2]), a unifying feature of these techniques in the field of hydraulic engineering is that they generally use a fixed grid. Of course this can be in conjunction with a uniform, non-uniform or evolving grid spacing. In any case a stationary boundary condition is standard. This work differs from all the other approaches in that we use a moving boundary approach in the numerical algorithm. The boundary moves as a function of time depending on local stationary conditions. Such a phenomenon helps in reducing the computational length of the channel and hence the required computational effort. This approach can be though of as a hybrid formulation to the traditional Eulerian and Lagrangian implementations. Namely, the computations are consistent with an Eulerian formulation but the boundary evolves in a Lagrangian sense.

The motivation of this work can best be described by illustrating the advantages in this approach. With reference to Fig. 1a, given a channel length and initial conditions, the specific boundary condition is a sudden closure of gate at the downstream end. This type of boundary reduces the flow velocity (to zero) leading to a corresponding increase of flow depth. The subsequent wave, called a negative surge, travels in the upstream direction. Fennema and Chaudhry [3] first studied this problem with the objective of demonstrating the shock capturing ability of various numerical schemes. Jha et al. [4] and Yost and Rao [5] further investigated the performance of their numerical algorithms on this test problem. All the above applications used a fixed and constant grid spacing while numerically integrating the flow equations for the entire computational domain. Since the length of the channel is very large, solving the flow equations for the entire length is computationally intensive, and often not required, especially when considering the resolution needed at the shock location. The physics of the problem dictate that the effect of the boundary is felt by its adjacent grid node in the first few time steps, after which the flow conditions remains unaltered (viz., steady and uniform flow exists). As the solution is advanced over time, the negative surge propagates further upstream resulting in uniform flow conditions behind the wave front. At time $t_1$, the wave has reached the mid-length of the channel (Fig. 1b).

Standard numerical algorithms continue to compute the flow variables within the length $L_2$, which do not change, taxing the computational resources. On the
other hand, moving the downstream boundary in the direction of the surge wave aids in truncating the flow domain. This reduces the CPU requirements with no deterioration of computed solution provided that the speed with which the boundary moves is consistent with a Lagrangian formulation, accounting for uniform and steady conditions. This discussion will continue in later sections when we consider various factors that affect the velocity of the boundary.

The rapid development of numerical schemes saw the advent of different strategies in speeding up the convergence of solutions. Some of the more recent techniques for speeding convergence include Multigrid techniques [6,7] and Radiation boundary methods [8,9]. The radiation techniques, also known as absorbing boundary conditions, are the closest to the present formulation. Radiation methods have found application in problems where an infinite domain is truncated to a finite domain. The effectiveness of these approaches relies heavily on the radiation/absorbing boundary equation. This equation translates the known boundary condition from the boundary of the initial infinite domain to the new truncated boundary. There are limitations to this methodology. First it can only be used for steady state simulation [8]. Secondly, for flow with shocks, truncating the domain to a desired length requires a priori knowledge of the steady state solution to ascertain that no discontinuities are formed in the truncated region. As the present approach involves moving the boundary as a function of time, and hence is related to the flow variables, it can be used for transient simulations. Moreover, as mentioned by Ali and Turkel [9], for flows with discontinuities, the advantage of using an absorbing boundary condition can be lost. For simplicity both the radiation techniques and this approach assume that the boundary condition is time independent, albeit not essential.

The objective of this paper is to numerically show that a moving boundary approach can satisfactorily simulate open channel flows accompanied with discontinuities. The use of a moving boundary accelerates the solution and constrains the size of the problem. Computationally, this saves computer storage and CPU time. In conjugation with the boundary condition, we have solved the flow equations using a second order Runge Kutta (RK) scheme and an Essentially Non-Oscillatory (ENO) scheme. While the RK scheme is a member of the finite difference family of schemes, the ENO scheme is a subset of high-resolution schemes [1]. We compare the performance of these two numerical techniques in hopes of providing some insight into their use. While much of the discussion in this paper is devoted to the moving boundary approach, to maintain continuity we have briefly touched upon the other numerical aspects with most details left as references.

2. Governing equations

For a rectangular channel the basic governing equations, based on the continuity and momentum principles, can be written in conservation form as [2]

\[ \frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} = 0, \tag{1} \]

\[ \frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2 + \frac{gh^2}{2})}{\partial x} = gh(S_0 - S_f), \tag{2} \]

where \( h \) is the flow depth, \( u \) the flow velocity, \( g \) the acceleration due to gravity, \( S_0 \) the bed slope of the channel and \( S_f \) is the frictional slope. The subscripts represent...
partial derivatives with respect to the subscripted variable. The frictional losses are computed using Manning’s equation, which can be written as

\[ S_t = \frac{m^2 u |u|}{h^{5/3}} \] (3)

where \( m \) is the Manning’s roughness coefficient. For smooth channels, the roughness coefficient is zero. The definition sketch illustrating the flow variables is shown in Fig. 2. In matrix notation Eqs. (1) and (2) reduce to

\[ \frac{\partial}{\partial t} [U] + \frac{\partial}{\partial x} [F] = [S] \] (4)

with the elements of matrices being \([U] = [h u u^2 + g h^2/2]^T\) and \([S] = [0 \ g h(S_0 - S_t)]^T\).

3. Numerical schemes

To facilitate a comparison between the central differencing and high-resolution formulations, we provide a self-contained description of a RK and ENO scheme on a uniform grid. There are significant differences between the two techniques. Solutions with the standard finite difference schemes result in non-monotone (oscillatory) solutions. While first order schemes diffuse the wave front, second and higher order schemes produce dispersive errors (also addressed to as “wiggles” and “spurious oscillations”) in the vicinity of the discontinuity [10]. To overcome this limitation, research in the early 80’s produced high-resolution shock capturing schemes [1] (i.e., ENO), resulting in monotone solutions.

3.1. RK formulation

The family of RK methods is perhaps the oldest technique for solving differential equations. Though the performance of different variants of finite difference family of schemes has been investigated for open channel flows, MacCormack in particular, no study was reported on the efficacy of RK methods. It was Jameson et al. [11] who first introduced the RK methods for solving the Euler equations in aerodynamics. In RK techniques, higher order accuracy in time is achieved by splitting the computation between the time levels \((n, n+1)\) [1]. Denoting the residual from Eq. (4) as

\[ \frac{\partial U}{\partial t} = R(U^n) = \frac{1}{2\Delta x} (F^n_{i+1} - F^n_{i-1}) - [S]^n \] (5)

the 2-step procedure can be written as,

\[ U^{(1)}_{i} = U^n_{i} - \alpha_1 \Delta t R(U^n, \dot{U}^n), \] (6a)

\[ U^{n+1}_{i} = U^n_{i} - \alpha_2 \Delta t R(U^{(1)}), \] (6b)

Second order accuracy can be obtained by assigning the values of \( \alpha_1 \) and \( \alpha_2 \) as 0.5 and 1.0, respectively. With the given initial conditions, Eq. (6b) can be used to compute the unknowns at all the interior grid nodes. As mentioned earlier, a characteristic feature of second and higher order schemes lies in producing dispersive errors near the vicinity of the discontinuity. These errors manifest as numerical oscillations and need to be smoothed for proper functioning of the code. This smoothing mechanism could either be naturally present in the equations or added to the code as a post-processor step. We used the procedure originally suggested by Jameson et al. [11] As per this approach, the flow variable \( f' \) in this discussion is post-process smoothed as,

\[ f'^{n+1}_i = f'^{n+1}_i + \xi_{i+1/2}(f'^{n+1}_{i+1} - f'^{n+1}_i) - \xi_{i-1/2}(f'^{n+1}_i - f'^{n+1}_{i-1}), \] (7)

where

\[ \xi_i = \frac{|h_{i+1} - 2h_i + h_{i-1}|}{|h_{i+1}| + |h_{i-1}|}, \] (8)

\[ \xi_{i+1/2} = \mu \max(\xi_i, \xi_{i+1}). \] (9)

The solution obtained by Eq. (7) is relatively smooth. The smoothing mechanism, as seen by Eq. (8), is triggered only in oscillatory regions. In regions where the flow is uniform, Eq. (8) vanishes. The parameter \( \mu \) in Eq. (9) is the dissipation constant, the magnitude of which depends on the intensity of the shocks.

3.2. ENO scheme

Shu and Osher [12] introduced the non-oscillatory formulations for solving the Euler equations in gas dynamics. The family of TVD schemes constitutes a smaller subset of the ENO schemes. Based on the close similarity between the Euler equations and Shallow water flow equations Eq. (4), Yost and Rao [5] investigated the performance of ENO schemes for this set of equations. Following Shu and Osher [12], the fluxes at any grid node are decomposed as

\[ F_i = F^+_i + F^-_i \] (10)

with \( F^\pm_i = 0.5(F_i \pm \alpha U_i) \) where \( \alpha = \max(\lambda_{i.1}) \) and \( \lambda_{i.1} \) represent the characteristic speeds of the Jacobian matrix. For the equations represented by Eq. (4), the Jacobian flux matrix is \( \partial F/\partial U \), with eigen values of \( \lambda_{1.2} = u \pm g h \).
A smooth Lax-Friedrich scheme can then be constructed by defining the fluxes at cell interfaces as

$$F_{i+1/2} = F_{i+1/2}^+ + F_{i-1/2}^-$$  \hspace{1cm} (11)

with

$$F_{i+1/2}^+ = F_i^+ + 0.5\Delta F_i^+ , \quad F_{i-1/2}^- = F_{i-1}^- - 0.5\Delta F_{i-1}^- .$$

The value $\Delta F$ gives the limited fluxes and is computed using the min mod function. The limited fluxes are expressed as

$$\Delta F_i^+ = \text{min mod}(F_{i+1}^+ - F_i^+, F_i^+ - F_{i-1}^+) ,$$

$$\Delta F_i^- = \text{min mod}(F_{i-1}^-, F_{i+1}^- - F_i^-).$$  \hspace{1cm} (12)

The min mod operator is defined as

$$\text{min mod}(x,y) = \frac{\text{sgn}(x) + \text{sgn}(y)}{2} \min(|x|, |y|).$$  \hspace{1cm} (13)

The flow variables at the new time level are computed by

$$U^{n+1}_i = U^n_i - \frac{\Delta t}{\Delta x} (F_{i+1/2} - F_{i-1/2}) + \Delta t [S].$$  \hspace{1cm} (14)

The solution obtained by Eq. (14) is conservative, free from oscillations, and less sensitive to grid spacing \[5\] as compared to the standard finite difference formulation.

4. Stability criteria

A characteristic feature of explicit schemes is the limits on the time step, which is governed from the stability criteria. The magnitude of time step, given by the well-known CFL stability condition \[2\], can be written as

$$\Delta t = C_n \frac{\Delta x}{\max(|u| + \sqrt{gh})},$$  \hspace{1cm} (15)

where $C_n$ is the Courant number (\(\leq 1\)), and $\Delta x$ is the grid spacing. With known grid spacing and flow conditions the time step is evaluated using Eq. (15).

5. Moving boundary algorithm

Moving the boundary in the direction of flow at an appropriate rate is the most important aspect for a satisfactory simulation. The boundary should not move faster than the velocity of the wave for it could result in wrong location of the wave front. This is comparable to an Lagrangian formulation where the boundary moves with fluid particles. While our boundary movement is independent of the fluid movement, one can think of an Lagrangian type computation as being an upper limit on the boundary movement. The real issue is knowing exactly how quickly the boundary will move. If the analytical velocity of the surge wave is known a priori, then assigning a suitable velocity to the boundary poses no problem. We do not utilize the analytical velocity of the wave, if known. There exists two approaches in calculating the velocity of boundary. First, for standard explicit formulations, the wave front advances by one grid node per time step (at least theoretically with a Courant Number of 1). Hence this alternative reduces the total grid nodes by a factor of one per iteration. The velocity of boundary is given as $\Delta x/\Delta t$. However since the stability condition is based on linearized flow equations, values for $C_n$ are often less than 1.0. Thus reducing the domain by one grid node overestimates the boundary velocity. The results we present in the later section demonstrate that this approach does not work well for flows with shocks.

The second approach tracks the location of surge as a function of time. Given the position of the surge, the flow domain can be truncated after a certain number of grid nodes by utilizing uniform flow conditions behind the shock. Our analysis uses this approach. Identifying how to move the boundary can be accomplished in two ways. In uniform flow conditions, the depth and discharge do not change along the channel. If $N$ indicates the location of the last grid node in the computational domain, then $N$ can be reduced by tracking the local deviation of flow depth as (see Appendix A for computer code implementation)

For each $N$ such that $(|h^n_N - h^{n-2}_N| \leq \varepsilon) ,$

then $N - 1 \Rightarrow N.$  \hspace{1cm} (16)

For $N$ such that $(|h^n_N - h^{n-2}_N| < \varepsilon),$  \hspace{1cm} (17)

then $L - \sqrt{gh^{n-1}_N} \Delta t \Rightarrow L$ and $L/\Delta x + 1 \Rightarrow N$.

Here $L$ is the length of the channel, $\Delta x$ the grid spacing, $L/\Delta x$ the number of interior grid nodes, and $\varepsilon$ is the allowable deviation from uniform flow conditions. The magnitude of $\varepsilon$ used is on the order of 0.01. The reason why the flow depth at node $N$ is compared with node $N-2$ is that the linearized flow equations are recast into characteristic form at the boundary nodes. Discretizing the characteristic equation at the end boundary node involves using the variables at the adjacent node. Hence it is consistent to check the deviation of depth between nodes $N$ and $N-2$. Additional results obtained by replacing $N-2$ with other choices (i.e., $N-3$ and $N-4$, etc.) did not change the flow profiles, although the location of downstream boundary was slightly affected.

As previously explained, Eqs. (16) and (17) truncates the flow domain once uniform flow conditions are established. Since the objective is in reducing the size of domain, one could also compare flow depths at boundary node with respect to time levels, checking if steady-state condition has been established. Mathematically this can be written as (see Appendix A, Eq. (A.3) for corresponding computer coding).
For \( N \) such that \((|h_{n+1}^N - h_n^N| \leq \varepsilon)\), then \( L \)
\[ - \sqrt{gh_{n+1}^N \Delta t} \]
\[ \Rightarrow L \text{ and } L/\Delta x + 1 \Rightarrow N. \]  

(18)

Computing the moving boundary with Eqs. (17) and (18) gave similar results except for flows with sharp streamline curvature at a boundary. This scenario could happen if the physical boundary was a weir or overfall creating a drawdown curve. In this case Eq. (17) fails, even for fairly small values of \( \varepsilon \) because the depth changes significantly over this localized region. Since Eq. (18) compare the depth at a given point between time levels, flow can become stationary but not necessarily uniform.

6. Application

The performance of RK and ENO schemes, coupled with Eqs. (16)–(18) for moving boundaries, was investigated on a variety of flows accompanied with shocks. To test the moving boundary strategy we selected flows with shocks since they are particularly sensitive to boundary information. The definition sketch of the first test problem is illustrated in Fig. 3. The initial conditions in the 5000 m \((\Delta x = 5 \text{ m})\) long channel are a depth of 6 m and a velocity of 3.125 m/s. To compare the obtained solution with the analytical one, we first assume the channel is horizontal \((S_0 = 0)\) and frictionless \((m = 0)\). At time \( t = 0^+ \), the gate at the downstream end is closed. This results in a zero flow velocity and an increase of flow depth (surge) at this location. As the solution marches in time, the surge propagates upstream. The analytical solution [13] indicates that the height of the surge is 8.66 m and its celerity is \(-7.86 \text{ m/s}\). Numerical comparisons were made at \( \tau \approx 318.2 \text{ s} \) when the surge is at mid-channel. Fig. 4 shows the transient computed profile with the analytical solution. A dissipation constant \((\mu \text{ in Eq. } (9))\) was 0.6. Based on numerical experiments this value was found to be optimal. While varying the value of \( \mu \) did not change the location of the wave front, the solution did contain more oscillations at other values of \( \mu \). At the end of the computations, the moving downstream boundary was located at 2520 m for the ENO scheme and at 2510 m in the RK formulation. The small amount of shock smearing with the RK solution is characteristic of second order schemes coupled with the smoothening mechanism. In the evaluation of variables at the boundary node, a zero flow velocity was specified at all time levels. The flow depth was computed using the \( C^+ \) characteristic equation, written as [2]
\[ h_{n+1}^N = h_n^{N-1} + \sqrt{\frac{h_{n+1}^{N-1}}{g}}. \]  

(19)

The location of moving downstream boundary, using the uniform flow criteria Eq. (17), is shown in Fig. 5 as a function of time. The plot indicates that the flow domain is substantially constricted during the computations. With a Courant number of 0.6, 1275 time steps were required for the shock to arrive mid-channel. The time step \((\Delta t)\) evaluated from the stability condition was 0.277 s. A moving boundary approach coupled with the ENO formulation reflected in a CPU saving of 30\%. The corresponding saving with the RK formulation was of the order of 37\%.

Note that constricting the domain at a set rate of one node per iteration would move the boundary at a velocity of 18.05 m/s (\(\Delta x/\Delta t\)), which is larger than the shock velocity (7.86 m/s). Since the specified moving strategy results in a wrong solution, this further justifies formulations in Eqs. (17) and (18). In addition, varying the value of \( \varepsilon \) which represents the allowable depth tolerance from its uniform value, did not change the general pattern of the results. Fig. 6 shows the sensitivity of the formulation with respect to \( \varepsilon \). Though no definite conclusion can be drawn from the plot, one can see that the relative error for the range of \( \varepsilon \) considered is small. The relative error (RE) was computed as
\[ \text{RE} = \frac{1}{N} \sum_{i=1}^{N} (h_i - h_{\text{exact}})^2. \]  

(20)

The values of flow depth in the vicinity of the discontinuity are documented in Table 1. This table includes the results with and without the moving boundary algorithm for the two schemes and for the two truncating criteria Eq. (17) or Eq. (18). The arrows
indicate the transient location of the analytical wave [13] for the said time period.

While the results are promising, real channels are not frictionless (smooth). Next we tested frictional (rough) channels. For the flow problem, shown in Fig. 3, a roughness coefficient \( m \) in Eq. (3) of 0.004 was used. Since there is no analytical solution, plots of the depth profiles for various \( \varepsilon \) are shown in Figs. 7–9 along with fixed boundary computations. The result indicates that the presence of the source term does not alter the effectiveness of the present formulation, as evident in the computed location of the discontinuity.

The second test case relates to the surge arising from a sudden opening of a gate. The definition sketch of the problem is illustrated in Fig. 10. The initial conditions in the 1000 m long channel are one-meter depth with a zero velocity. A grid spacing of 4 m, Courant number of 0.9 and a dissipation constant of 0.4 were selected. At time, \( t = 0^+ \), the flow depth at the upstream end was increased to 2.5, indicating a sudden opening of gate. This results in a positive surge propagating downstream. Fig. 11 is the transient plot at \( t \approx 100 \) s, for \( \varepsilon = 0.01 \). The boundary at the upstream end moved based on the uniform flow criteria Eq. (17). There is no significant difference in the results when using formulation 17 or 18 since the gradient of streamlines is small (representing steady uniform flow conditions). The close agreement between the three solutions further shows the robustness of the moving boundary approach. The magnitude of flow depth and the location of upstream boundary is documented in Table 2 for the two moving boundary criteria. The table shows that the shock is smeared with a phase shift using the RK approximation. The ENO scheme is slightly more consistent. Additional tests performed by varying the grid spacing and Courant number did not materially alter the results. A sensitivity analysis on the parameter \( l \) did marginally affect the location of the boundary in the RK formulation.

Based on the uniform flow criteria, Fig. 12 shows the location of the upstream boundary as a function of time for the two numerical schemes. The value of \( \varepsilon \) is 0.01. Its
trend is identical with Fig. 5 and was found to be in close agreement had the boundary been moved based on steady-state criteria. As it is difficult to arrive at any specific conclusion based on this plot, we performed a sensitivity analysis of $\varepsilon$ on the accuracy of the solution (Fig. 13). Though no specific trend can be established to indicate the optimal $\varepsilon$, one can fairly conclude that the ENO formulation is more robust. Since the formulation of ENO schemes, and in general the high-resolution schemes, are based on concepts like upwinding, field by

Table 1
Depth profiles near the location of shock for negative surge ($t \approx 318.2$ s, $\varepsilon = 0.01$)

<table>
<thead>
<tr>
<th>$x$</th>
<th>ENO</th>
<th>RK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixed$^a$</td>
<td>Moving$^a$</td>
</tr>
<tr>
<td>2485</td>
<td>6.03</td>
<td>6.03</td>
</tr>
<tr>
<td>2490</td>
<td>6.18</td>
<td>6.19</td>
</tr>
<tr>
<td>2495</td>
<td>6.94</td>
<td>6.99</td>
</tr>
<tr>
<td>2500</td>
<td>8.17</td>
<td>8.19</td>
</tr>
<tr>
<td>2505</td>
<td>8.57</td>
<td>8.58</td>
</tr>
<tr>
<td>2510</td>
<td>8.64</td>
<td>8.64</td>
</tr>
<tr>
<td>2515</td>
<td>8.65</td>
<td>8.65</td>
</tr>
<tr>
<td>2520</td>
<td>8.65</td>
<td>8.65</td>
</tr>
<tr>
<td>2525</td>
<td>8.65</td>
<td>8.65</td>
</tr>
<tr>
<td>2530</td>
<td>8.65</td>
<td>8.65</td>
</tr>
<tr>
<td>2535</td>
<td>8.65</td>
<td>8.65</td>
</tr>
</tbody>
</table>

$^a$ No moving boundary (i.e., fixed at $x = 5000$ m).
$^b$ Moving boundary based on uniform flow criteria Eq. (17).
$^c$ Moving boundary based on steady flow criteria Eq. (18).

Fig. 7. Transient depth profiles for negative surge at $t \approx 354$ s, ($\varepsilon = 0.001$, $m = 0.004$): (a) Normal view; (b) Zoom view. $x$ indicates the location of boundary from upstream.

Fig. 8. Transient depth profiles for negative surge at $t \approx 354$ s ($\varepsilon = 0.01$, $m = 0.004$): (a) Normal view; (b) Zoom view. $x(m)$ indicates the location of boundary from upstream.
field decomposition, Riemann solvers and flux limiting, their results for flows accompanied with shocks are qualitatively superior to space centered explicit formulations. When an attempt was made to move the boundary at the rate of one node per iteration, the code failed, since it over estimated the boundary velocity. In order to track the velocity of the boundary, the time step from the CFL condition Eq. (15) was found to be 0.4 s. Relating this time step with the corresponding grid spacing of 4 m, gives the boundary velocity of 10 m/s while the numerical velocity of the wave front is 6.6 m/s.

As evident in Tables 1 and 2, the difference in the boundary locations between the RK and ENO formulations can be attributed to the smoothing mechanism in the RK formulation. This smearing is independent of the dissipation constant magnitude Eq. (7) [3], and is a characteristic feature of space centered explicit schemes. A moving boundary approach coupled with the ENO formulation reflected in a CPU saving of 32%. The corresponding saving with the RK formulation was of the order of 41%.

7. Conclusions

In this work, we have explored the application of a moving boundary in simulating open channel flows. Flows resulting from sudden operation of control gates were studied using this approach. For simplicity we have confined this investigation to time independent boundary conditions. In summarizing this investigation, we wish to point out that there are many new aspects in this

![Fig. 9. Transient depth profiles for negative surge (t \( \geq \) 354 s, \( \varepsilon = 0.1, m = 0.004 \)): (a) Normal view; (b) Zoom view. x indicates the location of boundary from upstream.](image)

![Fig. 10. Definition sketch of a positive surge.](image)

![Fig. 11. Transient depth profile for positive surge at t \( \geq \) 100 s (\( \varepsilon = 0.01, m = 0 \)): (a) Normal view; (b) Zoom view. x indicates the location of boundary from upstream.](image)
work. This appears to be the first implementation of Lagrangian/Eulerian formulation in moving the boundary for open channel flows. All the studies reported until now [2–5,14–16] have used a fixed boundary in evaluating the transient flow profiles. Implementing this approach did highlight the fact that the performance of a moving boundary approach depends on the basic discretization technique and the associated smoothening mechanism, if required.

The results of our computation show that this approach is more versatile than the radiation techniques [8,9]. A satisfactory location of the wave front was related to the position of the associated boundary, to result in the constriction of flow domain. A smaller flow domain helps in faster convergence of the solution, thereby overcoming to certain extent the limitation imposed by using a explicit schemes i.e., small time step. We find that using a moving boundary approach is a significant improvement over conventional fixed boundary applications, being both accurate and computationally fast.

### Table 2

<table>
<thead>
<tr>
<th>x</th>
<th>ENO</th>
<th>RK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixeda</td>
<td>Movingb</td>
</tr>
<tr>
<td>636</td>
<td>2.51</td>
<td>2.48</td>
</tr>
<tr>
<td>640</td>
<td>2.51</td>
<td>2.55</td>
</tr>
<tr>
<td>644</td>
<td>2.52</td>
<td>2.57</td>
</tr>
<tr>
<td>648</td>
<td>2.52</td>
<td>2.45</td>
</tr>
<tr>
<td>652</td>
<td>2.37</td>
<td>2.45</td>
</tr>
<tr>
<td>656</td>
<td>1.99</td>
<td>2.18</td>
</tr>
<tr>
<td>660</td>
<td>1.52</td>
<td>1.71</td>
</tr>
<tr>
<td>664</td>
<td>1.20</td>
<td>1.30</td>
</tr>
<tr>
<td>668</td>
<td>1.06</td>
<td>1.10</td>
</tr>
</tbody>
</table>

a No moving boundary (i.e., fixed at x = 0 m).
b Moving boundary based on uniform flow criteria Eq. (17).
c Moving boundary based on steady flow criteria Eq. (18).

### Appendix A

Program coding for Eqs. (16)–(18), respectively.

If($|h_N^n - h_{N-2}^n| \leq \varepsilon$) then

\[ N = N - 1 \]  

End if

If($|h_N^n - h_{N-2}^n| \leq \varepsilon$) then

\[ L = L - \sqrt{gh_N^N \Delta t} \]  

\[ NG = L/\Delta x \]  

\[ N = NG + 1 \]  

End if

if($|h_N^{n+1} - h_N^n| \leq \varepsilon$) then

\[ L = L - \sqrt{gh_N^{n+1} \Delta t} \]  

\[ NG = L/\Delta x \]  

\[ N = NG + 1 \]  

End if

---

**Fig. 12.** Transient location of upstream boundary for sudden gate opening ($\varepsilon = 0.01$, $m = 0$).

**Fig. 13.** Effect of $\varepsilon$ on positive surge ($t \geq 100$ s, $m = 0$).
References