Staggered Scheme for Shallow Water Equations with Quadtree-subgrid

F. Fristella¹, S.R. Pudjaprasetya²

¹Computational Science, Faculty of Mathematics and Natural Science, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia
²Industrial and Finance Mathematics Research Group, Faculty of Mathematics and Natural Science, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132, Indonesia

E-mail: ¹friskafristella@ymail.com, ²sr.pudjap@math.itb.ac.id

Abstract. To get accurate results, high-resolution numerical simulations are often required. However, computation using high resolution grid often require larger computer memory and longer computational time. One way to overcome this problem is by adopting a scheme that can handle a non-homogeneous grid; i.e. a computational domain with two or more resolutions. In this work, a two-level quadtree-subgrid method was applied. It uses a combination of structured grid with hierarchical quadtree ordering. This method is applied on a staggered grid scheme for the two-dimensional shallow water equations. A benchmark simulation of wave run-up on a conical island was conducted for validation. Comparison between computing time of simulations using the fine grid and the quadtree-subgrid shows the effectiveness of this quadtree-subgrid scheme.

1. Introduction

It is well known that shallow water equations (SWE) are suitable model for simulating various surface flows in lakes, rivers, coastal areas, as well as flooding. In application of numerical models like SWE to predict floods or tsunami run up, high precision calculation are often needed. Modern technology could provide us a graphical information system with raster-based digital models of elevation/topography, abbreviated as DEMs. The existence of these high resolutions digital elevation data (DEMs) allows engineers to predict flooding with relevant accuracy.

However, computation using high resolution grid will require high computational cost. Sometimes this high resolution computations are not needed especially when there were wide areas of slowly varying topography. In order to increase the efficiency, and at the same time maintaining its accuracy, one can conduct computations using several grid sizes at the same time; larger grid size for nearly flat bottom and smaller grid size for highly varying bottom. Stelling in [5] formulates the quadtree-subgrid method implemented to the staggered grid domain for solving the 2-dimensional SWE.

In Stelling & Duijnmeijer (2003) [4], the momentum conservative scheme for solving SWE was discussed extensively. Basically, the scheme solves SWE on a 2-dimensional Arakawa C-grid. A special discretization procedure for the advection terms were implemented to ensure
that momentum is conserved during the evolution. As shown in [4], as well as in [7, 3] the momentum conservative scheme is relatively simple, efficient, and robust. This method is widely used in various free surface flow software. One drawback was, for grid refinement, this staggered grid based scheme requires some extra work. Simulation carried out here is the first step in implementing quadtree-subgrid for 2-dimensional SWE simulation. The benchmark simulation of a solitary wave run up on a conical island serves for validation purposes.

Here, we adopt the Shallow Water Equations, that holds with the underlying assumption that the vertical length scale is relatively smaller than the horizontal length scale. Further, neglecting the advection terms, lead us to the following half-nonlinear 2 dimensional SWE

\[
\begin{align*}
  h_t + (uh)_x + (vh)_y &= 0 \\
  u_t + g\eta_x &= 0 \\
  v_t + g\eta_y &= 0
\end{align*}
\]

where \( h(x, y, t) = \eta(x, y, t) - e(x, y) \) the water thickness, \( \eta(x, y, t) \) the water height, \( e(x, y) \) the bottom topography/bathymetry, see Figure 1. Here, \( g \) is the acceleration of gravity and \( u(x, y, t), v(x, y, t) \) denote horizontal velocity in -x and -y directions, respectively.

We implement the quadtree-subgrid as proposed by Stelling in [5] using only two different levels of subgrid. One large grid of level 1 is divided into four subgrids of level 0. We conduct one crucial benchmark test, i.e. simulation of wave run-up on a conical island [2]. Validation of numerical results with experiment data is given. Moreover, the effectiveness of quadtree-subgrid is shown by comparing the computation times between simulations using fine grids and quadtree-subgrid.

The organization of this paper is as follows. In Section 2, we discuss domain discretization of the quadtree-subgrid. The specific way of numbering cells of different levels, as well as their neighbors are described. In Section 3, discrete form of the half linear SWE are discussed, in particular how to manage cells that has neighbors from different level. In Section 4, the benchmark test of wave run up on a conical island was conducted for numerical validation purposes. Conclusions are given in the last section.

2. Quadtree-subgrid
In solving 2-dimensional SWE on a staggered grid, our focus is on implementing the quadtree-subgrid. First, we will discuss the hierarchical of quadtree ordering.

Each grid location is indexed by \( (l, m, n) \), with \( l \) represents the grid level, \( m \) and \( n \) are column and row positions of grid at level \( l \). In this work, level of grid is limited to the maximum level
Figure 2: (Left) Level of subgrids, (Middle) Domain discretization with a quadtree-subgrid, (Right) The quarter sub-domains.

1, meaning that only two grid sizes are involved in the computation, see Figure 2 (Left). Figure 2 (Middle) is the example of domain discretization with quadtree-subgrid, with $\Delta x_1 = 2^l \Gamma \delta x$ is the grid size at level $l$, and $\delta x$ is the size of the DEMs pixel, and $\Gamma$ is a constant factor. Here we just take $\Gamma = 1$, and the size of $\Delta x_0 = \delta x$ for the fine grid, and $\Delta x_1 = 2\delta x$ for the coarse grid. Furthermore, refinement follows a quadtree rule, the quarter sub-domains are defined following compass directions which are: $SW$ (Southwest), $SE$ (Southeast), $NW$ (Northwest), and $NE$ (Northeast), as in Figure 2 (Right), with $M$ is notation for sub-domains.

$$M_{(l,m,n)} = [(m-1)\Delta x_l, m\Delta x_l] \times [(n-1)\Delta y_l, n\Delta y_l]$$

$$M_{SW(l,m,n)} = [(m-1)\Delta x_l, (m-1/2)\Delta x_l] \times [(n-1)\Delta y_l, (n-1/2)\Delta y_l]$$

$$M_{SE(l,m,n)} = [(m-1/2)\Delta x_l, m\Delta x_l] \times [(n-1)\Delta y_l, (n-1/2)\Delta y_l]$$

$$M_{NW(l,m,n)} = [(m-1)\Delta x_l, (m-1/2)\Delta x_l] \times [(n-1/2)\Delta y_l, n\Delta y_l]$$

$$M_{NE(l,m,n)} = [(m-1/2)\Delta x_l, m\Delta x_l] \times [(n-1/2)\Delta y_l, n\Delta y_l].$$

Each of this subgrid also have their corresponding notations from higher level, which are as follows

$$M_{SW(l,m,n),SW} = M_{l,m,n-1,1}$$
$$M_{SE(l,m,n),SE} = M_{l-1,m,n,1,1}$$

This relationship is important for quadtree numbering and neighborhood finding.

2.1. Quadtree Numbering

Domain discretization using quadtree utilizes a set of grids with hierarchical ordering. Each computation grid is called leaf and has its own number which follows the quadtree numbering rule. The numbering of quadtree-subgrid starts from sub-domain $SW$, $SE$, $NW$, and the last is $NE$.

Figure 2 is an example of a quadtree numbering for 2-dimensional domain. The relationship between leaf number and sub-domain $M$ can be seen from these two examples, $leaf(2) = M_{(1,2,1)}$ and $leaf(7) = M_{SE(1,2,2)} = M_{(0,4,3)}$.

2.2. Neighborhood

The important step for working with numerical method using quadtree subgrid is numbering each leaf, as well as neighbors for each leaf. Each leaf has neighbors called $W$ (West), $E$ (East),
For position of each leaf\( (i) \) and their neighbors, there are three possibilities; in the same level, higher level, and/or lower level, see Figure 3.

Figure 3: Three different possibilities of leaf position and its neighbors, (Left) the same level, (Middle) higher level, and (Right) lower level.

3. Discrete form on a staggered scheme with Quadtree-subgrid
On the Arakawa C-grid the staggered scheme uses two different control volumes for continuity equation (mass cell) and momentum equation (momentum cell).

3.1. Conservation of fluid in mass cell
\[
V_t + \sum_{\partial \mathcal{M}} Q_{n_f} = 0 \quad \text{at } (l, m, n),
\]
where \( \sum_{\partial \mathcal{M}} Q \) is the total fluxes of all faces of \( \mathcal{M}_{(l,m,n)} \) and \( n_f \) is the outward normal vector. Several terms in (4) are elaborated below.

- Fluxes which equal to velocity times area
  \[
  Q_{l,m+\frac{1}{2},n}^x (t) = A_{l,m+\frac{1}{2},n}^x (t) u_{l,m+\frac{1}{2},n} (t)
  \]
  \[
  Q_{l,m,n+\frac{1}{2}}^y (t) = A_{l,m,n+\frac{1}{2}}^y (t) v_{l,m,n+\frac{1}{2}} (t)
  \]
- Area of faces
  \[
  A_{l,m+\frac{1}{2},n}^x (t) = \delta x \sum_{j=j_0}^{j=j_1} \max(0, \eta_{l,m+\frac{1}{2},n}^x - c_{i+\frac{1}{2},j})
  \]
for $i = 2^l \Gamma m$, $j_0 = 2^l \Gamma (n - 1) + 1$, $j_1 = 2^l \Gamma n$, and $e_{i+\frac{1}{2},j} = \max(e_{i,j}, e_{i+1,j})$.

\begin{equation}
A_{l,m,n+\frac{1}{2}}^y(t) = \delta y \sum_{i=i_0}^{i=i_1} \max(0, \eta_{l,m,n+\frac{1}{2}}^* - e_{i,j+\frac{1}{2}}) \tag{8}
\end{equation}

for $j = 2^l \Gamma n$, $i_0 = 2^l \Gamma (m - 1) + 1$, $i_1 = 2^l \Gamma m$, and $e_{i,j+\frac{1}{2}} = \max(e_{i,j}, e_{i,j+1})$.

- Value of $^*\eta$ is approximated by using the upwind rule

\begin{align}
^*\eta_{l,m+\frac{1}{2},n} &= \max \left(0, \frac{u_{l,m+\frac{1}{2},n}}{|u_{l,m+\frac{1}{2},n}|} \eta_{l,m,n} - \min \left(0, \frac{u_{l,m+\frac{1}{2},n}}{|u_{l,m+\frac{1}{2},n}|} \right) \eta_{l,m+1,n} \right) \tag{9} \\
^*\eta_{l,m,n+\frac{1}{2}} &= \max \left(0, \frac{v_{l,m,n+\frac{1}{2}}}{|v_{l,m,n+\frac{1}{2}}|} \eta_{l,m,n} - \min \left(0, \frac{v_{l,m,n+\frac{1}{2}}}{|v_{l,m,n+\frac{1}{2}}|} \right) \eta_{l,m,n+1} \right) \tag{10}
\end{align}

- Volume $V$

\begin{equation}
V_{l,m,n}(t) = \delta x \delta y \sum_{i=i_0}^{i=i_1} \sum_{j=j_0}^{j=j_1} \max(0, \eta_{l,m,n} - e_{i,j}) \tag{11}
\end{equation}

with $i_0 = 2^l \Gamma (m - 1) + 1$, $i_1 = 2^l \Gamma m$, $j_0 = 2^l \Gamma (n - 1) + 1$, and $j_1 = 2^l \Gamma n$.

Further, discrete form of momentum equations read as

\begin{align}
u_t + g \left( \frac{\eta^E - \eta^W}{(\Delta x^E + \Delta x^W)/2} \right) &= 0 \quad \text{at } (l, m + \frac{1}{2}, n) \tag{12} \\
v_t + g \left( \frac{\eta^N - \eta^S}{(\Delta y^N + \Delta y^S)/2} \right) &= 0 \quad \text{at } (l, m, n + \frac{1}{2}) \tag{13}
\end{align}

In Figures (5) we gave example on calculating $\eta^E$, $\eta^W$ for three cases below: $l^W = l^E$, $l^W < l^E$, and $l^W > l^E$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{(Left) Case $l^W < l^E$ and (right) case $l^W > l^E$.}
\end{figure}
\begin{itemize}
\item \(l^W = l^E\)

\[\eta^W = \eta_{l,m,n} \quad \eta^E = \eta_{l,m+1,n}\]

\item \(l^W < l^E\) (see Fig. 5 Left)

\[n\text{ even}\]
\[\eta^W = \frac{\eta_{l,m,n} + \eta_{l,m,n-1}}{2} \quad \eta^E = \eta_{l+1,(m/2)+1,n/2}\]
\[n\text{ odd}\]
\[\eta^W = \frac{\eta_{l,m,n} + \eta_{l,m,n+1}}{2} \quad \eta^E = \eta_{l+1,(m/2)+1,(n+1)/2}\]

\item \(l^W > l^E\) (see Fig. 5 Right)

\[n\text{ even}\]
\[\eta^W = \eta_{l+1,m/2,n/2} \quad \eta^E = \frac{\eta_{l,m+1,n} + \eta_{l,m+1,n-1}}{2}\]
\[n\text{ odd}\]
\[\eta^W = \eta_{l+1,m/2,(n+1)/2} \quad \eta^E = \frac{\eta_{l,m+1,n} + \eta_{l,m+1,n+1}}{2}\]
\end{itemize}

4. Wave run-up on a conical island

In 1995, Briggs et al. [1] conducted an experimental study of wave run-up on a conical island. This experiment constitutes as one of benchmark tests for tsunami run up modelling [6]. This experimental data is now available as an open source provided by NOAA Center for supporting researches related to tsunami. Here we take a simulation domain as a large basin 25 m \times 30 m with a small island located at \(x = 12.96\) and \(y = 13.80\), Figure 6 shows the detailed of the island. We discretize the domain using a quadtree-subgrid, including the refinement around the island, see Figure 8.

A solitary wave with amplitude 0.0144 m, as depicted in Figure 7 with explicit formula \(h(0,y,t) = 0.32 + 0.0144\) enters from the left side of the domain. Once the solitary wave has entered the domain, the left boundary was changed to a hardwall type.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{bathymetry.png}
\caption{Bathymetry of the conical island benchmark test.}
\end{figure}
Figure 7: A solitary profile for wave influx.

Figure 8: Quadtree-subgrid and gauges positions.

Hard-wall boundary conditions were implemented along the other sides of the basin: $x = 25, y = 0, y = 30$. For simulation that involves dry areas, we need to employ the following wet-dry procedure: discrete momentum equations (12, 13) are computed when the cell is wet (indicated by $h > 0$) $h_{\text{leaf}[i]} > 0$ or $h_{\text{leaf}[i]_{\text{neighbor}}} > 0$, when it is dry, we set the corresponding velocity equals to zero.

In the experiment there were 27 gauges located around the island. These gauges recorded wave height signals at several positions resumed in Table 1. Here we compare our numerical results at the four gauges, and they were presented in Figure 8.

The wave height at four different positions of wave gauges were calculated using fine grids, quadtree-subgrid were recorded and compared with experimental data. As shown in Figure 9, the computed wave signals at gauge 6 are nicely comparable with the experimental data, whereas wave signals at gauges 9, 16, and 22 are in fair agreement with the experiments.

Figure 9 presents simulation results in comparison with experimental data. It shows that the computed wave signals already gave fair agreement with the benchmark experimental data.

| Gauge | experiment | simulation |
|-------|------------|------------|
| 6     | (9.36, 13.80) | (9.35, 13.85) |
| 9     | (10.36, 13.80) | (10.35, 13.85) |
| 16    | (12.96, 11.22) | (12.95, 11.25) |
| 22    | (15.56, 13.80) | (15.55, 13.85) |
even tough the advection terms have not been incorporated. Moreover, comparison between wave signals computed using the quadtree-subgrid and the fine grid simulations have shown no significant differences. These simulations were conducted using C++ software on a 64 bit personal computer with 8GB RAM, processor core i5. The number of grid used and the computation times of these two computations, as resumed in Table 2 shows significant differences; the quadtree-subgrid simulation is three times faster than the fine grid simulation.

5. Conclusions
Efficient yet accurate numerical models have been considered. The two level of quadtree-subgrid on a staggered grid has been implemented to solve the 2-dimensional shallow water equations. Simulation of solitary wave run-up on a conical island has shown a good agreement with experimental data. Moreover, the comparison of computational times between the fine-grid and quadtree-subgrid has shown the effectiveness of this quadtree-subgrid scheme.

References
[1] Briggs M J, Synolakis C E, Harkins G S and Green D 1995 Pure Appl. Geophys. 144 569–93.
[2] Liu P, Cho Y S, Briggs M J, Kanoğlu U and Synolakis C E 1995 J. Fluid Mech. 302 259–85.
[3] Magdalena I, Erwina N and Pudjaprasetya S R 2015 J. Sci. Comput. 65 867–874.
[4] Stelling G S and Duinmeijer S P A 2003 Int. J. Numer. Meth. Fluids 43, 1329–1354.
[5] Stelling G S 2012 P. I. Civil Eng. Wat. M. 165 567 – 80.
[6] Synolakis C E, Bernard E N, Titov V V, Kanoğlu U and González F I, 2007 Standards criteria, and procedures for NOAA evaluation of tsunami numerical models https://nctr.pmel.noaa.gov/benchmark/SP_3053.pdf
[7] Pudjaprasetya S R and Magdalena I 2014 E. Asian J. Appl. Math. 4, 152–65.

Table 2: Comparison between the fine-grid versus the quadtree-subgrid computation times of wave run up on a conical island.